2010 International Workshop on Storage Network Architecture and Parallel I/Os

NAND Flash-based Disk Cache Using SLC/MLC Combined Flash Memory

Seongcheol Hong
School of Information and Communication Engineering
Sungkyunkwan University
Suwon, Korea
adonis0101@skku.edu

Dongkun Shin
School of Information and Communication Engineering
Sungkyunkwan University
Suwon, Korea
dongkun@skku.edu

***Abstract*—Flash memory-based non-volatile cache (NVC) is emerging as an effective solution for enhancing both the performances and the energy consumptions of storage systems. In order to attain significant performance and energy gains from NVC, it would be better to use multi-level-cell (MLC) flash memories since they can provide a large NVC capacity at low cost. However, the number of available program/erase cycles of MLC flash memory is smaller than that of single-level-cell (SLC) flash memory, which limits the lifespan of an NVC. In order to overcome this limitation, SLC/MLC combined flash memory is a promising solution for use in NVC. This paper proposes an effective management scheme for heterogeneous SLC and MLC regions of combined flash memory. It also proposes a design technique which is able to determine the optimal proportion between the two regions that maximizes performance and energy reduction, guaranteeing the lifespan constraint. We show experimentally how performance, lifespan, and energy consumption of the NVC-embedded hard disk change depending upon the configuration of the combined flash memory. We also show the superiority of the proposed NVC management policy in comparison to alternative policies.**

***Keywords*-flash memory; hybrid storage;**

I. Introduction

NAND flash memory has many advantages over hard disk drives, such as high random access performance, low-power consumption, small size, and high shock resistance. However, it has mainly been used for mobile consumer devices such as MP3 players, digital cameras, personal digital assistants, and cell phones due to its high cost. The recent dramatic price reduction of flash memory makes it possible to use a flash memory-based solid-state disk (SSD) for general purpose computing systems such as desktop PCs and enterprise servers.

While the main advantages of SSD are its low power consumption, high reliability and high random access performance, the main disadvantage is its high cost in comparison to that of a magnetic hard disk drive. The complementary features of NAND flash memory and hard disks have motivated several proposals on flash memory-based non-volatile cache (NVC), which stores data blocks that are likely to be accessed in the near future, allowing the hard disk to spin down for longer periods. There are two kinds of architectures for NVCs: 1) hybrid hard disk drive (HDD) with an embedded NVC [1], and 2) on-board NVC added to a host system without requiring changes to the existing disk drives [2]. Since both of the two NVC architectures have the same goal of improving the I/O performance of storage systems with a higher energy efficiency, we targeted to the hybrid HDD to simplify the descriptions.

When designing an NVC, several characteristics and limitations of flash memory should be considered. First, flash memory does not provide an "in-place" update, which means that a block needs to be erased before data can be written. While the write operation is performed by the page unit, the erase operation should be performed by the block unit, composed of several pages. In order to prevent frequent erase operations, updated data is generally written at other flash pages, and the old data is invalidated. Second, the lifespan of flash memory is limited by its maximum program/erase (P/E) cycles. Therefore, if the NVC of the hybrid HDD exhausts its P/E cycles, the hybrid HDD will act like a normal HDD.

There are two types of flash memories: single-level-cell (SLC) and multi-level-cell (MLC). While one flash memory cell represents one bit in SLC flash memory, two or more bits can be represented using multiple voltage thresholds in MLC flash memory [3]. For two-bit MLC, one cell retains two bits for two paired pages, thus one cell can be programmed twice for the least significant bit (LSB) and the most significant bit (MSB). MLC flash is cheaper than SLC flash because it provides a larger storage capacity than does SLC flash for the same-sized die. Therefore, MLC flash is a better solution for large-scale flash memory systems such as SSD or NVC for hybrid HDD. However, MLC flash memory is slower, less reliable, and has a smaller number of P/E cycles in comparison to SLC flash memory. MLC flash performance is about half that of SLC performance, and the available P/E cycles of MLC is about one-fifth that of SLC. Its poor performance and short lifespan are critical obstacles in the wide adoption of MLC.

As a compromised solution to overcome the limitations of SLC and MLC flash memories, SLC/MLC combined flash is considered. There are two combined architectures, one which uses both SLC flash chips and MLC flash chips to construct a large scale flash storage [4]. This scheme is undesirable for the NVC of a hybrid HDD since the

978-0-7695-4025-2/10 $26.00 © 2010 IEEE
DOI 10.1109/SNAPI.2010.11

embedded flash chip should occupy only a small space within the disk drive. The other, more flexible approach is to use an SLC/MLC combined flash chip, which has both SLC blocks and MLC blocks in a single chip. By programming only the LSB of a cell in the MLC flash memory, the cell can be used as an SLC [5], [6]. By programming only LSBs into a particular block (i.e., SLC mode), the effective properties of that block become similar to those of an SLC flash memory block. Conversely, if both the LSB and MSB are programmed (i.e., MLC mode), the high capacity provided by the MLC flash memory is able to be utilized. Therefore, the SLC/MLC combined flash provides dual modes, SLC mode and MLC mode. The dual mode programming allows two different types of blocks to exist simultaneously in the same flash chip. The MLC block using SLC mode has a shorter write latency than does the MLC block using MLC mode.

For the SLC/MLC combined flash memory chip, the flash memory blocks can be divided into two regions for ease of management, an SLC region and an MLC region [7]. Depending on the size of each region, the total storage capacity of the flash memory is determined. For example, if there are 1024 blocks in a flash memory chip where half of the blocks use the SLC mode and the other half of the blocks use the MLC mode, then the total capacity is 393 MB (i.e., 131 MB + 262 MB) when the block sizes in SLC mode and MLC mode are 256 KB and 512 KB, respectively. Therefore, as we increase the number of MLC mode blocks, the total size of the NVC also increases.

The proportion of the SLC region size to the MLC region size needs to be carefully determined since it directly affects performance, lifespan, and energy consumption of a hybrid HDD. The performance and energy consumption of a hybrid HDD depend upon the miss rate of the NVC since the miss penalty (i.e., access cost of HDD) is significantly higher than the access cost of flash memory. Generally, as the size of the NVC becomes larger, the miss rate decreases, therefore, a smaller number of disk drive accesses are invoked causing high performance and low energy consumption. However, in order to implement a large capacity NVC, more blocks should be used in MLC mode, which will reduce the lifespan of the NVC.

