Region-based Flash Caching with Joint Latency and Lifetime Optimization in Hybrid SMR Storage Systems

Zhengang Chen[1], Guohui Wang[1*], Zhiping Shi[2],Yong Guan[3],Tianyu Wang[4]
[1] College of Information Engineering, Capital Normal University, Beijing, China
[2] Beijing Key Laboratory of Electronic, System Reliability Technology, Capital Normal University, Beijing, China
[3] International Science and Technology Cooperation Base of Electronic System Reliability and Mathematical Interdisciplinary, Capital Normal University, Beijing, China
[4] Department of Computer Science and Engineering, The Chinese University of Hong Kong, HongKong, China
{zgchen, ghwang, shizp, guanyong}@cnu.edu.cn, tywang@cse.cuhk.edu.hk

***Abstract*— The frequent Read-Modify-Write operations (RMWs) in Shingled Magnetic Recording (SMR) disks severely degrade the random write performance of the system. Although the adoption of persistent cache (PC) and built-in NAND flash cache alleviates some of the RMWs, when the cache is full, the triggered write-back operations still prolong I/O response time and the erasure of NAND flash also sacrifices its lifetime. In this paper, we propose a Region-based Co-optimized strategy named Multi-Regional Collaborative Management (MCM) to optimize the average response time by separately managing sequential/random and hot/cold data and extend the NAND flash lifetime by a region-aware wear-leveling strategy. The experimental results show that our MCM reduces 71% of the average response time and 96% of RMWs on average compared with the Skylight (baseline). For the comparison with the state-of-art flash-based cache (FC) approach, we can still save the average response time and flash erase operations by 17.2% and 33.32%, respectively.**

***Index Terms*—Shingled magnetic recording, flash memory**

I. INTRODUCTION

The Shingled Magnetic Recording (SMR) disks utilize a shingled-like structure to increase disk capacity significantly but introduce Read-Modify-Write operations (RMWs), which perform badly in random write scenarios [1]–[3]. To further improve the storage efficiency of SMR disks, many cache-assisted architectures are proposed to absorb frequent random writes, such as the Persistent Cache (PC, residing in the disk outer diameter) [4], SSD as the cache [5] and NAND Flash as the Cache (FC) [6]. However, the persistent cache still generates a large number of RMWs due to the trigger of the cleaning operation when it is full; the flash-based cache suffers from inefficient garbage collection (GC) which prolongs system average response time and reduces the lifetime of the flash memory.

In this paper, we propose a Region-based Co-optimized strategy named Multi-Regional Collaborative Management (MCM) to optimize the average response time and extend the NAND flash lifetime. The basic idea is to manage sequential/random data and hot/cold data separately while reorganizing the NAND flash data distribution accordingly for efficient garbage collection and wear leveling. Thus, two questions need to be answered: (1) how to divide data into sequential hot data, sequential cold data, and random data and store them individually; (2) how to extend the lifetime of NAND flash and minimize RMWs through MCM.

To address the first challenge, we design a two-stage data partitioning module. In the first stage, the data is divided into random and sequential data based on the band-aware module. In the second stage, the sequential data is further divided into sequential hot and sequential cold data by a Time-window-based Double Counting Bloom Filter (TD-CBF). According to different data properties, the NAND flash is partitioned into sequential hot data region (SHR), random data region (RDR), and replacement buffer region (RBR, used to absorb the cache eviction).

To address the second challenge, we build a global page mapping table (GMT) to manage NAND flash spaces efficiently. Specifically, we will record the block temperature changes among different regions in NAND flash through the TD-CBF module, then allocate and reclaim blocks accordingly. The SHR and RDR regions can use each other's free blocks when their own blocks are exhausted. If both regions are fully occupied, the RBR region is used for the block migration to achieve wear-leveling. The GC will only be triggered when all the NAND flash spaces are used up, through which we can selectively write valid data blocks back to the SMR disk. By specially managing NAND flash data in different regions, we can achieve better wear-leveling to extend its lifetime while reducing GC frequency to minimize RMWs.

