Workload-aware Data-eviction Self-adjusting System of Multi-SCM Storage to Resolve Trade-off between SCM Data-retention Error and Storage System Performance

Reika Kinoshita[1], Chihiro Matsui[2], Atsuya Suzuki[1], Shouhei Fukuyama[1] and Ken Takeuchi[1,2]
[1]Department of Electrical, Electronic and Communication Engineering, Chuo University, Tokyo, Japan
[2]Research and Development initiative, Chuo University, Tokyo, Japan
E-mail: kinoshita@takeuchi-lab.org

Abstract - Storage Class Memories (SCMs) are used as non-volatile (NV) cache memory as well as storage. Multi-SCM storage with two types of SCMs, M-SCM (fast but small capacity memory-type SCM) and S-SCM (slow but large capacity storage-type SCM), has been proposed. In Multi-SCM storage, M-SCM works as NV-cache of S-SCM based storage. M-SCM such as MRAM is fast but may suffer from thermal instabilities and cause data-retention errors at high temperature. Therefore, data in M-SCM should be evicted to S-SCM at short interval before exceeding acceptable data-retention time. However, in case of short interval eviction, frequent data eviction from M-SCM to S-SCM severely degrades the storage system performance. To resolve this trade-off between data-retention reliability and the storage system performance, this paper proposes workload-aware data-eviction self-adjusting system. Proposed system is composed of Access Frequency Monitor (Proposal 1) and Evict Interval Adjustment (Proposal 2). Proposal 1 observes the access frequency of evicted data that directly affects data-retention time of M-SCM. By referring to the results of Proposal 1, Proposal 2 automatically changes the data-eviction interval so that long retention data are moved immediately to S-SCM and the storage system performance can be improved. As a result, maximum data-retention time of M-SCM decreases by 83%, and the storage system performance increases by 5.9 times. Moreover, the acceptable endurance increases by 10^3 times. Finally, measured data-retention errors and memory cell area decrease by 79% and 5.7%, respectively.

I Introduction

Storage Class Memories (SCMs) attract attention due to their higher performance than NAND flash memory and larger capacity than DRAM. Recently, hybrid storages with several types of non-volatile (NV) memory including SCMs have been proposed [1, 2]. SCMs can be categorized into two types, memory-type SCM (M-SCM) and storage-type SCM (S-SCM), based on their access speed, capacity and acceptable data-retention time. M-SCM like magnetoresistive RAM (MRAM) has fast access speed, small capacity and may suffer from short acceptable data-retention time at high temperature [3, 4]. On the other hand, S-SCM like 3D XPoint [5] has slow access speed, large capacity and long acceptable data-retention time. This paper adopts Multi-SCM storage [6] that uses both M-SCM and S-SCM. LPDDR2 is assumed for the interface of Multi-SCM storage. Multi-SCM storage uses M-SCM as NV-cache of S-SCM, and achieves 2.34 times higher performance than S-SCM only storage thanks to non-volatile write-back (NV-WB) algorithm [7]. Note that NV-WB stores frequently accessed data in M-SCM as shown in Fig. 1. However, data-retention characteristics of M-SCM is problematic at high temperature where acceptable data-retention time can be short, e.g., less than one day [4]. In

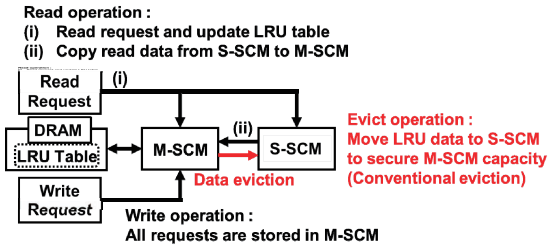
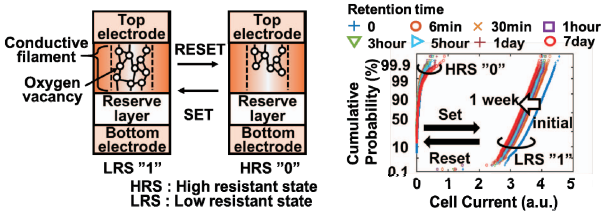
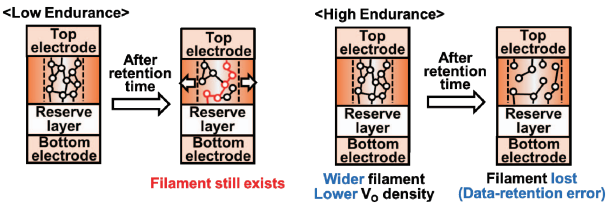


Fig. 1 NV-WB algorithm of Multi-SCM storage [6, 7].



