

# WPA: Write Pattern Aware Hybrid Disk Buffer Management for Improving Lifespan of NAND Flash Memory

Jun-Hyeong Choi<sup>1</sup>, Kyung Min Kim, and Jong Wook Kwak<sup>1</sup>

**Abstract**—Recently, consumer electronics devices have started adopting NAND flash memories as their storage units. NAND flash memories are suitable as portable consumer device storages, as they have several advantages such as non-volatility, high density and low power consumption. However, flash memories are difficult to be used directly as storages in consumer devices because of a limited lifespan, which is one of the major drawbacks of NAND flash memory. Therefore, researchers have investigated ways for improving the lifespan of flash memories. In this article, we propose a hybrid disk buffer management policy, which is called the write pattern aware hybrid disk buffer management policy (WPA). The key advantage of WPA is the reduction of write access to the flash-based storage, and this is possible because WPA analyzes the write access pattern that occurred through our pattern aware migration priority by LRU (PaMP-LRU) algorithm and a page eviction list (PEL). We compared our proposed policy with existing hybrid disk buffer policies to evaluate the performance. Experimental results show that under conditions of write-intensive workloads, the performance of WPA is found to be higher than that of other disk buffer policies. In particular, the improvement of WPA hit ratio is 87.6% on average, and WPA reduces block erase counts in flash storage 38.2% on average, and up to 66.8%, resulting in improving the lifespan of NAND flash memory, compared to other disk buffer management policies.

**Index Terms**—Hybrid disk buffer, NAND flash memory, non-volatile RAM, page replacement, write access patterns.

## I. INTRODUCTION

NAND flash memory has been widely used in various consumer devices such as embedded devices, IoT devices and smartphones [1]–[8]. As a result, NAND flash memory rapidly has replaced a magnetic disk as the storage of consumer electronics devices [9], [10]. However, the flash memory has several disadvantages because of its hardware characteristics, and many researchers have studied possible ways to resolve the issue within the limits of the flash memory [9]–[15]. The critical disadvantage of NAND flash memory is the limited lifespan, which is further degraded by frequent write accesses. In order to resolve this problem,

*garbage collection* and *wear leveling* techniques have been proposed and development of effective disk buffer management policies also have been investigated by various researchers [11]–[14]. In particular, as functions of consumer devices have been diversified, the size of the storage has increased. As a result, the need for the disk buffer in consumer devices has been emphasized.

Previously, disk buffers used *dynamic random-access memory* (DRAM) in flash-based storage systems for reducing the number of direct accesses to the storage. The disk buffer manages pages similar to page replacement policies of the main memory such as *least recently used* (LRU) algorithm, CLOCK algorithm and many other variations. The disk buffer responds to most of read and write requests of a host system instead of storage, which allows achieving the system performance improvement [8]. Consequently, using the disk buffers resulted in reducing the performance gap between storage and the main memory due to limiting accesses to storage. In particular, the reduced number of accesses to flash-based storage allows diminishing the number of block erase operations in flash-based storage. However, existing disk buffers are composed of DRAM alone, which has the limitations such as high power consumption and low density. Therefore, the system performance of existing disk buffers decreases in large size applications, which include frequent read and write access operations.

Recently, researchers have proposed hybrid disk buffer structure and its management policy to solve the limitation of the flash storage and to enhance the performance of DRAM alone disk buffer [7], [16]. Unlike DRAMs, *non-volatile random-access memories* (NVRAMs) have such advantages as the high density and the low power consumption [19]–[28]. Owing to these features of NVRAM, hybrid disk buffer resolves the limit problem of the previous disk buffer system, which had the low density and the high-power consumption [17], [18]. Due to the increased size of high-density disk buffer, the hybrid disk buffer can process more pages. This allows reducing the number of read and write accesses from the host system to flash-based storage. That is, when the hybrid disk buffer receives the requests of the host system, the lifespan of NAND flash memory can be improved due to the reduced direct read and write access requests to storage.

In this article we propose a hybrid disk buffer management policy called WPA (Write Pattern Aware) with *pattern aware*

Manuscript received November 1, 2019; revised January 9, 2020, February 24, 2020, and March 12, 2020; accepted March 13, 2020. Date of publication March 18, 2020; date of current version April 23, 2020. This work was supported by the 2019 Yeungnam University Research Grant. (Corresponding author: Jong Wook Kwak.)

The authors are with the Department of Computer Engineering, Yeungnam University, Gyeongbuk 38541, South Korea (e-mail: runchoice@ynu.ac.kr; bl43a@ynu.ac.kr; kwak@yu.ac.kr).

Digital Object Identifier 10.1109/TCE.2020.2981618

TABLE I  
COMPARISON OF MEMORY TECHNOLOGIES

	Cell size (F <sup>2</sup> )	Data Retention (years)	Read latency (ns)	Write latency (ns)	Voltage (V)	Write energy (J/bit)	Endurance (Write count)
SRAM	> 100	N/A	~ 1	~ 1	< 1	~ fJ	10 <sup>16</sup>
DRAM	6	~ 64 ms	~ 10	~ 10	< 1	~ 10fJ	> 10 <sup>15</sup>
NOR flash	10	> 10	~ 50	~ 10 <sup>4</sup> – 10 <sup>6</sup>	> 10	~ 100pJ	10 <sup>5</sup> ~ 10 <sup>6</sup>
NAND flash	4	> 10	~ 10 <sup>4</sup>	~ 10 <sup>5</sup> ~ 10 <sup>6</sup>	> 10	~ 10fJ	10 <sup>4</sup> ~ 10 <sup>5</sup>
ReRAM	4 ~ 12	> 10	< 10	< 10	< 1.5	~ 0.1pJ	10 <sup>6</sup> ~ 10 <sup>12</sup>
STT-MRAM	6 ~ 50	> 10	< 10	< 10	< 1.5	~ 0.1pJ	10 <sup>14</sup> ~ 10 <sup>15</sup>
PCM	4 ~ 30	> 10	< 10	< 50	< 3	~ 10pJ	10 <sup>8</sup> ~ 10 <sup>9</sup>

*migration priority by LRU* (PaMP-LRU) algorithm, using the write access pattern analysis. PaMP-LRU algorithm analyzes the write access patterns using two flags, called *re-reference* bit (*Re-Ref* bit) and *overwrite* bit (*OW* bit) that monitor the past and current temporal locality. Consumer devices such as gaming consoles and smart TVs tend to refer to the same address again for write operations. Therefore, the write pattern analysis is important for improving the performance of consumer devices and for enhancing the limited lifespan of NAND flash memory in consumer devices. In our policy, WPA allocates pages to DRAM and NVRAM buffers depending on request types of the host system. This means that WPA allocates write pages in the DRAM buffer and read pages in the NVRAM buffer. Unlike other existing policies, WPA considers the write pattern of evicted pages from the past in the hybrid disk buffer. To analyze the past write pattern, WPA composes an additional list, called *page eviction list* (PEL), which manages the page number of evicted pages. In WPA policy, the DRAM buffer manages pages by page units and the NVRAM buffer manages pages by block units. As a result, it efficiently changes several random write accesses to one sequential write access. Frequent random write accesses cause many block erase operations in flash-based storage, and thus, it has a severe negative impact on the lifespan of NAND flash memory. However, when the several random writes are changed to one sequential write, the number of block erase operations is reduced in storage. Therefore, applying the sequential write access to storage improves the lifespan of NAND flash memory.

It can be seen that a large performance difference occurs depending on whether disk buffers are effectively utilized or not [29]–[32]. The disk buffer and its policy have a big impact on consumers. Therefore, from the perspective of consumers, this article will help to improve the understanding of disk buffer operation, and consumers can directly feel the performance differences, by considering the disk buffer, such as program response time, battery duration time, and product lifespan, to help select their actual product.

The rest of the article is organized as follows. Section II describes the background of memory technologies and the existing hybrid disk buffer policies. Section III presents in detail the proposed WPA hybrid disk buffer policy with its structure, algorithm and operations. In Section IV, we measure

and evaluate the performance of WPA by comparing it with existing disk buffer policies. Finally, Section V presents the conclusions.