We implement our techniques based on a widely-used SMR simulator, Skylight [4], to show the effectiveness of MCM. The evaluation is performed with collected traces from Microsoft Research Cambridge [7]. The experimental results show that our MCM reduces 71% of the average response time and 96% of RMWs on average compared with the Skylight (baseline). For the comparison with the state-of-art flash-based cache (FC) approach, we can still save the average response time and flash erase operations by 17.2% and 33.32%, respectively. The main

978-3-9819263-7-8/DATE23/© 2023 EDAA

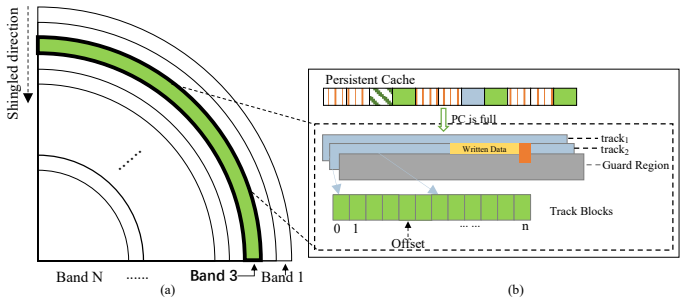

Fig. 1: An overview of an SMR disks. (a) Internals of SMR disks. (b) An illustrative example when persistent cache (PC) is full.

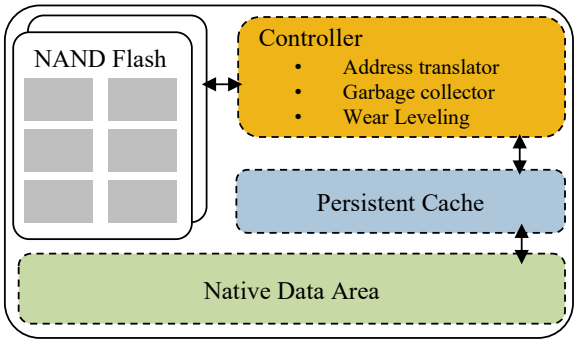

Fig. 2: Hybrid storage system overview.

contributions of this paper are summarized as follows:

- We propose a Region-based Co-optimized strategy named MCM to optimize the system average response time and extend the lifetime of NAND flash.
- We introduce a TD-CBF model to categorize data into different groups based on their band information and temperatures while managing NAND flash spaces accordingly. We also achieve better wear-leveling by a region-aware block migration strategy.
- We integrate the NAND flash controller into a widely-used SMR simulator, Skylight, to evaluate our schemes with various real-world workloads.

The rest of this paper is arranged as follows. Section II introduces the background and motivation of this work. Section III describes the technical details and the implementation of our MCM. The experimental results are presented in Section IV and the last section concludes this paper.

II. BACKGROUND AND MOTIVATION

A. SMR Disk and Persistent Cache

Shingle Magnetic Recording technology enhances storage density by partially overlapping adjacent tracks. As shown in Figure 1 (a), SMR disks split each platter into multiple bands (the basic unit of RMWs), each band contains k (e.g., k=2) tracks, and the guard region separates the bands. As shown in Figure 1 (b), each band contains two tracks with one guard region. Updating the data in the underlying track needs three steps: (1) sequentially read out all the data in the band [4]; (2) Merging the valid data after the update/modification; (3) sequentially write the merged data back to the original tracks. This process is called RMW which significantly degrades system performance. To mitigate RMWs, one simple solution is to set a portion of the SMR disk outer diameter (1%∼10%) as a PC. The written data is first cached sequentially into the PC. When the PC is full, the cleaning operation is triggered to write the valid data back to the Native Data Area (NDA) sequentially. However, the cleaning operations still generate a large number of RMWs.

B. NAND Flash as Cache

NAND flash is another candidate to absorb RMWs due to its outstanding read/write performance [8], [9]. As shown in Figure 2, when integrate NAND flash into the SMR disk to build a hybrid storage system, the NAND flash and SMR disk can be jointly managed through the shingled translation layer (STL), which can also alleviate RMWs, thus improve system performance [5], [6]. However, due to the limited P/E cycles, taking NAND flash as cache still needs to judiciously consider the wear-leveling and garbage collection strategy to achieve better lifetime.