To exploit the SLC/MLC combined flash for NVC, an effective management scheme is required for the heterogeneous flash memory regions. The goal is to produce a longer lifespan for the SLC/MLC combined NVC compared to that of an MLC-only NVC, while providing a higher performance and lower energy consumption than those of an SLC-only NVC. More specifically, we must determine how the SLC and MLC regions are managed in order to maximize the advantage of the SLC/MLC combined NVC and to minimize hard disk accesses. We must also determine the optimal proportion of the SLC region size to the MLC region size for the total number of blocks of SLC/MLC combined NVC as well as given target workloads.

This paper proposes a two-level NVC management scheme for SLC/MLC combined flash memory that effectively exploits the characteristics of the heterogeneous regions. The scheme utilizes the SLC region as a first-level buffer and the MLC region as a second-level buffer. The proposed scheme is compared with other possible alternative schemes. We also provide a design technique which can identify the optimal fraction of the SLC region within the NVC. The design scheme is based on the evaluation on three metrics, i.e., performance, lifespan and energy consumption of the hybrid HDD while varying the fraction of SLC region.

From simulation-based evaluations using various synthetic and real workloads, we show that performance, lifespan, and energy consumption of hybrid HDD can change significantly depending upon the proportion of each region.

The rest of the paper is organized as follows. Section 2 introduces related works and their drawbacks, and Section 3 describes the overall storage architecture for hybrid HDD. Section 4 explains the garbage collection technique for each region within the NVC. Section 5 describes the design technique used to find the optimal partition of the SLC and MLC regions. Section 6 presents the experimental results, and Section 7 concludes this paper with a summary and description of future works.

II. Related Works

A. Hybrid HDD

There are several researches on NVC management for hybrid HDD. Bisson et al. proposed an I/O redirection algorithm [8], NVC flushing polices [9], and disk spin-down algorithms [10] for hybrid HDD. The SmartSaver technique [11] partitioned the NVC into a caching area, prefetching area, and writeback area and then proposed the caching and prefetching techniques. The caching technique selects cache-critical data which can significantly extend disk idle time if it is in the NVC. Hsieh et al. [12] proposed a hash-based lookup strategy, an LRU replacement policy, and an energy-efficient flushing strategy for NVC. If the disk is spinning down, all data is kept in the cache whenever possible. Their scheme also divides the NVC into primary and overflow blocks in order to efficiently handle the flash memory. Kim et al. [13] proposed a data pinning policy that can provide fast access times for hot data, and the pinning technique can also be applied to reduce the system boot time as well as the application launching time. The EXCES technique [14] identifies the hot data to be prefetched, cached and buffered in order to increase disk inactivity periods. It tracks the popularity of the page accessed by applications and continuously adapts to workload changes. However, the technique does not address the management scheme for flash memory, such as garbage collection.

While these previous works consider only single-type flash memory, the paper by Kgil et al. [15] focuses on

hybrid HDD using SLC/MLC combined flash memory. They proposed a density control technique which dynamically switches every page from MLC to SLC. When the working set size is small, MLC pages are changed into SLC pages in order to improve the latency of flash storage, assuming that the increased miss rate due to a reduction in density is small. However, if the SLC block is reconfigured into an MLC block, it is difficult to estimate the remaining P/E cycles of the block, which is required for wear-leveling. The flash memory chip venders do not provide detailed test results for block reconfiguration because it is impossible to test all of the potential reconfigurations. The venders provide only the available P/E cycles of the SLC and MLC blocks, assuming no reconfiguration. Therefore, this technique is impractical.

Moreover, depending upon the HDD access cost relative to that of flash memory, a small amount of change in the miss rate of NVC may invoke a large difference in the overall performance, as we will demonstrate in the experiments. One proposed effective strategy is to split the NVC into read and write regions, since only the write region generates invalid pages, the garbage collection will require only a small overhead.

In comparison to the technique proposed in [15], we assume a static partition between the SLC region and the MLC region. We focus on how to determine the proportions of the two regions to optimize performance, lifespan, and energy consumption at design time.

B. SLC/MLC Combined Flash Memory

Several studies have been performed on SLC/MLC combined flash memory. Park et al. [16] proposed a Flash Translation Layer (FTL) called MixedFTL for an SLC/MLC combined flash chip. The MixedFTL sends all write requests to the SLC region and then moves the cold data into the MLC region if there are no updates over a long time period. Since it allocates an SLC mode block for each logical block (i.e., block-level mapping), the SLC mode blocks have low utilizations and thus invoke frequent garbage collection.

Lee et al. [17] proposed a file system called FlexFS for SLC/MLC combined flash memory, which determines the target region for a file depending on the storage sizes of the SLC and the MLC regions, rather than the hotness of the file. That is, if there are many free spaces, it sends most of the file write requests to the SLC region even though the file will not be frequently updated. FlexFS exploits the hotness/coldness of the file only to determine whether or not to move it into the MLC region. Similar to the technique in [15], MixedFTL and FlexFS are unrealistic solutions since they assume that the block mode can be reconfigured at run time.

Our previous work [18] proposed a management technique for SLC/MLC combined flash memory. However, it focused on the management of SLC/MLC combined flash memory as an independent storage device. This paper targets the combined flash memory used for NVC embedded within a hybrid HDD, while considering the affects on hard disk behavior.

III. Overall Architecture

Fig. 1 shows the proposed overall storage architecture. The flash chip, targeting to hybrid HDD rather than an on-board NVC, is located between the internal DRAM buffer and the hard disk. The flash chip is divided into an SLC region and an MLC region. If the combined flash memory is used for on-board NVC, it should be located below the buffer cache of the host system in the storage hierarchy.