## II. BACKGROUND AND RELATED WORKS

In this section, we describe memory technologies based on NVRAMs and discuss existing hybrid disk buffer policies.

### A. Non-Volatile Memory Technology

Recently, NVRAMs have been introduced to replace DRAMs in various architectures in order to address DRAMs limitations such as high leakage power consumption for the periodic refresh and DRAM also suffers relatively lower density [26]–[28]. NVRAMs are byte addressable memory and have low latency, which is similar to DRAMs. However, unlike DRAMs, NVRAMs are characterized by high density, low power consumption and non-volatility [19]–[25]. Due to these characteristics, NVRAMs are widely used in various areas such as consumer devices, embedded devices and storage systems. Table I presents comparisons among several memory technologies such as RAMs, flash memories, and NVRAMs. It describes density, latency, energy and endurance of each memory.

NAND/NOR flash memory is a type of non-volatile memories that is used in storage systems [9]–[15]. It is faster than magnetic disks, which are used as a traditional storage. In particular, NAND flash memory has higher density compared to other non-volatile memories. Due to these features of NAND flash memory, it rapidly has replaced magnetic disks in various storage devices such as consumer devices, embedded systems and personal mobile systems. Although NAND flash memory can be considered as an appropriate storage element, however, it has a problem of the limited lifespan. It is caused by the hardware characteristics of NAND flash memories. NAND flash memory consists of cells, which have the limited write endurance. Therefore, many researchers have studied the ways to improve the endurance of this memory type.

ReRAMs, STT-MRAMs and PCMs are representative NVRAMs and recently, they have been actively investigated to replace traditional DRAMs or even in flash-based storages in consumer electronics devices [26]–[28]. ReRAMs are one

of the state-of-the-art storages that are suitable for most of high performance computing environments, because of low latency, low operating voltage and low energy consumption. However, ReRAMs have a large variation in the endurance and the performance depending on the type of material and have many debates on reliability [33]–[36]. STT-MRAMs have the highest durability due to their magnetic properties. Therefore, it is advantageous in an environment in which steady writings occur such as a sensing device. In addition, STT-MRAM has high scalability, but high write current must be considered and, due to its magnetization properties, it is relatively vulnerable to thermal stability [20], [21]. PCMs have lower latency and higher durability than NAND flash memory. Among NVRAMs except NAND, the density of PCM is usually considered as the highest due to the lowest cell size. However, PCMs have lower durability than other next-generation NVRAMs, and they require a large write energy and latency, making it unsuitable for low-power mobile environments [22], [26]–[28].

### B. Hybrid Disk Buffer Policy

Hybrid disk buffer as well as disk buffer improves system performance by processing requests from the host system of consumer devices to the storage. Consumer devices consist of the limited disk buffer, therefore which requires an efficient buffer management policy. Unlike existing disk buffers, hybrid disk buffer consists of DRAM and NVRAM. Previously, hybrid disk buffer used LRU or CLOCK algorithms for the page replacement policy within the disk buffer. However, LRU and CLOCK policies do not consider the characteristics of the hybrid disk buffer and flash memory in its storage, such as limited lifespan and the unique performance of NVRAMs and NAND flash memories, and consequently, possibilities to improve the system performance in such cases are limited and they are difficult to be directly implemented in hybrid disk buffers with flash storages. Therefore, hybrid disk buffer management policy must consider these unique characteristics of NVRAMs and NAND flash memories. The following provides the description of corresponding disk buffer and hybrid disk buffer policies for flash-based storage systems.

*Block padding LRU (BPLRU)* is a disk buffer policy, which manages the requests of the host system by block units [16]. The disk buffer of BPLRU is implemented using DRAM, and the BPLRU is considered to be used in flash-based storage systems. Random write accesses to the storage have negative impact on the system performance as well as on the lifespan of NAND flash memory. To overcome this limitation, BPLRU changed random writes to a sequential write. When a disk buffer is full, BPLRU selects a victim block using the traditional LRU policy. BPLRU reads pages from the storage, which are included in the victim block, and merges these pages on the victim block. However, when BPLRU is applied to the hybrid disk buffer, the host system directly reads pages from the storage, because BPLRU considers write requests only. Consequently, when workloads include frequent read requests in most cases, the system performance is degraded.

*Cooperative Buffer Management (CBM)* is a hybrid disk buffer policy, which allows managing both read pages and

write pages, unlike BPLRU [17]. CBM consists of DRAM and NVRAM, which are used as read cache and write buffer, respectively. In particular, write buffer of CBM is separated as two regions, which are called page region and block region. When write buffer does not have a free page, CBM migrates the page, which is located by an LRU position in the page region, to the block region. The block region of CBM manages write pages by the block units, and CBM converts random writes to one sequential write. However, CBM manages write pages in the write buffer, which consists of NVRAM, having the high latency and the limited lifespan, compared to DRAMs. Therefore, when workloads are write-intensive, CBM frequently evicts pages in block region of CBM, even when read cache, which consists of DRAM, has enough free pages. Accordingly, the performance of CBM is impacted in write-intensive applications.

Unlike BPLRU and CBM, *CLOCK with DRAM and NVM hybrid write buffer (CLOCK-DNV)* followed the basic policy of CLOCK algorithm [18]. CLOCK-DNV allocates read pages in *CLOCK algorithm for DRAM (CLOCK-D)* and write pages in *CLOCK algorithm for NVM (CLOCK-NV)*. When CLOCK-D does not have a free page, CLOCK-DNV migrates the page from CLOCK-D to CLOCK-NV. In CLOCK-DNV, unlike CBM, CLOCK-D manages the page unit and CLOCK-NV manages the block unit. When CLOCK-NV is full, CLOCK-DNV selects a victim block, and CLOCK-D pages, which are included in the victim block, are merged on the victim block. However, since CLOCK-DNV manages pages through reference bit and dirty bit, like a traditional CLOCK, it is difficult to collect precise write patterns of evicted pages. Therefore, when the host system accesses the evicted pages again, the performance of CLOCK-DNV becomes progressively worse.

*Virtual-block-based buffer management scheme (VBBMS)* is a disk buffer policy, which manages the requests of the host system by virtual block unit [7]. VBBMS improves the disadvantage of BPLRU, which are the low hit ratio in the disk buffer, because BPLRU does not manage the read requests. The disk buffer of VBBMS is implemented using DRAM, and the VBBMS is considered to be used in flash-based storage systems. In addition, VBBMS enhances the performance of the system by using characteristics of random accesses and sequential accesses. VBBMS separates disk buffer by two lists, which are named *random request service region (RRSR)* and *sequential request service region (SRSR)*. VBBMS manages virtual blocks in RRSR by LRU algorithm, because that considers the temporal locality of random requests. On the other hand, VBBMS manages virtual blocks in SRSR by FIFO algorithm. However, when random requests are frequently occurred from the host system, VBBMS frequently evicts pages in the disk buffer. Due to unused pages in the virtual block in VBBMS, the efficient management of the disk buffer is difficult and garbage collection overhead is increased. As a result, overall of the system performance is reduced.

## III. THE DESIGN OF WPA

In this section, we propose the hybrid disk buffer management policy, which allows reducing the number of read

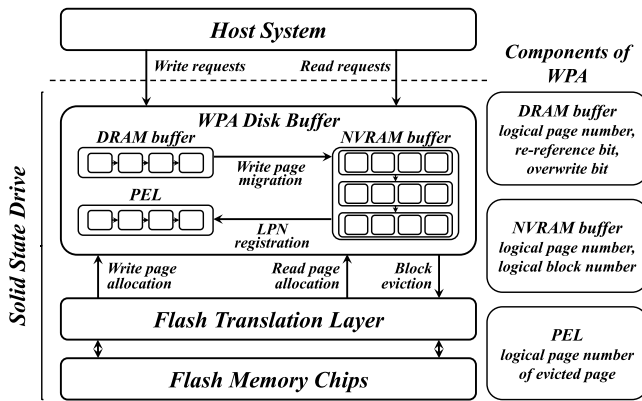


Fig. 1. The structure of WPA.

and write accesses and block erase counts in the flash-based storage. We named the hybrid disk buffer management policy as WPA. WPA has a feature, which manages pages using the write access pattern analysis through PaMP-LRU algorithm and PEL. In addition, WPA evicts the block unit from the NVRAM buffer to the storage by converting several random writes in DRAM buffer to one sequential write to reduce the number of block erases in the flash storage. The proposed hybrid disk buffer management policy has two key advantages:

1. It allows reducing the number of read/write accesses in the storage by providing higher hit ratio of hybrid disk buffer through the precise analysis of the past and current locality.
2. It allows reducing the block erase count in the flash storage by converting the random writes to one sequential write, resulting in the increase of lifespan in NAND flash memory.