C. Motivation

As mentioned in Section II-A, large-scale PC cleaning inevitably generates RMWs, affecting the system performance [10]–[12]. To illustrate the impact of RMWs caused by different workloads ($rsrch\_0$: 90.68% write, 9.32% read; $src2\_0$: 88.61% write, 11.39% read) on SMR disks, we do some preliminary experiments showing I/O latency of workloads in SMR disks. The results are shown in Figure 3 (Left), in which we can see that some I/O requests need 5 seconds or more to be processed. This latency will significantly slow down the system and hurt the user experience.

When adopting built-in NAND flash as the primary cache for the hybrid storage, the incoming I/O requests can be firstly processed in the NAND flash then selectively written back to SMR disk [13]. This method can reduces RMWs while improving data read efficiency [14], [15]. However, when NAND flash is full, the data needs to be flushed to SMR disk to reclaim free spaces in NAND flash through an erase operation. As shown in Figure 3 (Right), frequently erasing physical blocks in NAND flash will significantly hurt its endurance [16].

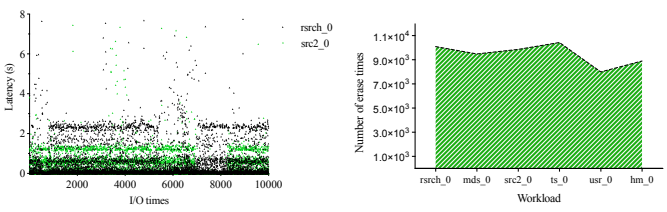

Fig. 3: Preliminary experiment on the influence of RMWs with different workloads on the system (Left) and the number of erase times (Right).

Thus, in this paper, we propose a scheme to jointly manage the SMR disk and NAND flash for the RMWs reduction and the flash lifetime extension. We distinguish data with different properties (temperature, randomness) and manage them individually to reduce RMWs. By avoiding excessive NAND flash erasures caused by some cold sequential data which only passes through NAND flash once, we can extend its lifetime. We also perform the data redistribution inside NAND flash to balance the P/E cycles among physical blocks.

III. DESIGN AND IMPLEMENTATION

In this section, we first present the system overview of the MCM in Section III-A. Then, in Section III-B, we describe a two-stage data allocation strategy. Last, Section III-C describes the block migration strategy for NAND flash during GC and the strategy for writing data back to SMR disks.

A. System Overview

As shown in Figure 4, the SMR disk serves as the main storage and NAND flash serves as the cache for random data and sequential hot data in the hybrid system. The MCM consists of four functional modules:

- Band-aware random and sequential data partitioning module.
- TD-CBF assisted hot and cold data identification module.
- Space Allocation and global address mapping module.
- Garbage collection and Wear leveling module for NAND flash.

Among them, the Two-stage Data Partition consist of Band-aware module and TD-CBF module. We divide the physical space of NAND flash into three regions for allocation. The SHR and RDR are both allocated with 7168 physical blocks, and 2048 physical blocks are allocated for RBR. We use a global page mapping table to record the address mapping between logical sector numbers and physical pages and store them in a B^+-tree for efficient indexing.

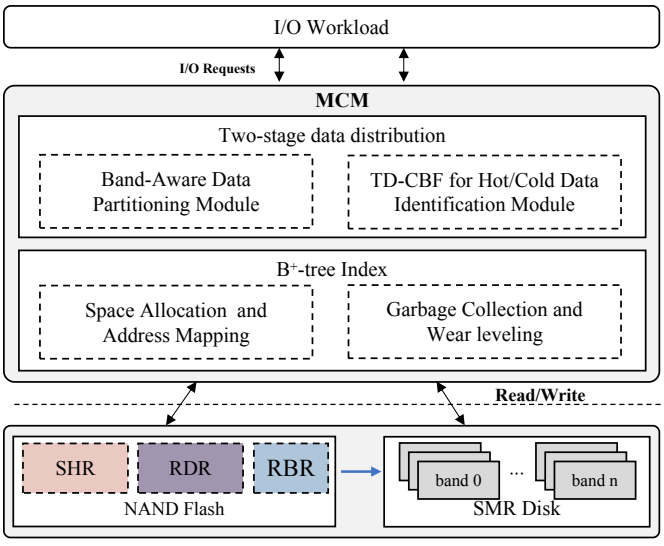

Fig. 4: An overview of the system architecture of MCM.

