2025 Design, Automation & Test in Europe Conference (DATE 2025)

EF-IMR: Embedded Flash with Interlaced Magnetic Recording Technology

Chenlin Ma [1], Xiaochuan Zheng [1], Kaoyi Sun [1], Tianyu Wang [1] and Yi Wang [1,2]
1. College of Computer Science and Software Engineering, Shenzhen University, China
2. State Key Laboratory of Radio Frequency Heterogeneous Integration
chenlin.ma@szu.edu.cn, {zhengxiaochuan2022,sunkaoyi2022}@email.szu.edu.cn, {tywang,yiwang}@szu.edu.cn

***Abstract*—Interlaced Magnetic Recording (IMR), a technology that improves storage density through track overlap, introduces significant latency due to Read-Modify-Write (RMW) operations. Writing to overlapped tracks affects underlying tracks, requiring additional I/O operations to read, back up, and rewrite them, resulting in significant head movement latency.**

We propose EF-IMR, a new architecture that ensures crash consistency in IMR while minimizing RMW latency and head movement. EF-IMR reduces head movement during RMW operations and decreases redundant RMW operations. Evaluations under real-world, intensive I/O workloads show that EF-IMR reduces RMW latency by 20.11% and head movement latency by 89.37% compared to existing methods.

***Index Terms*—Interlaced Magnetic Recording, Embedded Flash, Read Modify Write, Head Movement.**

I. Introduction

Shingled Magnetic Recording (SMR) improves cost-per-storage-unit by overlapping track but leads to performance degradation due to RMW operations [1], [2]. IMR, a new track-overlapping technology, performs better than SMR at the same storage density. However, it still suffers from performance degradation caused by RMW operations. When writing to the bottom track of IMR, the following steps are required: ***1) Read***: read the adjacent top tracks and back them up to the backup region; ***2) Modify***: modify the bottom track;***3) Write***: write the backed-up adjacent data back to its original position. In IMR, backing up data and writing it back incur latency due to head movement and read/write operations.

Most studies focus on reducing the frequency of RMW operations in IMR, with little consideration given to minimizing the latency generated from the RMW process. Several recent works have studied this issue. MOM [3] migrates the affected top track to the nearest free top track, partly reducing the latency caused by RMW. MOC [4] addresses the RMW issue caused by deleting data from the bottom track. MOC writes one of the affected top tracks to the deleted bottom track, reducing the latency caused by RMW. However, the other top track still requires extensive head movement for backup. Therefore, none of these studies minimizes head movements.

To minimize the head movement caused by RMW, this paper proposes a novel architecture, Embedded Flash with Interlaced Magnetic Recording (EF-IMR). EF-IMR integrates lightweight flash (1% of IMR capacity) into the IMR to leverage the low-latency writes of flash memory to back up the top track data during RMW operation. We also propose a rewrite strategy to avoid redundant RMW operations.

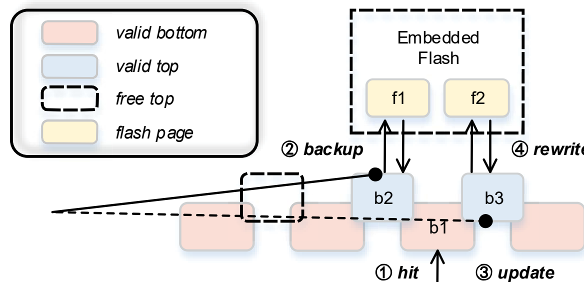

Fig. 1: The process of RMW Operation in EF-IMR

II. Design of EF-IMR

This paper presents EF-IMR to reduce the latency caused by RMW operations in IMR from both the system architecture and the system software levels. At the system architecture level, we propose a novel system architecture. IMR disk embedded with lightweight flash. Since the latency for reading and writing to flash is significantly lower than that of IMR and does not incur head movement, we embed lightweight flash to assist IMR in completing RMW operations. Unlike traditional IMR, where head movement is required between the target position and the backup region during RMW operations, EF-IMR, upon receiving a write command to the bottom track, reads the data from the adjacent top tracks and writes them to the flash location specified by the disk controller. After the IMR disk completes the write operation, the controller reads the backed-up data from the flash and instructs IMR to rewrite it back to its original position. In this process, head movement only occurs between adjacent tracks of the target position, significantly reducing the latency caused by head movement during RMW operations. Fig. 1 illustrates this process.

At the system software level, we propose a ***selective delay rewrite strategy*** to reduce redundant RMW operations, thus decreasing the latency caused by RMW. In this strategy, we selectively delay the ***Write*** phase of some RMW operations. Specifically, when a write operation to the target position triggers an RMW, we mark the data backed up to flash as invalid and temporarily delay the ***Write*** phase. When a subsequent write operation is performed on the target region, since the adjacent data are invalid, the write can be performed directly without triggering another RMW operation. The proposed EF-

978-3-9826741-0-0/DATE25/© 2025 EDAA

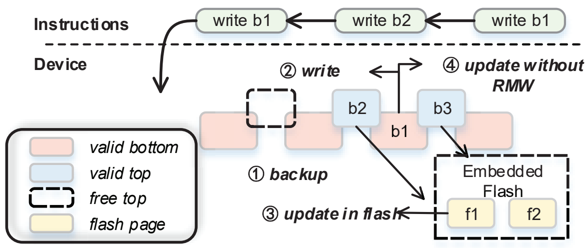

Fig. 2: Selective Delay Rewrite Strategy.

IMR only chooses to delay the write-back of top-track data to IMR for those tracks that have previously undergone RMW operations.