WPA disk buffer consists of DRAM buffer, NVRAM buffer and PEL for managing read and write pages. The overall structure of WPA is presented in Fig. 1. WPA determines the location of pages according to types of requests. First, WPA allocates write pages into the DRAM buffer and manages pages by a page unit. The DRAM buffer changes the migration priority by two flags, which are called *Re-Ref* bit and *OW* bit. The migration priority is set by the combination of *Re-Ref* bit and *OW* bit, which are used to determine the location of the DRAM buffer pages, using PaMP-LRU algorithm. Second, WPA allocates read pages in the NVRAM buffer, which manages pages by a block unit. Lastly, PEL collects page numbers of evicted pages, which are used to analyze the write pattern of the past. When pages are evicted to the storage, the page numbers of the evicted pages are inserted into PEL.

#### A. The Structure of WPA

WPA consists of three components, namely, DRAM buffer, NVRAM buffer and PEL for managing read and write pages. Fig. 1 illustrates the overall structure of WPA. In Fig. 1, the DRAM buffer of WPA manages entries by page units and the NVRAM buffer by block units.

When the host system sends a request for the write access to pages, WPA allocates the pages in the DRAM buffer. The DRAM buffer considers the temporal locality of inserted pages. In addition, WPA considers the access history in the past. To provide this feature, WPA uses *Re-Ref* bit and *OW* bit. *Re-Ref* bit and *OW* bit are used to analyze the write pattern, which is applied effectively for the DRAM buffer management. In addition, when the DRAM buffer does not have free pages, WPA migrates a page from the DRAM buffer to the NVRAM buffer to make a free page in the DRAM buffer. The candidate page for this migration is selected based on the migration priority and the position of the page in the DRAM buffer. *Re-Ref* bit and *OW* bit combination and the migration priority management of the DRAM buffer pages are explained in detail in the next subsection.

On the other hand, WPA allocates required pages, which are requested from the host system by the read access, to the NVRAM buffer. In addition, WPA migrates the DRAM buffer pages to the NVRAM buffer. Therefore, the NVRAM buffer manages to read pages and to migrate pages. The NVRAM buffer of WPA manages the pages as the block units for converting several random writes to one sequential write. When the DRAM buffer page is migrated to the NVRAM buffer, the block, which contains the migrated page, is moved to the *most recently used* (MRU) position in the NVRAM buffer. When the host system accesses pages in the NVRAM buffer by the read operation, the accessed block is also moved to MRU position in the NVRAM buffer.

WPA manages pages in the DRAM buffer through a write pattern analysis and a temporal locality. To collect the write pattern in the past, WPA composes an additional list, which is called PEL in DRAM. When the block is evicted from the NVRAM buffer to the storage, WPA inserts page numbers of evicted write pages (i.e., dirty pages) into PEL. The inserted *logical page number* (LPN) is the write history in the past and WPA uses the page number to analyze the write pattern. When the write page allocation into the DRAM buffer starts, WPA verifies page numbers in PEL for changing the migration priority of the page allocation. Due to the verification of page numbers in PEL, WPA applies high migration priority to the page, which was accessed by the write operation in the past. Pages written once in various environments are usually accessed again by rewrite operation. The high migration priority of pages addresses the requirement of the host system at the DRAM buffer side. Therefore, WPA precisely considers characteristics of write accesses by selectively allocating the page in DRAM or NVRAM based on the operation types and improves the system performance by further controlling the placed location of hybrid disk buffer using migration priority.

#### B. PaMP-LRU Algorithm

In this subsection, we describe in detail the main algorithm of our proposal, which is called *pattern aware migration priority by LRU* (PaMP-LRU). All pages in hybrid disk buffer are managed by PaMP-LRU algorithm. PaMP-LRU decides the migration priority of pages in the DRAM buffer using

TABLE II  
THE CONDITION OF THE BIT FLAGS SETTING

Bit flags	Description
<i>Re-Ref</i>	If the page number of page $p$ exists in PEL when page $p$ is newly allocated in DRAM buffer, <i>Re-Ref</i> bit of page $p$ is set as 1. Otherwise, <i>Re-Ref</i> bit in page $p$ is set as 0.
<i>OW</i>	When write operation is occurred to dirty page $p$ , <i>Overwrite</i> bit of page $p$ is set as 1. Otherwise, <i>Overwrite</i> bit of page $p$ is set as 0.

page numbers in PEL. Table II shows the condition of bit flags setting, which is used in migration priority decision. In Table II, the page  $p$  is the accessed page by the write operation. Pages in the DRAM buffer are managed by *Re-Ref* bit and *OW* bit, and these two flags change the migration priority of the page in the DRAM buffer. When the page is inserted into the DRAM buffer, *Re-Ref* bit of the page is determined by PEL as follows. WPA defines the condition of *Re-Ref* bit setting through PEL. When the page number of the inserted page exists in PEL, *Re-Ref* bit of the inserted page is set as 1, otherwise as 0. *Re-Ref* bit verifies the write history in the past. Unlike *Re-Ref* bit, *OW* bit verifies current write patterns. When a page is inserted into the DRAM buffer, WPA initializes *OW* bit of the inserted page as 0. Then, if the host system accesses the inserted page again for a write operation, WPA sets *OW* bit of the accessed page as 1.

The migration priority is defined by *Re-Ref* bit and *OW* bit of the DRAM buffer page as follows. When both *Re-Ref* bit and *OW* bit of the page are 0, WPA determines the migration priority as 0, the lowest priority. The 0 priority means that the frequency of the write operation is low in the past and present. When *Re-Ref* bit is 0 and *OW* bit is 1, the migration priority is set as 1. Consequently, when *Re-Ref* bit is 1 and *OW* bit is 0, the migration priority of the page is 2. Lastly, when both *Re-Ref* bit and *OW* bit is 1, the migration priority of the page is set as 3, which is the highest priority. The accessed page of the highest priority in WPA policy will be frequently accessed in the future with high probability. The priority decision is defined based on the awareness of write pattern, which weights write accesses in the past and present. Therefore, WPA weights the referenced page by using *Re-Ref* bit and *OW* bit, and it can effectively manage write pages in the hybrid disk buffer.

Using this migration priority, we defined migration priority group of migration priority level  $i$  as  $mpg_i$ . That is, the inserted pages in DRAM buffer are grouped together according to the same migration priority level of pages. The highest migration priority group ( $mpg_3$ ), that is (*Re-Ref*, *OW*) as (1, 1), is located at MRU side and  $mpg_0$ , the lowest migration priority group (0, 0), is located at LRU side, respectively. In PaMP-LRU policy, each migration priority group,  $mpg_i$  has its own sub-LRU list as well. That is, pages in the same migration priority group  $i$  are ordered by LRU list again as  $mrui$  and  $lru_i$  of  $mpg_i$ .