B. Two-stage Data Partition

In this section, we will present how the data is partitioned into three categories: random data, sequential hot data, and sequential cold data. Then, the data is divided into sequential and random data according to the Band-aware Data Partitioning Module. After that, the sequential data is further divided into hot and cold data through the TD-CBF module. Finally, the three kinds of data with their special characteristics are stored in different physical spaces.

a) Band-aware Data Partitioning Module: In SMR disks, there are multiple tracks in a band, and a track contains multiple physical blocks with 4KB for each. Note that the band size is not fixed (varies from 17MB to 36MB) due to the different diameter of each band. In the first stage, when the LSNs of consecutive write requests are matched to the same band, we regard those requests as sequential data writes. Specifically, we use a counter k to count how many consecutive writes are located in the same band. If a particular band receives three or more consecutive writes, we regard those writes are sequential data. If the following request is mapped to another band, then the value of k will be recounted from 1.

Figure 5 shows an example about how we divide sequential and random data based on the counter k. For simplification, we use linear LSN to PSN (Physical Sector Number) mapping with each band containing 120 sectors. Given a sequence of write requests $(101, A), (201, B), (202, C), (203, D), (301, E), (380, H)$,and $(390, K)$, the corresponding band numbers are 1, 2, 2, 2, 3, 4, and 4, respectively. Considering the threshold of k ($= 3$), the data A is classified as random data ($k = 1$), data B, C, and D are regarded as sequential data ($k = 3$), and data H, K are categorized to random data ($k = 2$).

b) TD-CBF for Hot/Cold Data Identification Module: As Counting Bloom Filter (CBF) methods have the advantage of accurately recording data access frequency, when combining it with the time window, we can set data temperature by analyzing the data locality. We propose TD-CBF which can make good use of the advantages of counting bloom filters and compensate for the disadvantage that the original algorithm cannot guarantee the timeline. The structure of TD-CBF is shown in Figure 6, which is mainly partitioned into two parts. (1) A two-bit counting bloom filter. (2) The history table (HT) and current table (CT) based on the timeline.

In the second stage, K hash functions with a band number as the parameter generate K hash values in TD-CBF. Based on the K hash values, the corresponding position of the hash

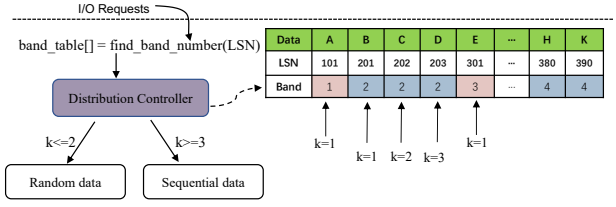

Fig. 5: Structure framework of Band-aware Module.

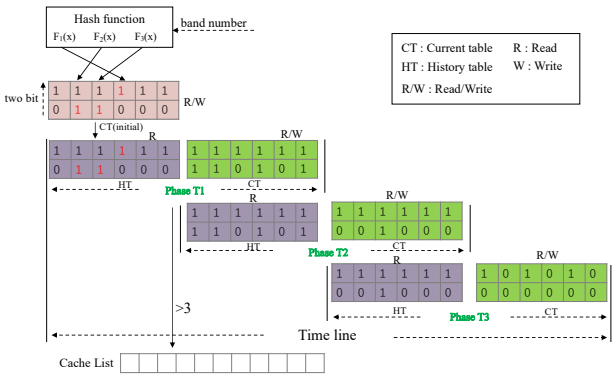

Fig. 6: Structure framework of TD-CBF.

table is set to 1 (The initial CT value is 0). If the same band is accessed, the values of the bloom counters are increased by 1 through shifting one bit to the downside (as the hash table with pink background in Figure 6 shows). When the initial CT is full (more than 90%), it is transferred to a read-only HT in phase T1. Then, a new CT is created for writing/reading. When the CT in phase T1 is full, the previous HT is destroyed. In phase T2, the full CT in phase T1 is also transferred to a read-only HT. A new CT is created again for writing/reading. The TD-CBF is runing with repetitive operations. In this process, if the value of HT or CT is greater than or equal to 3 (i.e., the threshold value is 3), the data is identified as sequential hot data. In contrast, if the value is less than 3, the data is identified as sequential cold data. Then, these data are cached to the cache list. Note that the MCM will consider the sequential data as sequential cold data at initial time. In Figure 6, the Phase T1, T2 and T3 are consecutive request phases. The reason why we set up the CT and HT tables is that the data may change from cold to hot in a period of time. We don't want to discard the former data so quickly, and using the HT table as search, the current data of the CT has the effect of reducing misclassification.