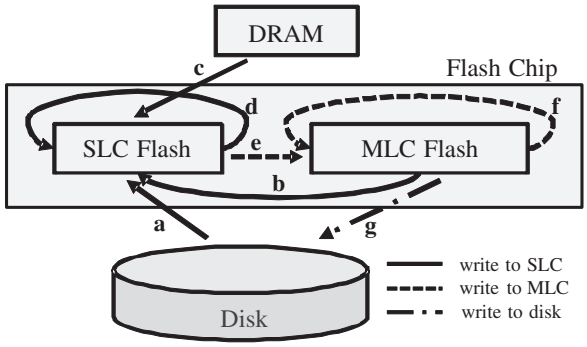


Figure 1. Overall architecture of the proposed hybrid HDD.

For read requests from the host, the system searches the requested data in the DRAM buffer, flash memory (NVC), and hard disk in the order of hierarchy. If the data is found in the flash memory or hard disk, it is copied into the DRAM buffer in order to provide shorter latency for the following read requests. When the data is found in the MLC region or the hard disk, it is also copied into the SLC region (i.e., the arrows marked by **a** and **b**).

For write requests from the host, all data is first written to the DRAM buffer and is sent to the SLC region by a replacement policy of the DRAM buffer (**c**). The data evicted from the DRAM buffer may be written directly to the hard disk if the disk status is spin-up, as proposed in [11]. However, we only consider the case in which all the write requests are sent to the NVC, while the disk status-aware redirection algorithm can be applied to our scheme.

The update requests for the data in the NVC will generate several invalid pages since the flash memory does not support the update operation. If the SLC region needs more free space, the garbage collection (GC) for the SLC region is invoked in order to reclaim the invalid pages. The GC first finds the victim block and then moves valid pages of the block into another space. The block can be reused for future write requests by erasing the victim block after valid page migrations. The GC moves the valid data into other blocks in the SLC region (**d**) or the MLC region (**e**). The MLC region GC moves the valid data within the MLC region (**f**) or into the hard disk (**g**).

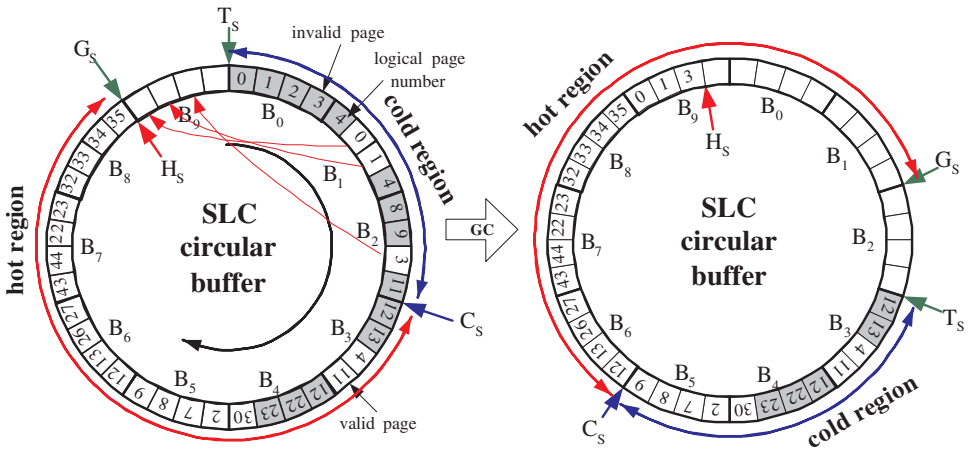


Figure 2. Migration within the SLC region.

Consequently, the SLC region and the MLC region contain hot data and cold data, respectively. Therefore, we can consider the SLC region and the MLC region as a first-level NVC and a second-level NVC, respectively. This scheme can exploit the higher performance and endurance of the SLC region.

IV. Garbage Collection

The SLC and MLC regions are maintained as circular buffers, which are useful in handling flash memory pages as the circular buffers simplify the garbage collection. Each circular buffer has four pointers: a tail pointer (T_S or T_M), a head pointer (H_S or H_M), a cold pointer (C_S or C_M), and a GC pointer (G_S or G_M). Figures 2 and 3 show the garbage collection on an SLC region with 10 blocks ($B_0 \sim B_9$), each of which has four pages. The tail pointer of the circular buffer points to the oldest page, while the head pointer points to the youngest page. For a write request, the data is appended to the page pointed to by the head pointer. If the write request updates the data which is already written at the NVC, its old version becomes invalidated.

The GC pointer for each region is located prior to the tail pointer in the circular buffer structure in such a way that the number of free pages between the GC pointer and the tail pointer is 10% of the total pages within the region. This is to reserve free space that garbage collector uses to move valid pages. When the head pointer becomes the same as the GC pointer, the GC is then invoked. In order to reclaim invalid pages, the GC erases all the blocks in the cold region which is located between the tail pointer (T_S) and the cold pointer (C_S). The cold pointer is located after the tail pointer in the SLC region in such a way that the number of pages between T_S and C_S is 30% of the number of pages between T_S and G_S. In the MLC region, the size of the cold region is 40% of the number of pages between T_M and G_M.

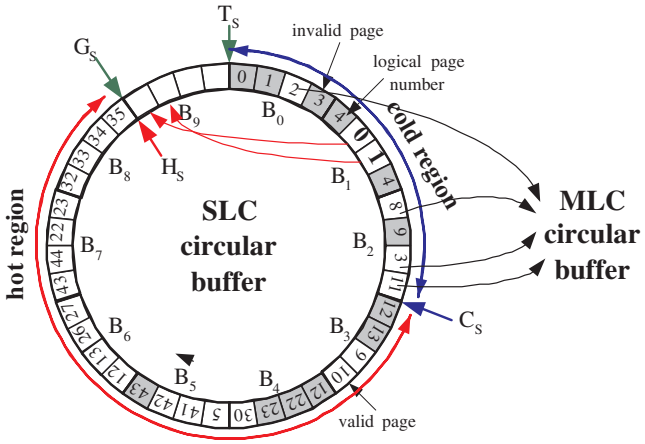


Figure 3. Migration to the MLC region.

The size of the cold region within the MLC region is larger than that of the SLC region since the MLC region GC can invoke write requests on the hard disk. It is probable that the hard disk is in the spin-down state due to the NVC when the GC sends write requests to the hard disk. Then, there is an additional overhead for spinning-up the hard disk. Therefore, once the GC is invoked within the MLC region, it is better to move as many pages as possible to the hard disk so as to amortize the overhead.