Fig. 2 shows an example of the migration priority group management. When the host system newly accesses the page 9 with a write page allocation, WPA verifies the page number

### Algorithm 1 PaMP-LRU (Page $p$ , Operation $op$ )

---

```

Ensure: The accessed page is  $p$ , the operation is  $op$ , the block
that includes the page  $p$  is  $block(p)$ , the free page is  $page_f$ , the
free block is  $block_f$  and the migration priority group number is  $i$ ;
1 if  $p$  in DRAM buffer then
2   if  $op = write$  then
3     // Change migration priority group of  $p$  by Re-Ref bit and OW bit;
     ChangePriority( $p$ );
4   end if
5   Move  $p$  to  $mrui$  of new  $mpg_i$  in DRAM buffer;
6   else if  $p$  in NVRAM buffer then
7     Move  $block(p)$  to MRU of NVRAM buffer;
8   else
9     if  $op = write$  then
10      if DRAM buffer is full then
11        // Migrate the write page in DRAM buffer to NVRAM buffer;
         $page_f = DRAMPageMigration()$ ;
12      end if
13      ChangePriority( $p$ );
14      Insert  $p$  into  $page_f$ ;
15      Move  $p$  to  $mrui$  of new  $mpg_i$  in DRAM buffer;
16    else if  $op = read$  then
17      if NVRAM buffer is full then
18        // Evict the LRU block of NVRAM buffer to SSD;
         $block_f = NVRAMBlockEviction()$ ;
19      end if
20      Insert  $p$  into  $block_f$  and move to MRU of NVRAM buffer;
21    end if
22  end if

```

---

in PEL. In this example, the page number of the accessed page exists in PEL. Therefore *Re-Ref* bit of page 9 is set as 1. *OW* bit of page 9 is set as 0, because page 9 is newly inserted into the DRAM buffer. As a result, the migration priority group of page 9 is 2 ( $mpg_2$ ), and then, by PaMP-LRU policy, the page 9 is inserted in the position of  $mrui_2$ , which means MRU position of  $mpg_2$ .

Algorithm 1 describes in detail the PaMP-LRU algorithm. First, the host system requests the page  $p$  in the hybrid disk buffer, and WPA processes the page  $p$  through the allocated position of the page  $p$  (lines 1-7). When the page  $p$  is allocated in the DRAM buffer by a write operation, the migration priority of the page  $p$  is changed by *Re-Ref* bit and *OW* bit. Then the page  $p$  is moved to the MRU position of the changed migration priority group (lines 1-5). Conversely, when the page  $p$  exists in the NVRAM buffer, WPA moves the block, which includes the page  $p$ , to MRU position in the NVRAM buffer (lines 6-7).

When the page  $p$  does not exist in the hybrid disk buffer, the allocation position of the page is determined according to the operation type of the page  $p$  (lines 8-21). First, when the operation type of the page  $p$  is a write operation, WPA allocates the page  $p$  into the DRAM buffer (lines 9-15). When the DRAM buffer does not have free pages for inserting the page  $p$ , WPA selects the page to be migrated from the DRAM buffer to the NVRAM buffer according to the migration priority group management. After this, WPA migrates the page to the NVRAM buffer, the page  $p$  is inserted into a free page position and finally is moved to MRU position of the changed migration priority group (lines 10-15). Second, when the host system requires the page  $p$  as a read operation, the page  $p$  is allocated in the NVRAM buffer, and the block containing the page  $p$  is moved to MRU position in the NVRAM buffer (lines 16-20).

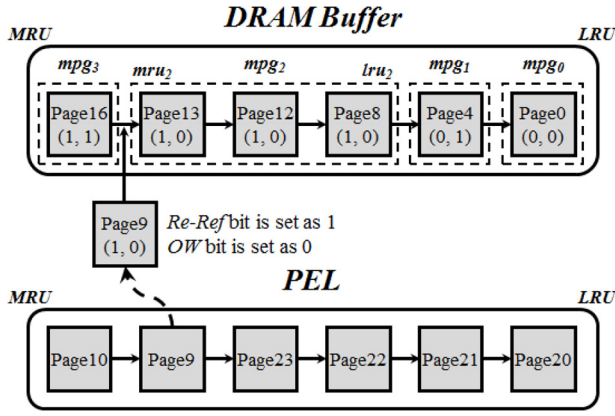


Fig. 2. Migration priority group management in DRAM buffer with PEL.

However, when the NVRAM buffer is full, WPA selects a victim block in LRU position of NVRAM buffer to obtain a free block in the NVRAM buffer (lines 17-19). Lastly, the page  $p$  is inserted into the free block, and the block containing the page  $p$  is moved to MRU position in the NVRAM buffer (line 20).

### C. Scenario of WPA

In this subsection, we present the description of representative scenarios. The scenarios have two assumptions. Firstly, the DRAM buffer and PEL consist of 4 pages, and the NVRAM buffer consists of 3 blocks with 4 pages in each block. Secondly, write pages, that are dirty pages, are illustrated by a gray color and read pages, that are clean pages, are depicted by a white one.

Fig. 3 provides an example of the page allocation in the DRAM buffer and the page migration from the DRAM buffer to the NVRAM buffer. When the host system requires page 0 for a write operation, WPA inserts page 0 into the DRAM buffer. However, in this example, the DRAM buffer does not have a free page. Therefore, the DRAM buffer of WPA migrates a page to the NVRAM buffer for inserting page 0. WPA selects page 11 as the victim page, because the migration priority of page 11 is 0 ( $mpg_0$ ) and the page resides in  $lru_0$  (Step 1). Page 11 as the victim page is migrated to block 2 in the NVRAM buffer, and block 2 is moved to MRU position in the NVRAM buffer (Step 2). Next, WPA verifies the page number of page 0 in PEL, before page 0 is inserted in the DRAM buffer. Because the page number of page 0 exists in PEL, *Re-Ref* bit of page 0 is set as 1, and the migration priority group of page 0 is  $mpg_2$  (Step 3). Lastly, because the migration priority of page 0 is 2, WPA inserts page 0 into the DRAM buffer between page 4 and page 12, that is set into  $mru_2$ , the MRU position of migration priority group 2 (Step 4).

Fig. 4 shows an example of the page allocation in the NVRAM buffer and the scenario of the block eviction from the NVRAM buffer to the flash-based storage. When the host system requires page 8 for a read operation, WPA allocates page 8 into the NVRAM buffer. However, the NVRAM buffer does not have a free block for inserting page 8. Therefore, WPA evicts the block in LRU position of the NVRAM buffer. In this example, WPA selects block 0 which was in LRU as the

TABLE III  
THE PARAMETER IN EXPERIMENT

Parameter	Description	
Page size	4 KB	
Block size	256 KB (64 pages)	
Page read latency	DRAM	10 ns
	NVRAM	10 ns
	SSD	10 $\mu$ s
Page write latency	DRAM	10 ns
	NVRAM	50 ns
	SSD	100 $\mu$ s
Block erase latency	SSD	2 ms

victim block for evicting to the storage (Step 1). At this time, WPA merges DRAM pages, which can be included in the victim block. In this example, page 0 and page 1 are merged in the victim block (Step 2). In this way, WPA converts several random writes to one sequential write. Afterward, WPA inserts the page numbers of write pages into PEL as the evicted order (Step 3), to gather the past write access pattern through the victim block, and WPA evicts the victim block to the storage (Step 4). Lastly, WPA inserts page 8 and moves block 2 to MRU position in the NVRAM buffer (Step 5).

## IV. EVALUATION

In this section, we describe the evaluation environment and discuss the performance enhancement of our proposed approach, comparing it with the performances of existing hybrid disk buffer management policies.

### A. Evaluation Environment

We measured the performance of our hybrid disk buffer management policy using a trace-driven simulator. We developed a simulator based on DiskSim considering the physical characteristics of DRAM and NVRAM [21], [37]. In our experiment, we measured the performance of our proposed policy by changing the hybrid disk buffer size. In our experiments, the size of NVRAM in hybrid disk buffer is set as 4 times of that of DRAM, to provide consistent performance comparison [38], [39]. The rest of parameters for memory and storage are shown in Table III.

As benchmark programs for the performance evaluation, we adopted gaming console, smart TV and smart home hub, as synthetic traces, in real world workloads to consider typical applications used in consumer devices. CAMWEBDEV trace is collected from SNIA and it has characteristics of write-intensive and many I/O requests [40], [41].