The TD-CBF is designed with three key parameters: the Bloom Filter size (M), the number of hash functions (K), and the number of samples (N). The relationship of the parameters as follows:

$$M = -\frac{N \ln p}{(\ln 2)^2} \tag{1}$$

Where p is the maximum false positive rate (FPR) in the process. After determining M, the value of K can be derived from (2).

$$K = -\log_2 p \tag{2}$$

The FPR is set to 14.7% with 2^{10} input samples per round. Through (1) and (2), we can get M≈4087 bits (choose 2^{12}) and $K = 3$. Each independent TD-CBF contains three hash functions (i.e., Direct modular arithmetic, Folding method, Middle-square method). Each item of the hash table occupies 2-bit, so the memory consumption is $2KB$. The length of the cache list is set to 1024, and each item takes $4Byte$ memory to store the band number. The cache list consumes $4KB$ memory for the launch. Hence, the total memory consumption is $6KB$.

Algorithm 1: Wear Leveling and Garbage Collection

```
1  if SHR or RDR cannot meet writes then
2      if SHR and RDR is full then
3          if RBR is full then
4              trigger the GC process;
5              select the block with more invalid pages;
6              foreach valid page to TD-CBF do
7                  if valid page is hot page then
8                      write back to NAND flash.
9                  else
10                     write back to SMR disks.
11                 end
12             end
13         else
14             migrate the colder data blocks in the SHR to the RBR, then
                 erase the original physical blocks and remapping;
15         end
16     end
17     block migration between SHR and RDR;
18 else
19     continue to write;
20 end
21 return;
```

C. Wearing Leveing and Garbage Collection

In order to balance the P/E cycles among all physical flash blocks and trigger the garbage collection as less as possible, a new wear-leveling and garbage collection strategy is designed. As shown in Algorithm 1, the write operations continue when there are free blocks in SHR and RDR (Line 1). When the SHR(or RDR) runs out of its free blocks, we will check if another region RDR(or SHR) still has free blocks for allocation (Line 16). When the two regions (SHR and RDR) cannot satisfy the writing request, check whether there is a free block in the RBR (Line 13). If there are free blocks in RBR, the cold data blocks in SHR (or RDR) are migrated to RBR. Meanwhile, the original physical block is erased, and a new free block in RBR is picked up and remapped to SHR. To evaluate the temperature levels of flash blocks, we propose (3) to calculate the Thv (the temperature value of a particular block).

$$\begin{cases} Thv = \alpha \times \pi + \beta \times Ebs[i], i \in Blocks \\ \alpha + \beta = 1 \end{cases} \tag{3}$$

Where $Ebs[i]$ represents the number of erases of the i_{th} block and π represents the number of invalid pages in the block. The variables α and β are parameters to represent the weight of invalid pages and erasures in the flash block, respectively. In the experiment, α = 0.6 and β = 0.4 are set. Through (3), the physical block with the smaller (i.e., colder) value in the SHR (or RDR) is selected for migration to the RBR.

When all the three regions are full, GC is triggered. Then the valid data in the RBR is written back to the SMR disk (Line 4-9). Before writing data back to the SMR disk, we need to search the hash table in TD-CBF to identify the current hot and cold status of the valid pages. If the valid pages are hot, they are written back to SHR in NAND flash while only flushing cold pages to the SMR disk.