The cold region in the circular buffer may have many invalid pages. The valid pages in the cold region can be regarded as cold data since they are not updated until the corresponding flash blocks are inserted into the cold region. If there are only a small number of valid pages within the cold region, the valid pages are then moved into other blocks within the SLC region, as shown in Fig. 2, in which pages 0, 1 and 3 are moved into bock B_9. This provides the cold pages with more of a chance to be in the SLC region

if sufficient free space can be obtained through garbage collection. However, if there are too many valid pages within the cold region, they migrate to the MLC region, as shown in Fig. 3. In order to increase the read hit ratio of the SLC region, a cold page migrates within the SLC region if it is marked with read hit (the pages 0 and 1) even when the number of valid pages is large.

Fig. 4 shows the detailed GC algorithms. If the number of invalid pages in the cold region is larger than 70% of the total number of pages, then the valid pages migrate within the SLC region at the threshold ratio of 0.7. If a valid page is marked with read hit and the portion of invalid pages is larger than 0.4, the page is moved within the SLC region, otherwise, it migrates to the MLC region. The read hit mark is set when a page is read, and reset again when it is moved by GC. After garbage collection, the tail pointer and the GC pointer are updated. The cold pointer is also updated to maintain its position relative to the tail and GC pointers.

GC algorithm for SLC region

```
1: if (0.7 ≤ portion of invalid pages in cold region )      then
2:    all valid pages migrate within SLC region;
3: else if  (0.4 ≤ portion of invalid pages in cold region < 0.7 )
4:    only read-hit valid pages migrate within SLC region
         and other valid pages migrate to MLC region;
5: else   /* portion of invalid pages in cold region < 0.4 */
6:    all valid pages migrate to MLC region;
```

GC algorithm for MLC region

```
1: if (HDD is spin-up)   then
2:    if (0.6 ≤ portion of invalid pages in cold region )      then
3:       all valid pages migrate within MLC region;
4:    else
5:       all valid pages are flushed into HDD;
6: else /* HDD is spin-down */
7:    if (0.4 ≤ portion of invalid pages in cold region )
8:       all valid pages migrate within MLC region;
9:    else
10:      all valid pages are flushed into HDD;
```

Figure 4. Garbage collection algorithms.

The threshold ratio that determines migration within the MLC region or migration to the hard disk depends upon the HDD power state. If HDD is in a spin-down status, we use 0.4 for the threshold value, and if it is in a spin-up status, a higher value is used for the threshold ratio in order to send more cold pages to the hard disk. The MLC region uses tighter threshold values than does the SLC region in order to reduce the write requests on the hard disk since the write cost of the hard disk is significantly larger than that of the MLC region.

V. NVC Region Partitioning

In order to use SLC/MLC flash memory for the NVC of a hybrid HDD, the optimal proportions of the SLC region and the MLC region must be determined. For given flash memory parameters (PE_S, PE_M, P and B) and the input workload W_H, the optimal value for region ratio r which minimizes access latency and energy consumption while guaranteeing the lifespan constraint L_c should be calculated.

PE_S and PE_M are the available P/E cycles of the SLC flash block and the MLC flash block, respectively. The values of PE_S and PE_M of the current products are 50 K and 10 K, respectively. P is the size of one block when it is configured as an SLC block, B is the total number of flash blocks in an SLC/MLC combined flash memory chip, and r is the region ratio, which is the ratio between the number of SLC blocks and the number of MLC blocks. There are a finite number of candidate values for r. If r is 0, then the SLC/MLC combined flash chip functions as an MLC chip. Therefore, when we assume that the MLC mode cell can contain two bits and the region ratio is r, $P \cdot B \cdot r$ and $2P \cdot B \cdot (1-r)$ are the storage capacities of the SLC region and the MLC region, respectively. By multiplying the available P/E cycle by the capacity of each region, we determine the total data size writable to the region before its lifespan is expired as follows:

$$D_S = PE_S \cdot P \cdot B \cdot r \quad (1)$$

$$D_M = PE_M \cdot 2P \cdot B \cdot (1-r) \quad (2)$$

We are to represent the lifespan of flash memory with the maximum write amount for the repetitive input workload. When the input workload writes W_H amount of data at SLC/MLC combined flash memory, W_S and W_M amounts of data will be written to the SLC and MLC regions, respectively, as follows:

$$W_S = W_H + W_{S\to S} + W_{M\to S} + W_{D\to S} \quad (3)$$

$$W_M = W_{S\to M} + W_{M\to M} \quad (4)$$

where $W_{S\to S}$, $W_{M\to S}$, and $W_{D\to S}$ are the data amounts written by the page migrations within the SLC region, the page migrations from the MLC region to the SLC region (via the read hit at the MLC region), and the page migrations from the disk to the SLC region (via the read hit at the hard disk), respectively. $W_{S\to M}$ and $W_{M\to M}$ are the data amounts written by the page migrations from the SLC region to the MLC region and the page migrations within the MLC region, respectively. The values of W_S and W_M depend on the region ratio since the data migrations from the garbage collection are affected by each region's size.

Since the proposed scheme manages the SLC region and the MLC region as circular buffers, it can be regarded to use a perfect wear-leveling technique. Using W_S and W_M, the lifespans of both the SLC region and the MLC region, L_S

and L_M, can be represented under perfect wear-leveling as follows:

$$L_S = W_H \cdot N_S \quad (\text{where } N_S = D_S/W_S) \quad (5)$$
$$L_M = W_H \cdot N_M \quad (\text{where } N_M = D_M/W_M) \quad (6)$$

Depending on the region size and the workload behavior, one of the two regions will exhaust all of its P/E cycles before the other, i.e., $N_S \neq N_M$. Then, the remaining region alone will serve host requests. From Equations (1), (2), (5) and (6), we can derive that $N_S \geq N_M$ if $r \geq 2W_S/(5W_M + 2W_S)$ as follows.