(a) (b)

Fig. 2 (a) Physical model of ReRAM [8]. (b) Measured current distribution of HRS and LRS [10].



(a) (b)

Fig. 3 Mechanism of trade-off between endurance and data-retention in ReRAM with (a) low endurance and (b) high endurance stress [10].

this paper, resistive RAM (ReRAM) [8] is assumed for M-SCM, that has relatively longer acceptable data-retention time than MRAM and shorter latency than PRAM [9]. Fig. 2 illustrates the switching mechanism and measured current distribution of ReRAM. It is reported that data-retention characteristics of ReRAM become shorter at high Set/Reset endurance cycles [10, 11]. As endurance cycles increase, oxygen vacancies (V_O) diffuse and the conductive filament spreads horizontally in the low resistance state (LRS) of ReRAM as shown in Fig. 3. The endurance cycles of M-SCM become over 50 times higher than that of S-SCM [6]. Hence, if ReRAM is used as M-SCM with extremely high endurance cycles, ReRAM short retention should be resolved.

For DRAM and NAND flash memory, some researchers have proposed to solve data-retention failure from the aspect of device/circuit [12, 13] and system [14, 15]. However, for ReRAM, system-level approach has not been proposed while

978-1-7281-4123-7/20/$31.00 ©2020 IEEE

device-level [10, 16] and circuit-level solutions [17] to reduce data-retention errors have already been reported. Although previous work [18] proposed the algorithm which reduces the maximum data-retention time in M-SCM, this algorithm needs timestamp information that is not managed in storage controller. Therefore, this paper proposes system-based data-retention management without timestamp information for short retention M-SCM.

In case of DRAM, data in each cell are refreshed every 64 ms [19] because data-retention time becomes zero by overwrite access. While there is no endurance limit in case of DRAM, that of ReRAM is limited [10]. Thus, refresh that increases the endurance of ReRAM is not applied to NV-cache based on ReRAM. As an alternative to the refresh, we may use Periodic Eviction [18] that all data in M-SCM are moved to S-SCM at regular intervals. Data-retention time is controlled by the length of eviction interval, and eviction interval should be short to achieve high reliability of M-SCM. However, frequent eviction degrades the storage system performance because data eviction needs multiple read from M-SCM and write to S-SCM. Fig. 4 shows trade-off between device reliability and the storage system performance.

This work proposes data-eviction self-adjusting system of Multi-SCM storage (Fig. 5) to balance data-retention time and the storage system performance. Because long retention data should be evicted immediately from M-SCM to S-SCM, the proposal makes evict interval shorter. On the other hand, short retention data do not require frequent eviction. Therefore, evict interval is extended for data with short data-retention time to improve the storage system performance. This system is consisted of two proposals as described in Fig. 5: Proposal 1 is Access Frequency Monitor, and Proposal 2 is Evict Interval Adjustment.

The major contributions of this paper are as follows:

- Periodic Eviction is adopted for Multi-SCM storage to avoid data staying in M-SCM for a long time.
- We propose Access Frequency Monitor to estimate data-retention time of data in M-SCM (Proposal 1). Timestamp information is not provided to the storage/SSD controller and thus is not available. Proposal 1 estimates the approximate data-retention time in M-SCM based on data access frequencies without timestamp information.
- We also propose Evict Interval Adjustment (Proposal 2) that adjusts data-eviction interval to balance the storage system performance and device reliability.

By combining ReRAM device measurement and system-level storage simulation, the proposed system reduces maximum data-retention time of M-SCM by 83% and improves the storage system performance by 5.9 times. As for the ReRAM reliability, measured acceptable endurance increases by 10^3 times and measured data-retention errors decrease by 79%. Since the ReRAM cell reliability is improved, weaker error correcting code (ECC) with smaller parity overhead becomes acceptable. As a result, the memory cell area decreases by 5.7%.