An example is shown in Figure 7. Blocks #1001 and #102 need to be recycled. The scheme uses the TD-CBF thresholds to determine the hot and cold status of the pages in the two blocks. If a valid page in the block is hot (i.e., pages 1011 and

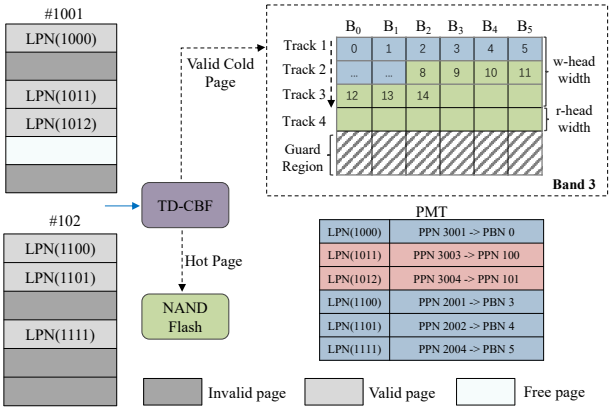

Fig. 7: An example of garbage collection.

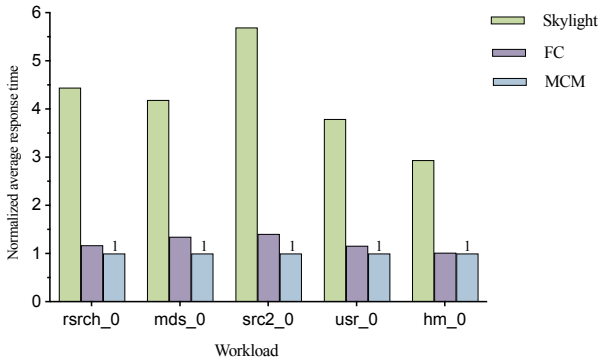

Fig. 8: Normalized average response time.

1012), it is written back to a free page in the SHR. Otherwise, it is written back to SMR disk directly. Finally, the mapping table is updated.

IV. EVALUATION

In this section, we will introduce the experimental setup in Section IV-A, and then present the experimental results and discussion in Section IV-B.

A. Experimental setup

We simulate a Seagate ST5000AS0011 5TB SMR disk based on the specification in Skylight [17]. The rotation speed is set 5900 rpm, and the write-head width is three times than the read-head width. We set 1% of the total capacity as the PC while the rest of the area is the NDA and simulate a 16GB MLC NAND flash. The latency of NAND flash page read, write and block erase are 0.75ms, 1.3ms, and 3.8ms, respectively [18].

We have evaluated the real-world traces collected from Microsoft Research Cambridge [7]. The detailed characteristics are shown in Table I. The traces including the total number of requests, write ratio (%), sequential data (%), and update ratio (%). Among these traces, $rsrch\_0$, mds_0 and $src2\_0$ are write-intensive workloads, and the rest traces are write-medium workloads.

B. Results and Discussion

In this section, we evaluate the effectiveness of our MCM, Skylight and FC by using the following performance metrics: a) average response time; b) write amplification factor; c) number of erase times and valid copies.

TABLE I: The characteristics of the traces.

Traces	Total Requests	Write Ratio (%)	Sequence data (%)	Update Ratio(%)
rsrch_0	1,433,655	90.68%	48.76%	99.92%
mds_0	1,211,034	88.11%	43.85%	95.50%
src2_0	1,557,814	88.61%	32.47%	94.76%
hm_0	3,993,316	64.50%	38.50%	92.10%
usr_0	2,237,889	62.54%	35.71%	99.88%

a) Average response time: We evaluate the overall performance improvement by comparing the average system response time of MCM with FC and Skylight. The experimental results are shown in Figure 8. It is observed that the average response time can be significantly reduced by parallel writing and efficient utilization of NAND flash. In $src2\_0$ workload, we achieve a maximum reduction of 82.44% compared to Skylight. On the average, we outperform Skylight by 71% and also save 17.2% response time compared with the state-of-art FC.

b) Write amplification factor: The main bottleneck of SMR disks is the long response time caused by RMWs. The PC can alleviate RMWs to a limited extent. When PC is full, the valid data of the PC have to be written back to the NDA, which causes high latency. Since our scheme avoids frequent writes and updates of hot data in the SMR, it mitigates the frequency of RMWs triggering and thus reduces the write amplification factor (i.e., WAF is calculated as the total write data size over the total input data). As shown in Figure 9, we normalized the WAF to compare our MCM with FC and Skylight. The scheme reduces the WAF by 25.2 times on average compared with Skylight and decreases 18.8% on average compared with FC.