$$\begin{aligned} & N_S \geq N_M \\ \Leftrightarrow \quad & \frac{PE_S \cdot P \cdot B \cdot r}{W_S} \geq \frac{PE_M \cdot 2P \cdot B \cdot (1-r)}{W_M} \\ \Leftrightarrow \quad & \frac{5r}{W_S} \geq \frac{2-2r}{W_M} \quad (\because PE_S = 5 \cdot PE_M) \\ \Leftrightarrow \quad & r \geq \frac{2W_S}{5W_M + 2W_S} \end{aligned}$$

If $N_S > N_M$, the MLC region expires before the SLC region. The amount of data that is written to the SLC-only chip or to the MLC-only chip, W_{S_only} or W_{M_only}, can be represented as follows:

$$W_{S\_only} = W_H + W'_{S\to S} + W'_{D\to S} \quad (7)$$
$$W_{M\_only} = W_H + W'_{M\to M} + W'_{D\to M} \quad (8)$$

As shown in Fig. 5, if $N_S > N_M$, the SLC region consumes $N_M \cdot W_S$ of capacity among D_S when the lifespan of the MLC region expires. Therefore, the remaining storage capacity is $D_S - N_M \cdot W_S$, which will be consumed at a rate of W_{S_only}. For the case of $N_S < N_M$, the lifespan of the NVC can be estimated using a similar model. Then, the lifespan of the overall NVC, L_{NVC}, can be represented as follows:

$$L_{NVC} = W_H \cdot N_{NVC} \quad (9)$$
$$\text{if } (r \geq \frac{2W_S}{5W_M + 2W_S}) \;\; // \; N_S \geq N_M$$
$$N_{NVC} = \frac{D_S - N_M \cdot W_S}{W_{S\_only}} + N_M \quad (10)$$
$$\text{if } (r < \frac{2W_S}{5W_M + 2W_S}) \;\; // \; N_S < N_M$$
$$N_{NVC} = \frac{D_M - N_S \cdot W_M}{W_{M\_only}} + N_S \quad (11)$$

Therefore, for the given workload W_H and the region r, the lifespan of the NVC can be calculated by measuring the values of W_S, W_M, W_{S_only} and W_{M_only}. Then, we can search the minimum value r_{min} which satisfies the constraint $L_{NVC} \geq L_C$ by simulating several candidate values. Since we have finite number of candidates and the change in lifespan is nearly monotonic for the region ratio, as the experiments show, we can find the optimal value with a few number of simulations.

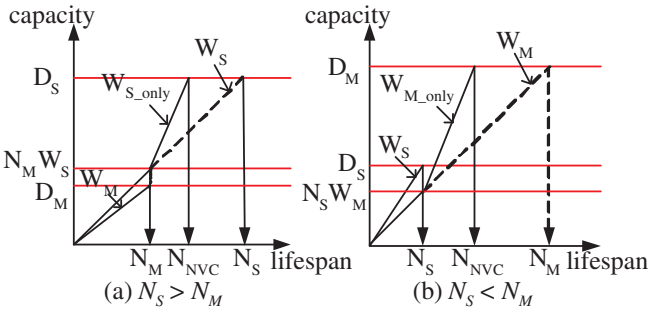


Figure 5. The lifespans of the SLC and MLC regions.

VI. Experiments

In order to evaluate the efficiencies of the proposed techniques, we implemented a hybrid HDD simulator which uses SLC/MLC combined flash memory as an NVC. We used four workloads as simulator inputs, bonnie++, desktop, financil1 and finacial2. bonnie++ is a popular storage benchmark program. Since it is generally used to evaluate the performances of storage systems, the inter-arrival times between requests are too short and, thus, it is not suitable for examining hybrid HDDs that exploit idle intervals. We therefore modified the timing values of the I/O requests so that there would be sufficient idle times between requests. Desktop is a real I/O trace collected executing desktop applications. Financial1 and financial2 are OLTP application traces used in [15].

We use the power state transition model shown in Fig. 6. The hard disk is modeled based on Samsung's HM080H1 product and has three power states, *active*, *idle* and *standby*. The threshold-based power management (TPM) technique is used to spin-down the hard disk if there are no requests during a predefined idle time. Then, the hard disk is changed into the stand-by state, which consumes less energy.

When the hard disk is servicing read/write requests, its state is *active*, which requires 2.55 W of power consumption. If there are no more requests from the host after one second, the state of HDD changes to *idle*, which consumes 0.8 W. Switching between the active state and the idle state takes 0.5 seconds and consumes 1.15 J of energy. Additionally, if there are no requests for 3.38 seconds, then the HDD stops spinning, and the HDD state changes to *standby*, which consumes only 0.25 W. The change into standby state takes 2.3 seconds and consumes 2.94 J of energy. When a request comes from the host, the standby state is changed into the active state to service the request, which invokes time and energy overheads of 1.6 seconds and 5.0 J.

We used the timing and endurance parameters of NAND flash memory as shown in Table I. The SLC and MLC

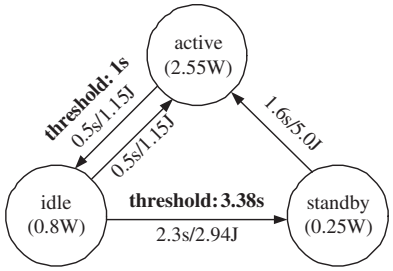


Figure 6. The power state transition diagram of an HDD.

Table I
NAND FLASH PARAMETERS.

Block type	read	write	erase	P/E cycles
SLC mode	409	431	872	50K
MLC mode	403	994	872	10K

mode blocks have similar read performances but their write performances are quite different.

A. Performance of the Hybrid HDD

In order to show the effectiveness of NVC, we first compared the hybrid HDD with a normal hard disk without NVC. The NVC is composed of 2048 flash blocks, each of which is 256 KB in SLC mode and 512 KB in MLC mode. Therefore, the size of NVC is 512 MB when all the blocks are SLC mode while the size is 1 GB when all the blocks are MLC mode. We measured three metrics of the hybrid HDD while varying the size of the SLC region in the NVC: the read latency, the lifespan of the NVC, and the energy consumption of the hard disk. We focused only on the read performance since all the write requests are first sent to the DRAM buffer and, thus, the write latencies are the same. However, the read latency depends on the location of the requested data. The lower the read-miss rate of NVC is, the better read performance the hybrid HDD provides.