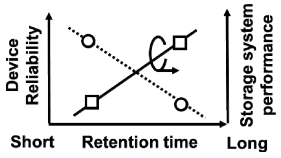


Fig. 4 Trade-off between SCM device reliability and storage system performance.

II. Two Proposals for Short Retention M-SCM

A. *Access Frequency Monitor (Proposal 1)*

Conventional Multi-SCM storage cannot manage data-retention time of each data because the storage controller does not handle timestamp information. In Proposal 1, write count (not read count) to M-SCM is regarded as pseudo time. To prevent data from remaining in M-SCM for a long time, all data in M-SCM are evicted to S-SCM every time storage receives certain count of write accesses to M-SCM (evict interval count). When data are evicted by Periodic Eviction, the number of evicted pages is monitored. Note that access unit of SCM is sector (512 Bytes), and the unit of data movement such as eviction is page (16 KBytes) in this paper. Additionally, "valid page" means the page where data exist. The page that contains no data is called "blank page." This paper reports that the number of evicted valid pages changes by average overwrite of workloads as indicated Fig. 5 [7]. For workloads with high average overwrite, accesses concentrate on small part of M-SCM. Thus, the number of valid pages in M-SCM becomes small. In contrast, for workloads with low average overwrite, the memory access distributes widely in M-SCM. Therefore, the number of valid pages in M-SCM becomes large. Moreover, average overwrite affects the data-retention time [20] because data-retention time becomes zero when memory cell is overwritten by new data. While pseudo retention time of frequently overwritten data is short, that of rarely overwritten data is long.

Fig. 6(a) shows the problem of conventional eviction triggered when M-SCM capacity becomes almost full. With

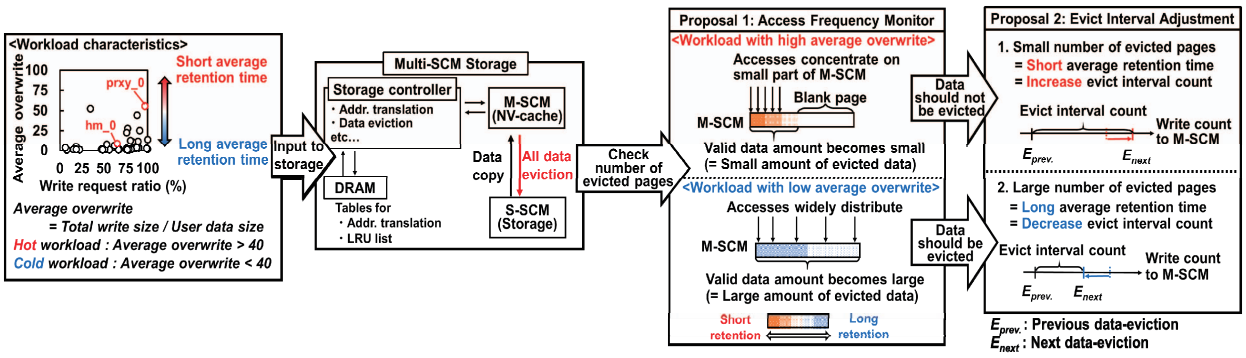


Fig. 5 Proposed data-eviction self-adjustment system of Multi-SCM storage.

conventional eviction, only least recently used (LRU) data are evicted, and long retention data remain in M-SCM. With proposed eviction, all data in M-SCM are evicted when write count to M-SCM exceeds evict interval count, and pseudo retention time is monitored by the number of evicted valid pages as shown in Fig. 6(b). When accesses concentrate on specific part of M-SCM, the number of evicted pages becomes small. On the other hand, when accesses are widely distributed in M-SCM, the number of evicted valid pages becomes large. Fig. 7 indicates the relationship between the number of evicted valid pages and the number of overwrite requests from the host when evict interval count is set to 1,000. As a result, data-retention time is judged as "short" when monitored