c) Number of erase times and valid copies: The scheme uses NAND flash to mitigate RMWs, like FC. FC uses flash memory as a cache, and all the writes will be performed in a flash first and then be written back to SMR disks selec-

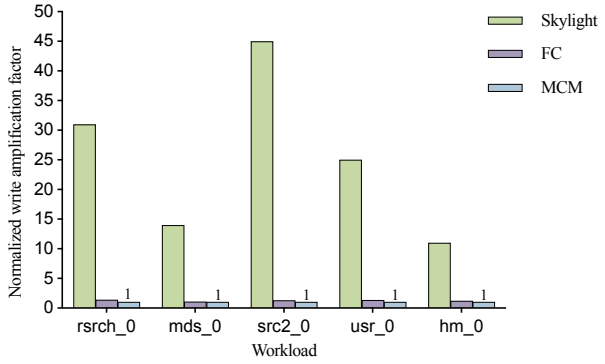

Fig. 9: Normalized write amplification factor.

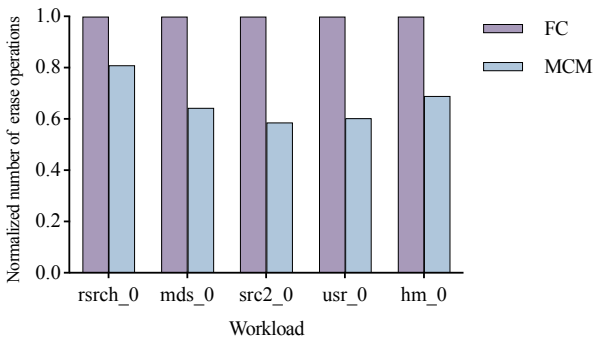

Fig. 10: Normalized number of erase times.

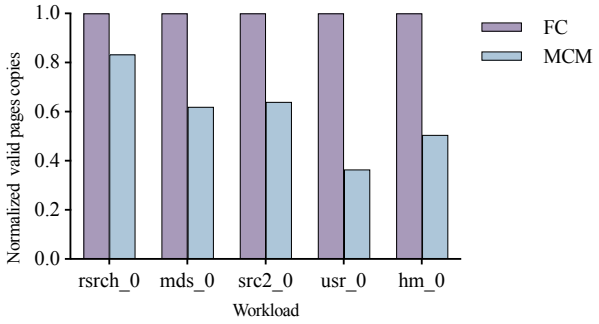

Fig. 11: Normalized number of valid page copies.

tively. However, it will result in a large number of erases simultaneously. The MCM scheme avoids generating many erase operations when writing data back to SMR disk, and MCM overlaps part of the GC time with normal processing. Experiments show that this part of the time is much lower compared to FC. As shown in Figure 10 and Figure 11, compared with FC, the maximum reduction of erase operations in our scheme is 41.34%, and the average reduction is 33.32%. Regarding the number of valid page copies, our scheme reduces the NAND flash erasures by about 40.7% on average.

V. CONCLUSION

This paper proposes a region-based co-optimization strategy for the hybrid SMR storage with built-in NAND flash. Specifically, we design a region-based co-managed MCM storage scheme based on NAND flash and SMR disks to solve the problems of redundant RMWs and NAND flash lifetime extension. The MCM first divides the requested data into sequential hot data, random hot data, and random data. Then it partitions the NAND flash and stores different types of data individually. Finally, the goals of extending flash lifetime and reducing RMWs are achieved by a region-aware block migration strategy and a new garbage collection method. Our experimental results show that our MCM can reduce the average response time of the system by more than 71% compared with the Skylight. We can also save the number of flash block erases by about 33.32% on average, compared to FC, with the same number of write requests. In the future, we will investigate learning policy-based approaches to more accurately discern the characteristics of I/O requests and improve system performance.

ACKNOWLEDGMENT

This work was supported by the Natural Science Foundation of China (62272323, 62272322, 62002246), the Project of Beijing Municipal Education Commission (KM202010028010), and the Academy for Multidisciplinary Studies, Capital Normal University. Guohui Wang is the corresponding author.

REFERENCES

[1] W. He and D. H. Du, "SMaRT: An Approach to Shingled Magnetic Recording Translation," in *15th USENIX Conference on File and Storage Technologies (FAST 17)*, Santa Clara, CA, Feb. 2017, pp. 121–134.