### B. Experiments

In this subsection, we present the hit ratio in hybrid disk buffer, the number of accesses to the flash-based storage, the write count in NVRAM, the average response time in consumer devices based on flash system and the block erase count in the flash-based storage. In BPLRU, CBM, CLOCK-DNV, VBBMS and WPA, the performance is generally improved correspondingly to the increase of hybrid

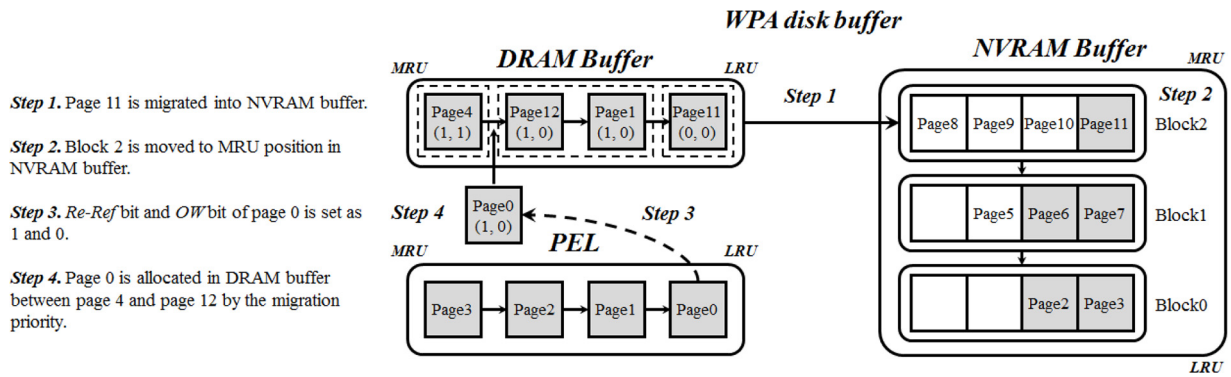


Fig. 3. An example of the page allocation in the DRAM buffer and the page migration from the DRAM buffer to the NVRAM buffer.

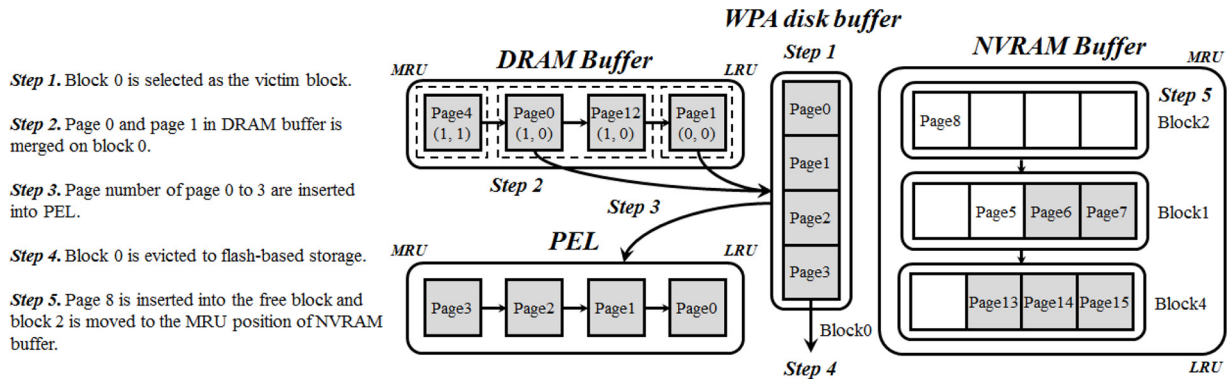


Fig. 4. An example of the page allocation in the NVRAM buffer and the block eviction from the NVRAM buffer to the flash-based storage.

TABLE IV  
THE SPECIFICATION OF WORKLOADS

Workload	Trace name	Read-Write ratio (%)	Average request size (KB) (Read / Write)	Trace size (KB)
	Gaming console	65 : 35	34 / 10	66,396
Synthetic	Smart TV	68 : 32	28 / 8	14,336
	Smart home hub	23 : 77	32 / 16	17,828
MSR Cambridge	CAMWEBDEV	8 : 92	21 / 17	36,491

disk buffer size. However, each policy shows their own performance depending on characteristics of traces.

Fig. 5 shows the hit ratio in hybrid disk buffers, compared to BPLRU, CBM, CLOCK-DNV and VBBMS. The hit ratio in hybrid disk buffers is a key factor to improve the system performance, because higher hit ratio of hybrid disk buffers reduces the number of direct operations to storages, resulting in positive effect for the system performance in terms of access latency, power consumption and lifespan of NAND flash memories. In Fig. 5, the hit ratio in the hybrid disk buffer of WPA is observed from 21.7% and up to 72.1% and the improvement of WPA hit ratio is 87.6% on average, compared to the other policies. It means that WPA efficiently manages page writes and page reads in the hybrid disk buffer. It not only considers the write pattern for DRAM allocation but also utilizes the temporal locality of read operations for NVRAM allocation. Especially the DRAM buffer of WPA effectively holds pages

accessed in the past through the write pattern analysis using PEL.

Fig. 6 shows the average response time in hybrid disk buffer of WPA, normalized by BPLRU. The response time is directly related to the hit ratio of disk buffer and the read/write access counts to flash-based storages. WPA reduces average response time in the hybrid disk buffer up to 58.2%, 31.1%, 37.9% and 42.5%, compared to BPLRU, CBM, CLOCK-DNV and VBBMS, respectively. In addition, WPA reduces the average response time in the hybrid disk buffer based flash storage systems by 36.7%. The response time reduction of WPA comes from the improvement of both the hit ratio in the hybrid disk buffer and the number of direct accesses in flash storage. As a result, due to the efficient disk buffer management of WPA, the response time is accordingly reduced in our proposal.

The primary goal of WPA is to improve the limited lifespan of flash memories. To achieve this, hybrid disk buffers

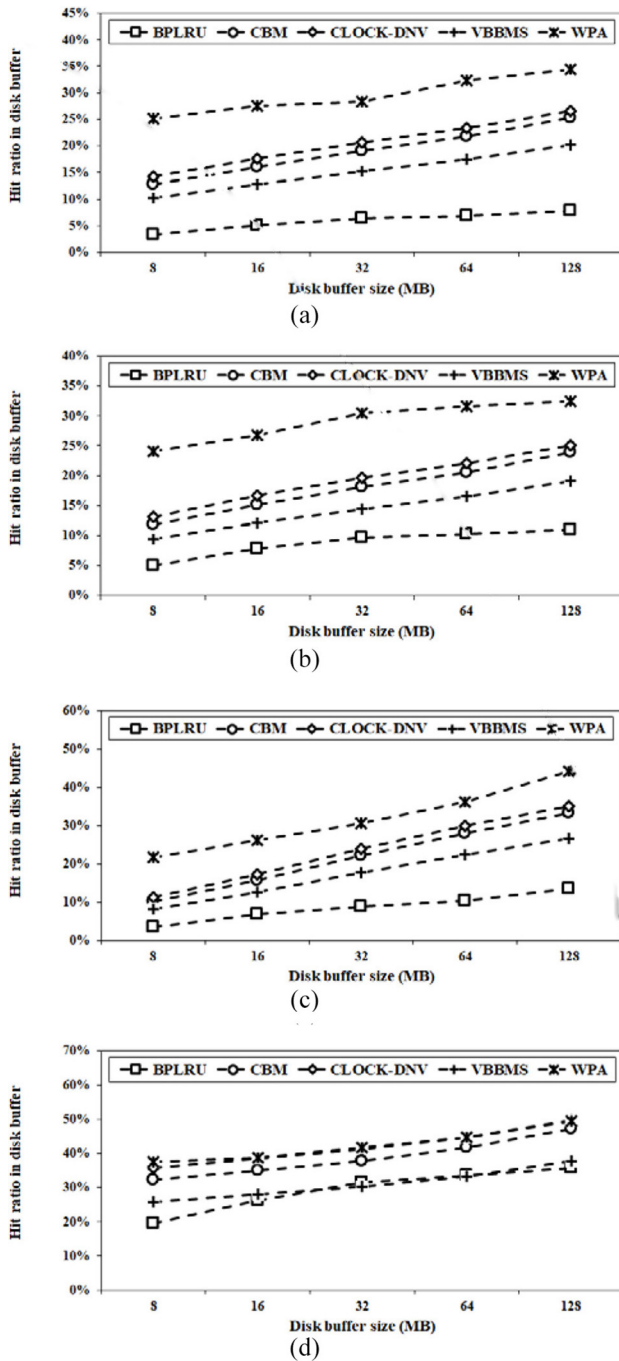


Fig. 5. The hit ratio in disk buffers. (a) Gaming console. (b) Smart TV. (c) Smart home hub. (d) CAMWEBDEV.