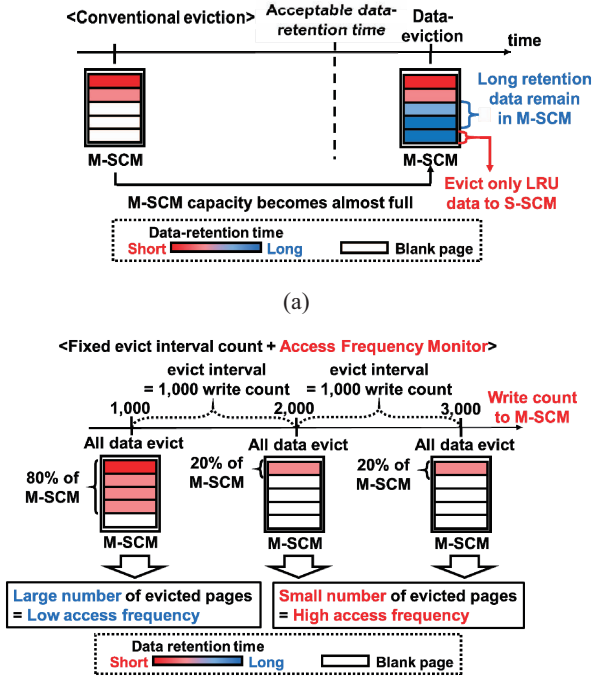


Fig. 6 (a) Problem of conventional data-eviction. (b) Proposal 1: Eviction Interval Management. Data are evicted with fixed evict interval count.

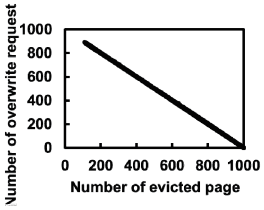


Fig. 7 Relationship between the number of evicted pages and the number of overwrite requests from the host.

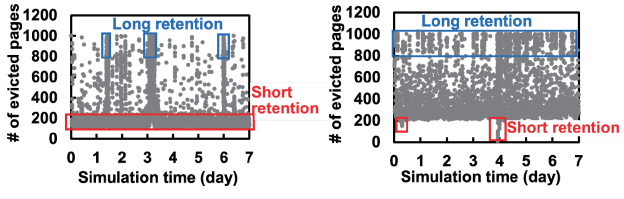


Fig. 8 Monitored number of evicted pages with fixed evict interval count of 1,000 for (a) prxy_0 write-hot workload and (b) hm_0 write-cold workload [7, 21].

number of evicted valid pages by Proposal 1 is less than 20% of evict interval count. In contrast, data-retention time is judged as "long" when monitored number of evicted valid pages is more than 80% of evict interval count. Fig. 8 shows monitored number of evicted valid pages for real workloads at data centers where evict interval count is fixed at 1,000 (fixed interval eviction). For evaluation, prxy_0 write-hot workload and hm_0 write-cold workload are evaluated [7, 21]. Both short retention and long retention data are mixed in two workloads and temporal localities exist. While many short retention data and few long retention data exist in prxy_0, few short retention and many middle/long retention data exist in hm_0. As shown in Fig. 8(a), prxy_0 is dominated by short retention data. In contrast, hm_0 is dominated by long retention data as shown in Fig. 8(b).

B. Evict Interval Adjustment (Proposal 2)

Proposal 2 self-adjusts data-eviction frequency from M-SCM to S-SCM based on monitored retention time by Proposal 1. When the number of evicted valid pages is small, data-retention time is short. Thus, evict interval count is set large to decrease evict frequency and improve the storage system performance. When the number of evicted pages is large, data-retention time is long. Therefore, evict interval count is set smaller to evict long retention data more frequently and prevent data-retention errors of M-SCM.

Fig. 9 illustrates self-adjusting evict interval count by monitoring number of evicted valid pages in detail. Initial evict interval count is set at 1,000 write counts, the minimum interval count. In case of small number of evicted valid pages, evicted data are judged as short retention data and evict interval count increases by N_{Adjust}. In case of large number of evicted valid pages, evicted data are judged as long retention data and evict interval count decreases, except when evict interval count is already set to minimum. Threshold to change the next evict interval count is shown in Fig. 10. Threshold itself is dynamically changed. Pseudo retention time is judged as short and N_{Adjust} increases when the number of evicted valid pages is less than 20% of evict interval count. On the other hand, pseudo retention time is judged as long and N_{Adjust} deceases when the number of evicted valid pages exceeds 80% of evict interval count. In other cases, pseudo retention time is judged as middle and evict interval count is not changed.