[2] J. Wan, N. Zhao, Y. Zhu, J. Wang, Y. Mao, P. Chen, and C. Xie, "High Performance and High Capacity Hybrid Shingled-Recording Disk System," 09 2012, pp. 173–181.

[3] D. Sun and Y. Chai, "SAC: A Co-Design Cache Algorithm for Emerging SMR-Based High-Density Disks," ser. ASPLOS '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 1047–1061.

[4] A. Aghayev, M. Shafaei, and P. Desnoyers, "Skylight—A Window on Shingled Disk Operation," vol. 11, no. 4, oct 2015.

[5] W. Xiao, H. Dong, L. Ma, Z. Liu, and Z. Qiang, "HS-BAS: A hybrid storage system based on band awareness of Shingled Write Disk," in *IEEE 34th International Conference on Computer Design (ICCD)*, 2016.

[6] C. Ma, Z. Shen, L. Han, R. Chen, and Z. Shao, "FC: Built-in flash cache with fast cleaning for SMR storage systems," *J. Syst. Archit.*, vol. 98, pp. 214–220, 2019.

[7] SNIA-IOTTA, "MSR Cambridge Block I/O Traces," http://iotta.cs.hmc.edu/traces/block-io/388.

[8] Q. Zhang, X. Li, L. Wang, T. Zhang, Y. Wang, and Z. Shao, "Lazy-RTGC: a real-time lazy garbage collection mechanism with jointly optimizing average and worst performance for nand flash memory storage systems," *ACM Trans. Des. Autom. Electron. Syst.*, vol. 20, no. 3, jun 2015.

[9] Y. Wang, Z. Qin, R. Chen, Z. Shao, Q. Wang, S. Li, and L. T. Yang, "A Real-Time Flash Translation Layer for NAND Flash Memory Storage Systems," *IEEE Transactions on Multi-Scale Computing Systems*, vol. 2, no. 1, pp. 17–29, 2016.

[10] C. Ma, Z. Shen, L. Han, and Z. Shao, "RMW-F: A Design of RMW-Free Cache Using Built-in NAND-Flash for SMR Storage," vol. 18, no. 5s, oct 2019.

[11] C. Ma, Z. Shen, Y. Wang, and Z. Shao, "Alleviating Hot Data Write Back Effect for Shingled Magnetic Recording Storage Systems," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 38, no. 12, pp. 2243–2254, 2019.

[12] Y. Pan, Z. Jia, Z. Shen, B. Li, W. Chang, and Z. Shao, "Reinforcement Learning-Assisted Cache Cleaning to Mitigate Long-Tail Latency in DM-SMR," in *58th ACM/IEEE Design Automation Conference (DAC)*, 2021, pp. 103–108.

[13] C. Wang, D. Wang, Y. Chai, C. Wang, and D. Sun, "Larger cheaper but faster: SSD-SMR hybrid storage boosted by a new SMR-oriented cache framework," in *Proc. IEEE Symp. Mass Storage Syst. Technol.(MSST)*, 2017.

[14] C. Ma, Z. Zhou, Y. Wang, Y. Wang, and R. Mao, "MU-RMW: Minimizing Unnecessary RMW Operations in the Embedded Flash with SMR Disk," in *2022 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2022, pp. 490–495.

[15] Y. Song, Q. Li, Y. Lv, C. Li, and L. Shi, "DWR: Differential Wearing for Read Performance Optimization on High-Density NAND Flash Memory," in *2022 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2022, pp. 897–902.

[16] L. Shi, L. Luo, Y. Lv, S. Li, C. Li, and E. H.-M. Sha, "Understanding and Optimizing Hybrid SSD with High-Density and Low-Cost Flash Memory," in *IEEE 39th International Conference on Computer Design (ICCD)*, 2021, pp. 236–243.

[17] "Drive-Managed SMR Performance Model," http://sssl.ccs.neu.edu/skylight.

[18] Samsung-Corporation, "MT29F16G08CBACA 16GB MLC NAND Flash," https://datasheetspdf.com/pdf/843449/Micron/MT29F16G08CBACA/1.