should effectively reduce the number of block erase counts in flash memories [13]–[15]. In Fig. 7, the number of block erase counts of WPA is reduced up to 66.8%, 52.5%, 33.5% and 44.6%, compared to BPLRU, CBM, CLOCK-DNV and VBBMS, respectively. On average, WPA reduces the number of block erase counts in flash memories by 38.2%, compared to other policies. WPA efficiently converts frequent random writes to one sequential write in order to reduce the number of block erase counts in flash memories, as frequently occurred random writes to flash-based storages cause the increase of block erase counts in NAND flash memories. When the victim

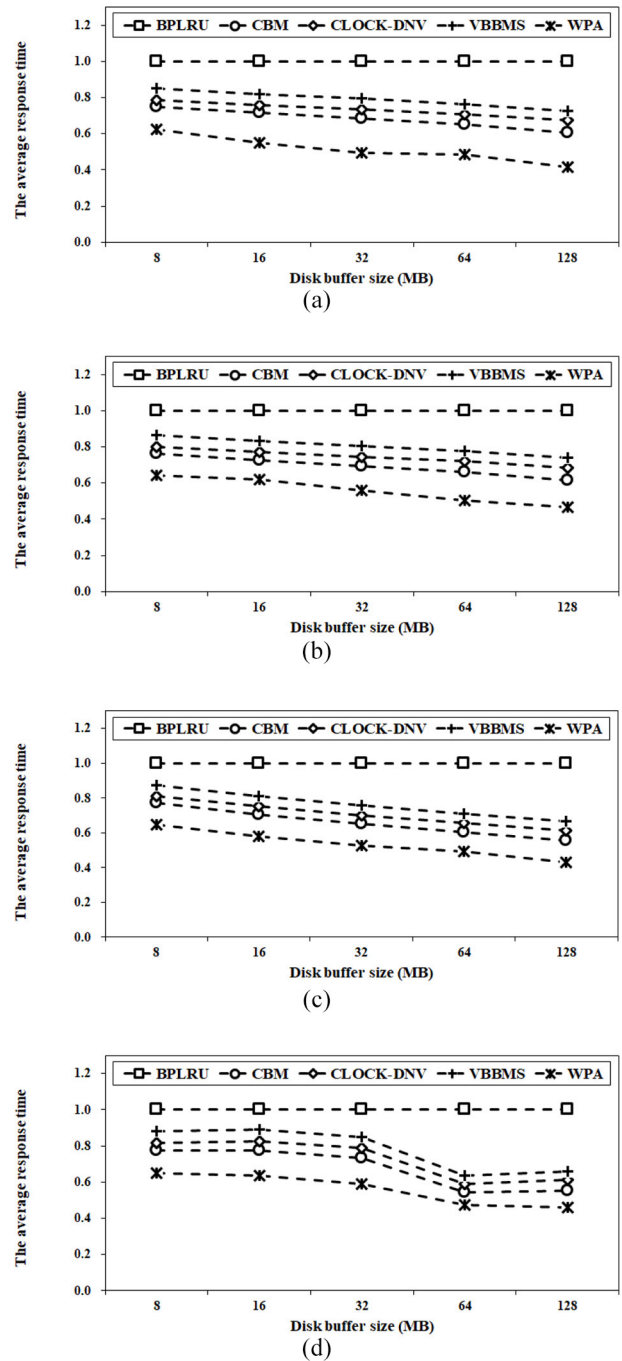


Fig. 6. The normalized average response times. (a) Gaming console. (b) Smart TV. (c) Smart home hub. (d) CAMWEBDEV.

block is evicted to storages, WPA merges buffer pages in DRAM into this victim block, for converting several random writes to one sequential write. As a result, WPA can reduce the number of block erase counts in flash-based storages, resulting in the improvement of the limited lifespan of NAND flash memories.

## V. CONCLUSION

Today, most consumer electronics devices have used NAND flash memory as their main storage. However, flash-based storages used in consumer devices have disadvantages which



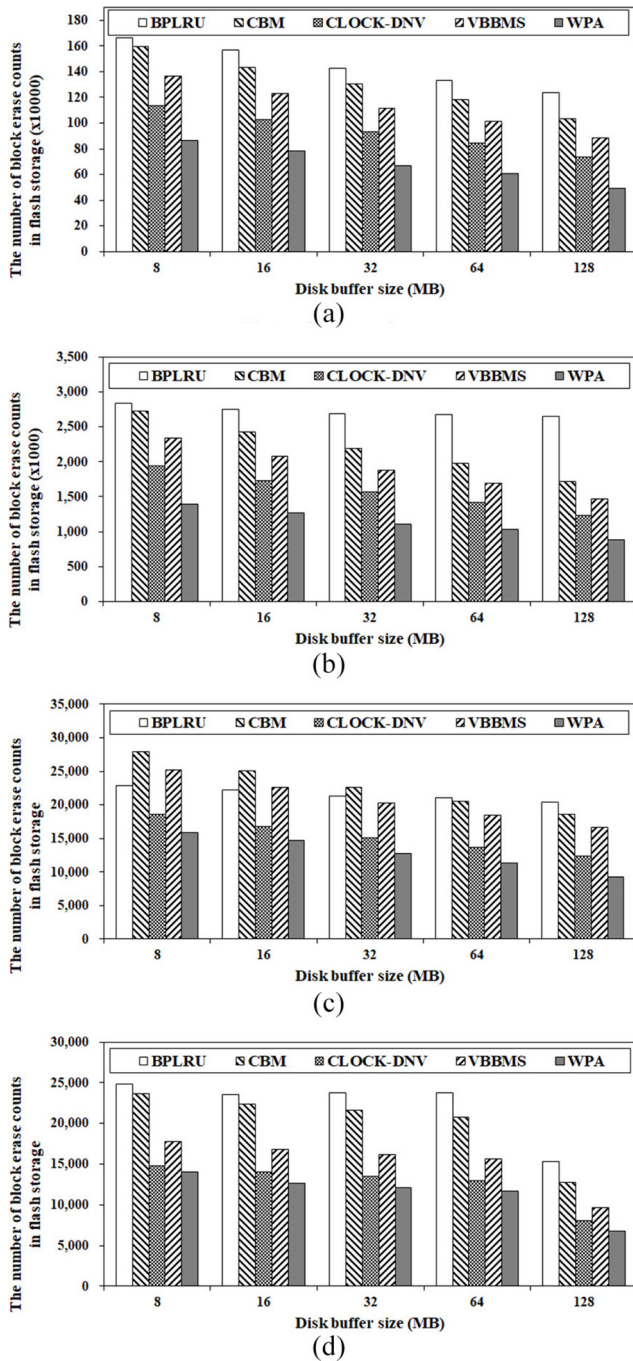


Fig. 7. The number of block erase counts in flash storages. (a) Gaming console. (b) Smart TV. (c) Smart home hub. (d) CAMWEBDEV.

are the limited lifespan and the low latency, and the existing disk buffer management policies have also disadvantages when applying to consumer devices, caused by write operation characteristics in consumer device application that tended to refer again to the same write location. For improving the performance of consumer electronics devices, we proposed WPA, the hybrid buffer management policy for consumer electronics devices, which uses write pattern analysis through the precise trace of the past and current locality. WPA uses PaMP-LRU algorithm and PEL of WPA manages the page number of evicted write pages. Owing to the write pattern

analysis through PaMP-LRU with PEL, our proposal can effectively trace previous write patterns and predict proper locations in disk buffer. As a result, WPA allows improving the lifespan and the latency of flash-based storage in consumer electronics through reducing the number of access counts and block erase counts in NAND flash memory. Evaluation results confirm that the performance of WPA is better than that of other policies, especially in the write-intensive workloads. WPA improves the hit ratio 87.6% on average, compared to BPLRU, CBM, CLOCK-DNV and VBBMS. In addition, WPA reduces the number of accesses in flash-based storage, the write count in NVRAM and the average response time in disk buffer 32.9%, 35.2% and 36.7% on average, respectively. Finally, WPA improves the lifespan of NAND flash memory, 38.2% on average, up to 66.8%, compared to other policies. Through these evaluation results, WPA can enhance response time, energy consumption and lifespan of consumer electronics devices and the consumers can directly feel the performance differences, by considering WPA in disk buffer to help select their actual consumer devices, such as device response time, battery duration time, and product lifespan.