Fig. 7 (a) shows the average read latency and energy consumption of a hybrid HDD with NVC, normalized by those values of the normal HDD. Figures 7 (b) and (c) show how the read-miss rate and the GC overhead of the NVC change when the number of SLC mode blocks varies. The GC overhead means the page migration cost by GC. The read latencies are reduced by 52∼97%, and the energy consumptions are reduced by 4∼71%. The reduction in read latency depends on the read-miss rates. Depending on the localities of the workloads, the read-miss rates are significantly different. Since financila1 and financial2 have low read-miss rates in the NVC, their read latencies are significantly reduced. However, their energy reductions are small since the reduced miss rates do not increase the hard disk idle times above the threshold time for power management.

The differences between the maximum and minimum value of the miss rate are 27%, 21%, 19%, and 7% for bonnie++, desktop, financial1 and financial2, respectively. The GC overheads show little difference depending on the SLC region size. The overheads for all of the workloads except financial1, which is write-intensive, are smaller than 15% of the total I/O cost. For the financial1 workload, when the total size of the NVC is small due to the allocation of many blocks to the SLC region, the number of HDD accesses increases and, thus, the GC overhead increases. The GC overheads of SLC-only chips and MLC-only chips are small in comparison to those of mixed configurations since they do not invoke page migrations between the two regions.

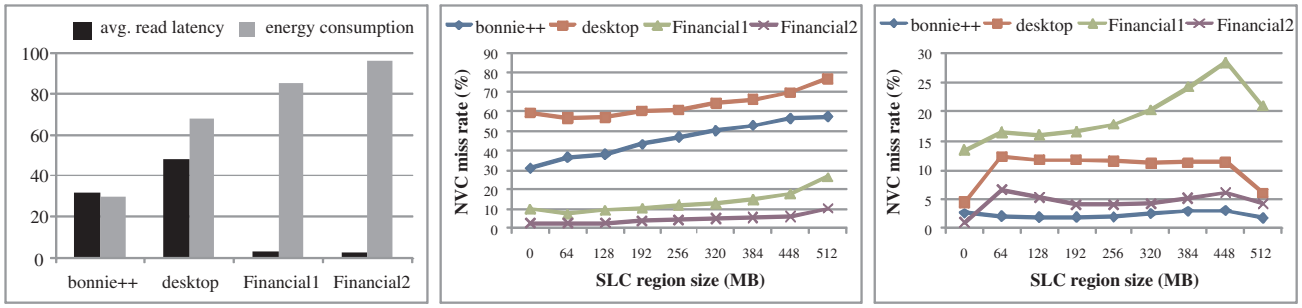
Fig. 8 shows the normalized average read latency, lifespan of NVC, and energy consumption when varying the size of the SLC region. As the size of the SLC region increases, the total size of the NVC decreases. Therefore, the read latency and energy consumption increase due to higher NVC miss rates. However, the lifespan of the NVC increases since the number of P/E cycles of an SLC block is five times that of an MLC block. The energy consumptions of most workloads, except bonnie++, show very small differences depending on the SLC region size. This is because the changes in miss rate cause no significant change in the idle intervals that are above the TPM threshold value.

As we assumed at Section V, these results for the three metrics increase monotonically as the proportion of the SLC region increases. Using Fig. 8, we can explain why the SLC/MLC combined flash memory is useful for NVC. For example, if we have the constraint that the lifespan of NVC should be more than twice the lifespan of MLC-only NVC, the minimum SLC region size for the financial1 workload is identified as 336 MB from the result in Fig. 8(b). By selecting the SLC region size, the read latency is reduced by 67% in comparison to the SLC-only NVC. Consequently, we can implement an SLC/MLC combined NVC which provides better performance than an SLC-only NVC and a longer lifespan than an MLC-only NVC.

In order to determine the minimum size of the SLC region satisfying the lifespan constraint, we should measure the values of W_S, W_M, W_{S_only}, and W_{M_only} and estimate the NVC lifespan for each configuration. Fig. 9 shows these values for each SLC region size. As the SLC region size increases, $W_{S\rightarrow S}$ increases since the GC victim block has many invalid pages. $W_{M\rightarrow S}$ decreases as the size of MLC region decreases, and $W_{D\rightarrow S}$ increases since a significant amount of data is evicted to HDD when the size of the SLC region is large. $W_{M\rightarrow M}$ and $W_{S\rightarrow M}$ decrease as the MLC region decreases.

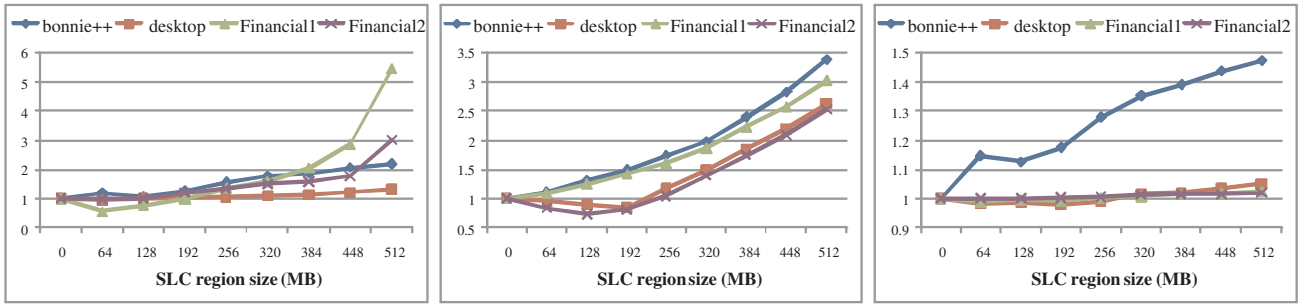
B. Comparison with alternative policies

To ensure that the proposed NVC management policy (Policy 0) is better than other alternative policies, we compared it with the following three alternative policies:



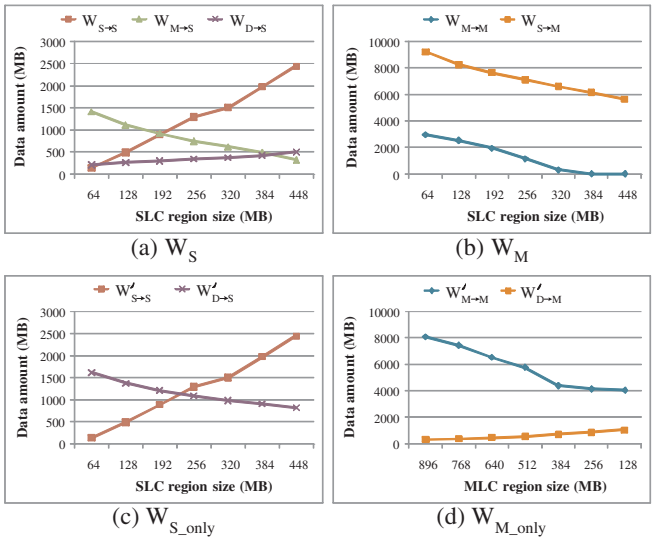
(a) comparison with normal HDD (b) read-miss rate (c) GC overhead

Figure 7. Performance of the hybrid HDD.



(a) change of avg. read latency (b) change of NVC lifespan (c) change of energy consumption

Figure 8. Changes in hybrid HDD behaviors by varying the size of the SLC region.

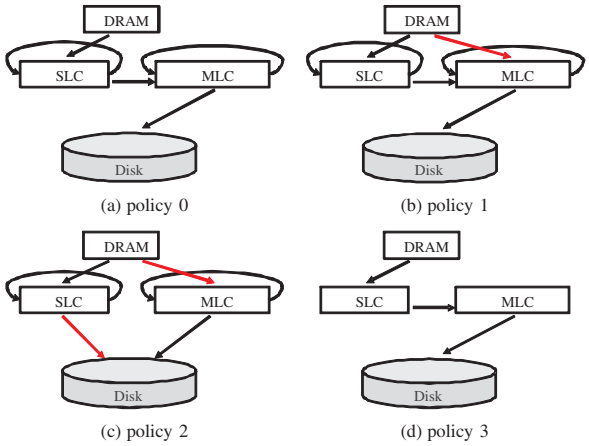


(a) W_S (b) W_M

(c) W_{S_only} (d) W_{M_only}

Figure 9. The changes in the financil1 workload with SLC region size.

- Policy 1: same as Policy 0 except that the large-sized write requests bypass the SLC region, assuming the large data will be cold data.
- Policy 2: the SLC and MLC regions are managed separately, thus there is no migration between two regions. The write requests for large data are sent to the MLC region.
- Policy 3: same as Policy 0 except that the garbage collection for each region moves cold pages into lower-level storage without internal migrations.



(a) policy 0 (b) policy 1

(c) policy 2 (d) policy 3

Figure 10. Comparison with the alternative policies.

Fig. 10 illustrates the alternative policies. Figures 11, 12 and 13 show the average read latencies, lifespans and energy consumptions of the alternative policies normalized by the results of Policy 0. The motivation of Policy 1 is to reduce the migration overhead of cold data. Since it is probable

for the cold data to be moved into the MLC region by GC without any updates, it will be better to bypass the SLC region. However, the experimental results show that Policy 1 is slightly worse than Policy 0 because the miss penalty of the MLC region is high in comparison to the flash access cost, and there are many cases in which the predictions for the cold data are incorrect. If we use a more accurate prediction scheme, Policy 1 may show better results.

Policy 2 shows the worst results for read latency and energy consumption since it invokes many requests on the hard disk. However, it shows good results for NVC lifespan since it does not invoke page migrations between the SLC and the MLC regions. Especially when the SLC region size is 176∼272 MB (i.e., when the SLC blocks are 33∼51% of the total blocks), the lifespan is maximized since the two regions are balanced.

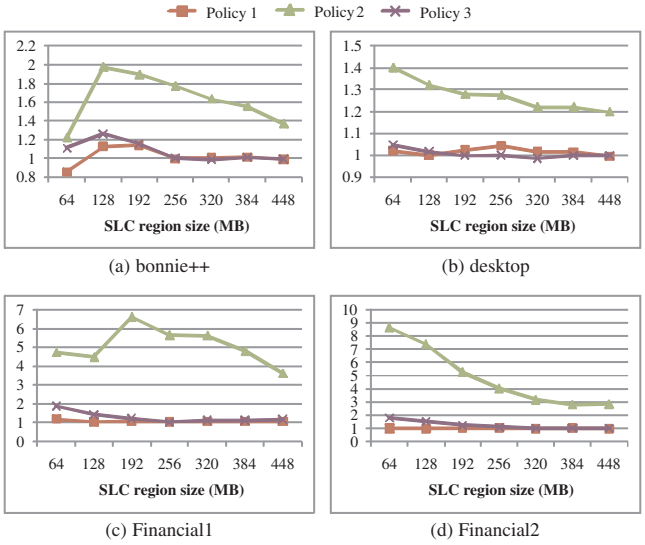


Figure 11. Normalized average read latency of the hybrid HDD.

The read latency and energy consumption of Policy 3 are slightly worse than those of Policy 0 while it provides a longer lifespan than Policy 0, especially when the size of SLC region is small. Since Policy 3 does not perform the internal page migrations, it is profitable in reducing the write amount on NVC. Therefore, it is recommended to use Policy 3 when the lifespan constraint is too tight. Actually, Policy 3 can be regarded as an implementation of Policy 0 because if we use tight thresholds for internal migrations, Policy 0 becomes Policy 3.