Fig. 2 illustrates how ***Selective Delay Rewrite Strategy*** reduces RMW. When writing to b1, b2, and b3 are backed up to flash, after b1 is written. b2 and b3 are kept in flash (when a write command for b2 is received, it is directly written to flash). As we write later in b1, it does not cause RMW because b2 and b3 are invalid.

III. Evaluation

A. Experimental Setup

We simulate an IMR disk embedded with flash based on the descriptions in [5]–[8]. Six workloads from MSR [9] are selected. We implement five methods: Seagate, MOM, MOC, EF-IMR, and EF-MOC, which run the MOC method under the EF-IMR architecture.

B. Performance Evaluation

a) ***Head Movement Latency****:* Fig. 3 show the total head movement latency of all RMW operations for EF-IMR and other methods across six traces. Compared to MOM, the most effective method in reducing head movement, on average, EF-IMR reduces head movement latency by 89.37%. Furthermore, with MOC, on average, EF-IMR reduces head movement latency by 99.73% compared to traditional IMR. The significant reduction in head movement is due to the EF-IMR strategy of backing up affected tracks to flash, limiting head movement to the target and adjacent tracks. Furthermore, the ***Selective Delay Rewrite Strategy*** reduces the occurrence of RMW operations, further lowering head movement latency.

b) ***AVG RMW Latency****:* Fig. 4 shows the RMW latency generated by the proposed EF-IMR compared to other methods. Across six traces, EF-IMR reduces RMW latency

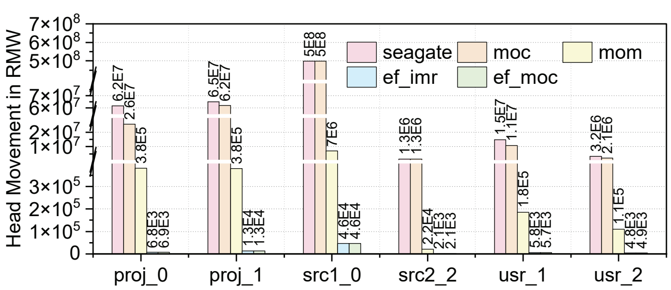

Fig. 3: Head Movement Latency caused by RMW Operation

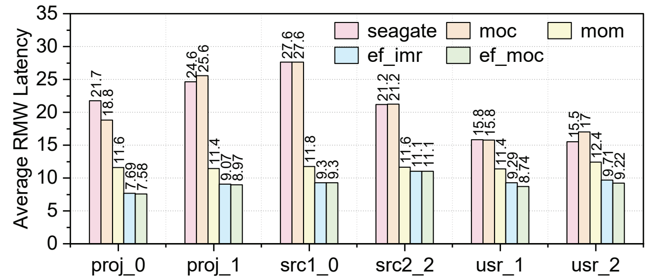

Fig. 4: Average RMW Latency.

by 20.11% on average. This result is attributed to the new architecture of EF-IMR, which reduces the latency caused by head movement. The proposed selective delay rewrite strategy further reduces redundant RMW operations. This strategy can decrease the latency of each RMW operation.

IV. Conclusion

In this paper, we proposed EF-IMR to minimize head movement and reduce RMW overhead. EF-IMR limits head movement to neighboring tracks and avoids redundant RMWs with selective delay rewrite strategy. Evaluation results show that EF-IMR effectively reduces RMW latency.

V. Acknowledgement

The work described in this paper is supported in part by NSFC (62122056, 62102263, and U23B2040), in part by Guangdong Basic and Applied Basic Research Foundation (2022A1515010180), in part by Shenzhen Science and Technology Program (RCJC20221008092725019, JCYJ20210324094208024, and 20220810144025001), in part by Guangdong Province Key Laboratory of Popular High Performance Computers (2017B030314073), in part by Guangdong Province Engineering Center of China-made High Performance Data Computing System and Shenzhen Key Laboratory of Service Computing and Applications. Yi Wang is the corresponding author.

References

[1] C. Ma, Y. Wang, Z. Shen, and Z. Shao, "KFR: Optimal Cache Management with K-Framed Reclamation for Drive-Managed SMR disks," in *DAC*, 2020, pp. 1–6.

[2] 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 *DATE*, 2022, pp. 490–495.

[3] Y. Liang and M.-C. Yang, "Move-On-Modify: An Efficient yet Crash-Consistent Update Strategy for Interlaced Magnetic Recording," in *DAC*, 2021, pp. 97–102.

[4] Z. Zeng, J. Cui, L. Wan, and L. T. Yang, "Optimizing Secure Deletion in Interlaced Magnetic Recording With Move-On-Cover Approach," *TACD*, pp. 1–5, 2024.

[5] E. Hwang, J. Park, R. Rauschmayer, and B. Wilson, "Interlaced Magnetic Recording," *IEEE Transactions on Magnetics*, vol. 53, no. 4, pp. 1–7, 2016.

[6] K. Gao, W. Zhu, and E. Gage, "Write Management for Interlaced Magnetic Recording Devices," Nov. 29 2016, uS Patent 9,508,362.

[7] K. Gao, W. Zhu, and E. Gage, "Interlaced Magnetic Recording," Aug. 8 2017, uS Patent 9,728,206.

[8] Z. Zeng, X. Chen, L. T. Yang, and J. Cui, "IMRSim: A Disk Simulator for Interlaced Magnetic Recording Technology," in *IFIP NPC*, 2022, pp. 267–273.

[9] "MSR Cambridge Block I/O Traces," http://iotta.cs.hmc.edu/traces/388.