## REFERENCES

- [1] H.-S. Lee, S. Park, and D.-H. Lee, "RMSS: An efficient recovery management scheme on NAND flash memory based solid state disk," *IEEE Trans. Consum. Electron.*, vol. 59, no. 1, pp. 107–112, Feb. 2013, doi: [10.1109/TCE.2013.6490248](https://doi.org/10.1109/TCE.2013.6490248).
- [2] D. Seo and D. Shin, "Recently-evicted-first buffer replacement policy for flash storage devices," *IEEE Trans. Consum. Electron.*, vol. 54, no. 3, pp. 1228–1235, Aug. 2008, doi: [10.1109/TCE.2008.4637611](https://doi.org/10.1109/TCE.2008.4637611).
- [3] R. Jin, H.-J. Cho, and T.-S. Chung, "Three-state log-aware buffer management scheme for flash-based consumer electronics," *IEEE Trans. Consum. Electron.*, vol. 59, no. 4, pp. 795–802, Nov. 2013, doi: [10.1109/TCE.2013.6689691](https://doi.org/10.1109/TCE.2013.6689691).
- [4] J. Park, E. Lee, and H. Bahn, "DABC-NV: A buffer cache architecture for mobile systems with heterogeneous flash memories," *IEEE Trans. Consum. Electron.*, vol. 58, no. 4, pp. 1237–1245, Nov. 2012, doi: [10.1109/TCE.2012.6414991](https://doi.org/10.1109/TCE.2012.6414991).
- [5] R. Chen and M. Lin, "Energy-aware buffer management scheme for NAND and flash-based consumer electronics," *IEEE Trans. Consum. Electron.*, vol. 61, no. 4, pp. 484–490, Nov. 2015, doi: [10.1109/TCE.2015.7389803](https://doi.org/10.1109/TCE.2015.7389803).
- [6] B.-K. Kim and D.-H. Lee, "LSF: A new buffer replacement scheme for flash memory-based portable media players," *IEEE Trans. Consum. Electron.*, vol. 59, no. 1, pp. 130–135, Feb. 2013, doi: [10.1109/TCE.2013.6490251](https://doi.org/10.1109/TCE.2013.6490251).
- [7] C. Du, Y. Yao, J. Zhou, and X. Xu, "VBBMS: A novel buffer management strategy for NAND flash storage devices," *IEEE Trans. Consum. Electron.*, vol. 65, no. 2, pp. 134–141, May 2019, doi: [10.1109/TCE.2019.2910890](https://doi.org/10.1109/TCE.2019.2910890).
- [8] J. Cui, W. Wu, Y. Wang, and Z. Duan, "PT-LRU: A probabilistic page replacement algorithm for NAND flash-based consumer electronics," *IEEE Trans. Consum. Electron.*, vol. 60, no. 4, pp. 614–622, Nov. 2014, doi: [10.1109/TCE.2014.7027334](https://doi.org/10.1109/TCE.2014.7027334).
- [9] F. Chen, T. Luo, and X. Zhang, "CAFTL: A content-aware flash translation layer enhancing the lifespan of flash memory based solid state drives," in *Proc. USENIX Conf. File Stor. Technol. (FAST)*, San Jose, CA, USA, 2011, pp. 77–90, doi: [10.5555/1960475.1960481](https://doi.org/10.5555/1960475.1960481).
- [10] J.-Y. Paik, E.-S. Cho, R. Jin, and T.-S. Chung, "Selective-delay garbage collection mechanism for read operations in multichannel flash-based storage devices," *IEEE Trans. Consum. Electron.*, vol. 64, no. 1, pp. 118–126, Feb. 2018, doi: [10.1109/TCE.2018.2812062](https://doi.org/10.1109/TCE.2018.2812062).
- [11] Y.-H. Chang, J.-W. Hsieh, and T.-W. Kuo, "Improving flash wear-leveling by proactively moving static data," *IEEE Trans. Comput.*, vol. 59, no. 1, pp. 53–65, Jan. 2010, doi: [10.1109/TC.2009.134](https://doi.org/10.1109/TC.2009.134).
- [12] L.-P. Chang, T.-Y. Chou, and L.-C. Huang, "An adaptive, low-cost wear-leveling algorithm for multichannel solid-state disks," *ACM Trans. Embedded Comput. Syst.*, vol. 13, no. 3, p. 55, Dec. 2013, doi: [10.1145/2539036.2539051](https://doi.org/10.1145/2539036.2539051).