VII. Conclusions

SLC/MLC combined flash memory is a viable non-volatile cache solution for providing both a high performance and a long lifespan to a hybrid HDD. To maximize the advantages of the hybrid architecture, the proposed technique efficiently exploits the SLC and MLC regions

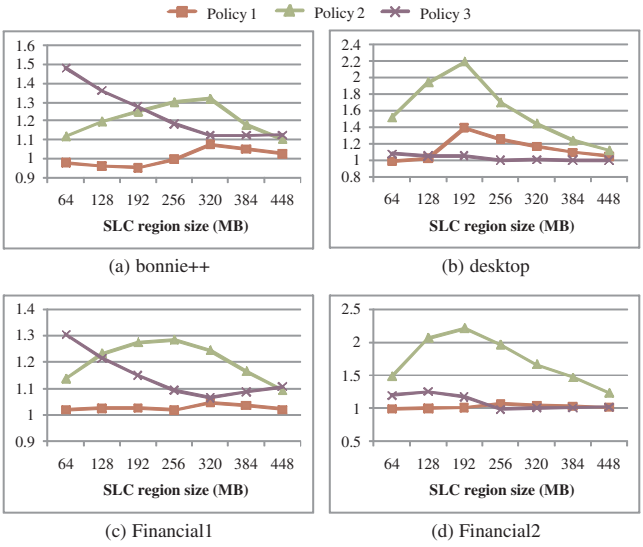


Figure 12. Normalized lifespan of the NVC.

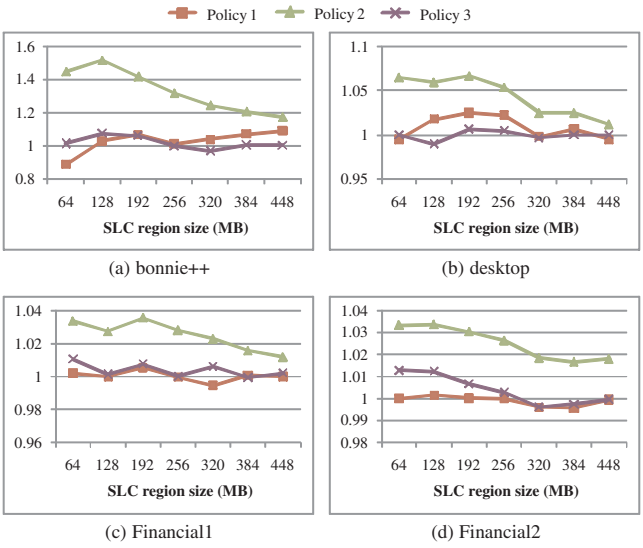


Figure 13. Normalized energy consumption of the hybrid HDD.

by utilizing the SLC region as a first-level write buffer for hot data and the MLC region as a second-level write buffer for cold data. It can also determine the optimal sizes for the two regions in order to maximize performance and energy reduction guaranteeing the lifespan constraint. In future works, we will study adaptive management schemes which can dynamically adjust the garbage collection policy depending upon the workload change.

References

[1] R. Panabaker, "Hybrid hard disk & readydrive technology: improving performance and power for Windows Vista mobile PCs," Microsoft WinHEC, 2006.

[2] J. Matthews, S. Trika, D. Hensgen, R. Coulson, and K. Grimsrud, "Intel turbo memory: Nonvolatile disk caches in the storage hierarchy of mainstream computer systems," *ACM Transactions on Storage*, vol. 4, no. 2, pp. 1–24, 2008.

[3] M. B. et al., "A multilevel-cell 32Mb flash memory," in *Proc. of the Solid-State Circuits Conference*, 1995, pp. 132–133.

[4] L.-P. Chang, "Hybrid solid-state disks: Combining heterogeneous nand flash in large ssds," in *Proc. of Asia and South Pacific Design Automation Conference (ASP-DAC)*, 2008, pp. 428–433.

[5] F. Roohparvar, "Single level cell programming in a multiple level cell non-volatile memory device," In United States Patent, No. 7,366,013.

[6] T. C. et al., "A dual-mode nand flash memory: 1-Gb multilevel and high-performance 512-Mb single-level modes," *IEEE Journal of Solid-State Circuits*, vol. 36, no. 11, 2001.

[7] "Samsung Electronics, 4Gb Flex-OneNAND M-die," http://www.samsung.com.

[8] T. Bisson and S. A. Brandt, "Reducing hybrid disk write latency with flash-backed i/o requests," in *Proc. of the International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems*, 2007, pp. 402–409.

[9] ——, "Flushing policies for nvcache enabled hard disks," in *Proc. of IEEE Conference on Mass Storage Systems and Technologies*, 2007, pp. 299–304.

[10] T. Bisson, S. A. Brandt, and D. D. Long, "A hybrid disk-aware spin-down algorithm with i/o subsystem support," in *Proc. of the International Performance, Computing and Communications Conference*, 2007, pp. 11–13.

[11] F. Chen, S. Jiang, and X. Zhang, "SmartSaver: turning flash drive into a disk energy saver for mobile computers," in *Proc. of the international symposium on Low power electronics and design*, 2006, pp. 412–417.

[12] J.-W. Hsieh, T.-W. Kuo, P.-L. Wu, and Y.-C. Huang, "Energy-efficient and performance-enhanced disks using flash-memory cache," in *Proc. of the international symposium on Low power electronics and design*, 2007, pp. 334–339.

[13] Y.-J. Kim, S.-J. Lee, K. Zhang, and J. Kim, "I/o performance optimization techniques for hybrid hard disk-based mobile consumer devices," *IEEE Transactions on Consumer Electronics*, vol. 53, no. 4, pp. 1469–1476, 2007.

[14] L. Useche, J. Guerra, M. Bhadkamkar, M. Alarcon, and R. Rangaswami, "EXCES: External caching in energy saving storage systems," in *Proc. of International Symposium on High Performance Computer Architecture*, 2008, pp. 89–100.

[15] T. Kgil, D. Roberts, and T. Mudge, "Improving NAND flash based disk caches," in *Proc. of the International Symposium on Computer Architecture*, 2008, pp. 327–338.

[16] S. H. Park, J. W. Park, J. M. Jeong, J. H. Kim, and S. D. Kim, "A mixed flash translation layer structure for SLC-MLC combined flash memory system," in *Proc. of SPEED'08*, 2008.

[17] S. Lee, K. Ha, K. Zhang, J. Kim, and J. Kim, "FlexFS: A flexible flash file system for MLC NAND flash memory," in *Proc. of USENIX Technical Conf.*, 2009.

[18] S. Im and D. Shin, "Storage architecture and software support for SLC/MLC combined flash memory," in *Proc. of ACM Symposium on Applied Computing (SAC'09)*, 2009.