- [13] S. H. Kim and J. W. Kwak, "Garbage collection technique using erasure interval for NAND flash memory-based storage systems," *Int. J. Appl. Eng. Res. (IJAER)*, vol. 11, pp. 5188–5194, Apr. 2016, doi: [10.9708/IJKSCL.2017.22.04.025](https://doi.org/10.9708/IJKSCL.2017.22.04.025).
- [14] C. Gao *et al.*, "Exploiting chip idleness for minimizing garbage collection-induced chip access conflict on SSDs," *ACM Trans. Design Autom. Electron. Syst.*, vol. 23, no. 2, pp. 15–43, Jan. 2018, doi: [10.1145/3131850](https://doi.org/10.1145/3131850).
- [15] T.-S. Chung, D.-J. Park, S. Park, D.-H. Lee, S.-W. Lee, and H.-J. Song, "A survey of flash translation layer," *J. Syst. Archit. Embedded Syst. Design*, vol. 55, nos. 5–6, pp. 332–343, May 2009, doi: [10.1016/j.jsysarc.2009.03.005](https://doi.org/10.1016/j.jsysarc.2009.03.005).
- [16] H. Kim and S. Ahn, "BPLRU: A buffer management scheme for improving random writes in flash storage," in *Proc. USENIX Conf. File Stor. Technol. (FAST)*, San Jose, CA, USA, 2008, pp. 239–252, doi: [10.5555/1364813.1364829](https://doi.org/10.5555/1364813.1364829).
- [17] Q. Wei, C. Chen, and J. Yang, "CBM: A cooperative buffer management for SSD," in *Proc. Symp. Mass Stor. Syst. Technol. (MSST)*, San Jose, CA, USA, 2014, pp. 1–12, doi: [10.1109/MSST.2014.6855545](https://doi.org/10.1109/MSST.2014.6855545).
- [18] D. H. Kang, S. J. Han, Y.-C. Kim, and Y. I. Eom, "CLOCK-DNV: A write buffer algorithm for flash storage devices of consumer electronics," *IEEE Trans. Consum. Electron.*, vol. 63, no. 1, pp. 85–91, Feb. 2017, doi: [10.1109/TCE.2017.014700](https://doi.org/10.1109/TCE.2017.014700).
- [19] J. Boukhobza, S. Rubini, R. Chen, and Z. Shao, "Emerging NVM: A survey on architectural integration and research challenges," *ACM Trans. Design Autom. Electron. Syst.*, vol. 23, no. 2, pp. 14–45, Jan. 2018, doi: [10.1145/3131848](https://doi.org/10.1145/3131848).
- [20] S. Yu and P.-Y. Chen, "Emerging memory technologies: Recent trends and prospects," *IEEE Solid-State Circuits Mag.*, vol. 8, no. 2, pp. 43–56, Jun. 2016, doi: [10.1109/MSSC.2016.2546199](https://doi.org/10.1109/MSSC.2016.2546199).
- [21] X. Dong, C. Xu, Y. Xie, and N. P. Jouppi, "NVSim: A circuit-level performance, energy, and area model for emerging nonvolatile memory," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 31, no. 7, pp. 994–1007, Jul. 2012, doi: [10.1109/TCAD.2012.2185930](https://doi.org/10.1109/TCAD.2012.2185930).
- [22] M. K. Qureshi, V. Srinivasan, J. A. Rivers, "Scalable high performance main memory system using phase-change memory technology," in *Proc. Int. Symp. Comput. Archit. (ISCA)*, Austin, TX, USA, 2009, pp. 24–33, doi: [10.1145/1555754.1555760](https://doi.org/10.1145/1555754.1555760).
- [23] T. Endoh, H. Koike, S. Ikeda, T. Hanyu, and H. Ohno, "An overview of nonvolatile emerging memories-spintronics for working memories," *IEEE J. Emerg. Sel. Topics Circuits Syst.*, vol. 6, no. 2, pp. 109–119, Jun. 2016, doi: [10.1109/JETCAS.2016.2547704](https://doi.org/10.1109/JETCAS.2016.2547704).
- [24] S. Mittal and J. S. Vetter, "A survey of software techniques for using non-volatile memories for storage and main memory systems," *IEEE Trans. Parallel Distrib. Syst.*, vol. 27, no. 5, pp. 1537–1550, May 2016, doi: [10.1109/TPDS.2015.2442980](https://doi.org/10.1109/TPDS.2015.2442980).
- [25] S. Mittal, "A survey of techniques for architecting processor components using domain-wall memory," *ACM J. Emerg. Technol. Comput. Syst. (JETC)*, vol. 13, no. 2, pp. 29–53, Mar. 2016, doi: [10.1145/2994550](https://doi.org/10.1145/2994550).
- [26] S. Kim, S.-H. Hwang, and J. W. Kwak, "Adaptive-classification clock: Page replacement policy based on read/write access pattern for hybrid DRAM and PCM main memory," *Microprocessors Microsyst.*, vol. 57, pp. 65–75, Mar. 2018, doi: [10.1016/J.MICPRO.2018.01.003](https://doi.org/10.1016/J.MICPRO.2018.01.003).
- [27] S. Lee, H. Bahn, and S. H. Noh, "CLOCK-DWF: A write-history-aware page replacement algorithm for hybrid PCM and DRAM memory architectures," *IEEE Trans. Comput.*, vol. 63, no. 9, pp. 2187–2200, Sep. 2014, doi: [10.1109/TC.2013.98](https://doi.org/10.1109/TC.2013.98).
- [28] Y.-J. Lin, C.-L. Yang, H.-P. Li, and C.-Y. M. Wang, "A buffer cache architecture for smartphones with hybrid DRAM/PCM memory," in *Proc. IEEE Non Volatile Memory Syst. Appl. Symp. (NVMSA)*, Hong Kong, 2015, pp. 1–6, doi: [10.1109/NVMSA.2015.7304363](https://doi.org/10.1109/NVMSA.2015.7304363).
- [29] B. Tallis. (2018). *The Lastest High-Capacity M.2: The Samsung 860 EVO 2TB SSD, Reviewed*. Accessed: Jan. 8, 2020. [Online]. Available: <https://www.aNANDtech.com/show/12408/the-samsung-860-evo-m2-2tb-ssd-review>
- [30] B. Tallis. (2019). *The Adata Ultimate SU750 1TB SSD Review: Realtek Does Storage, Part1*. Accessed: Jan. 8, 2020. [Online]. Available: <https://www.aNANDtech.com/show/15138/the-adata-ultimate-su750-1tb-ssd-review>
- [31] B. Tallis. (2018). *The Crucial MX500 500GB SSD Review: Asecond Look*. Accessed: Jan. 8, 2020. [Online]. Available: <https://www.aNANDtech.com/show/12263/the-crucial-mx500-500gb-review>
- [32] B. Tallis. (2017). *Toshiba Announces TR200 Retail SATA SSDS With 3D NAND*. Accessed: Jan. 8, 2020. [Online]. Available: <https://www.aNANDtech.com/show/11661/toshiba-announces-tr200-retail-ssds-with-3d-nand>
- [33] L. Xia, M. Liu, X. Ning, K. Chakrabarty, and Y. Wang, "Fault-tolerant training with on-line fault detection for RRAM-based neural computing systems," in *Proc. Annu. Design Autom. Conf. (DAC)*, Austin, TX, USA, 2017, pp. 1–6, doi: [10.1145/3061639.3062248](https://doi.org/10.1145/3061639.3062248).
- [34] W. Huangfu *et al.*, "Computation-oriented fault-tolerance schemes for RRAM computing systems," in *Proc. Asia South Pac. Design Autom. Conf. (ASP-DAC)*, Chiba, Japan, 2017, pp. 794–799, doi: [10.1109/ASPDAC.2017.7858421](https://doi.org/10.1109/ASPDAC.2017.7858421).
- [35] C. C. Hsieh, Y. F. Chang, Y. Jeon, A. Roy, D. Shahrjerdi, and S. K. Banerjee, "Short-term relaxation in HFOX/CEOX resistive random access memory with selector," *IEEE Electron Device Lett.*, vol. 38, no. 7, pp. 871–874, Jul. 2017, doi: [10.1109/LED.2017.2710955](https://doi.org/10.1109/LED.2017.2710955).
- [36] Y. F. Chang *et al.*, "Intrinsic SiOx-based unipolar resistive switching memory. I. oxide stoichiometry effects on reversible switching and program window optimization," *J. Appl. Phys.*, vol. 116, Jul. 2014, Art. no. 043708, doi: [10.1063/1.4891242](https://doi.org/10.1063/1.4891242).
- [37] J. S. Bucy and G. R. Ganger. (2009). *The DiskSim Simulation Version 4.0*. [Online]. Available: <http://www.pdl.cmu.edu/DiskSim>
- [38] X. Cai, L. Ju, M. Zhao, Z. Sun, and Z. Jia, "A novel page caching policy for PCM and DRAM of hybrid memory architecture," in *Proc. Int. Conf. Embedded Softw. Syst. (ICCESS)*, Chengdu, China, 2016, pp. 67–73, doi: [10.1109/ICCESS.2016.17](https://doi.org/10.1109/ICCESS.2016.17).
- [39] J. Hu, Q. Zhuge, C. J. Xue, W.-C. Tseng, and E. H.-M. Sha, "Software enabled ware-leveling for hybrid PCM main memory on embedded systems," in *Proc. Design Autom. Test Europe Conf. Exhibit.*, Grenoble, France, 2013, pp. 599–602, doi: [10.7873/DATE.2013.131](https://doi.org/10.7873/DATE.2013.131).
- [40] D. Narayanan, A. Donnelly, and A. Rowstron, "Write off-loading: Practical power management for enterprise storage," *ACM Trans. Stor.*, vol. 4, no. 3, pp. 253–267, Nov. 2008, doi: [10.1145/1416944.1416949](https://doi.org/10.1145/1416944.1416949).
- [41] (2011). *SNIA IOTTA Repository*. [Online]. Available: <http://iotta.snia.org/traces/130>



**Jun-Hyeong Choi** received the B.S. degree from the Department of Computer Engineering, Yeungnam University College, Daegu, South Korea, in 2013, and the M.S. degree from the Department of Computer Engineering from Yeungnam University, Gyeongsan, South Korea, in 2016, where he is currently pursuing the Ph.D. degree. His current research interests include memory and storage system design in consumer electronics and high performance distributed systems.



**Kyung Min Kim** received the B.S. and M.S. degree from the Department of Computer Engineering, Yeungnam University, Gyeongsan, South Korea, in 2017 and 2019, respectively, where he is currently pursuing the Ph.D. degree. His current research interests include software architectures in consumer electronics and embedded systems from nonvolatile memory aspect.



**Jong Wook Kwak** received the B.S. degree in computer engineering from Kyungpook National University, Daegu, South Korea, in 1998, and the M.S. degree in computer engineering and the Ph.D. degree in electrical engineering and computer science from Seoul National University, Seoul, South Korea, in 2001 and 2006, respectively. From 2006 to 2007, he was a Senior Engineer with the System-on-Chip (SoC) Research and Development Center, Samsung Electronics Company, Ltd., Suwon, South Korea. From 2012 to 2013, he was a Visiting

Scholar with the Georgia Institute of Technology, Atlanta, GA, USA. From 2018 to 2019, he was a Visiting Scholar with Arizona State University, Tempe, AZ, USA. He is currently a Professor with the Department of Computer Engineering, Yeungnam University, Gyeongsan, South Korea. His current research interests include consumer electronics system design, advanced processor architecture, low-power mobile embedded system design, and high-performance parallel and distributed computing.