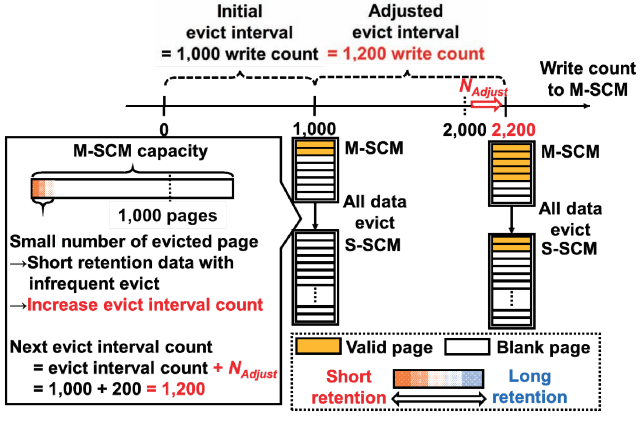


Fig. 9 Proposal 2: Eviction Interval Management that adjusts evict interval count.

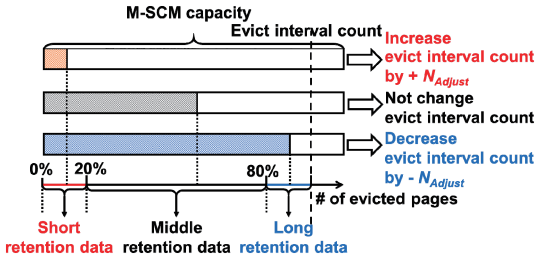


Fig. 10 Self-adjusting of data-eviction interval by dynamically changing threshold that adjusts next evict interval count.

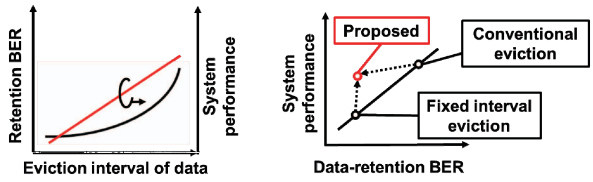


Fig. 11 (a) Effect of eviction interval on data-retention BER of SCM and storage system performance. (b) Comparison of proposed and conventional data-eviction.

Table 1 Comparison of data-eviction

	Evict interval	Evict frequency	Reliability	Performance
Conventional data eviction	Long	Low	Low	High
Fixed interval eviction (1,000 counts)	Short	High	High	Low
Proposed data eviction	Self-adjusting	Self-adjusting	High	High

Evict interval count influences both data-retention bit error rate (BER) and the storage system performance as shown in Fig. 11(a). Fig. 11(b) and Table 1 compare three types of eviction: conventional eviction (Fig. 6(a)), fixed interval eviction (Fig. 6(b)) and proposed eviction (Fig. 9). Conventional eviction has low evict frequency, low reliability and high performance. Fixed interval eviction has high evict frequency, high reliability but low performance. Finally, proposed eviction adjusts the evict frequency and thus achieves both high reliability and high performance.

C. Write-LRU for Selecting Victim Pages

In conventional eviction [7], the end page of LRU list is chosen as victim page (evicted page). When data in LRU list are written or read, data move to the top of LRU list as illustrated in Figs. 12(a) and 12(b). Because data-retention time is not reset by read, long retention data remain in M-SCM in conventional eviction as shown in Fig. 12(b). In proposed eviction, victim page is chosen from the end of Write-LRU (W-LRU) list, not LRU list, which ignores the multiple read access and manages only write access order to monitor pseudo retention time as shown in Fig. 12(c). That is, if data is read again, the location in LRU list stays at the same location. In addition to W-LRU, proposed system also uses LRU for NV-cache management of M-SCM. When cache miss happens and S-SCM is written or read, data in S-SCM are copied to M-SCM. Therefore, proposed system uses both LRU and W-LRU for NV-cache algorithm and data-retention time monitor, respectively, as shown in Table 2. Fig. 13 describes two types of proposed eviction trigger: one is when the number of write accesses reaches evict interval count

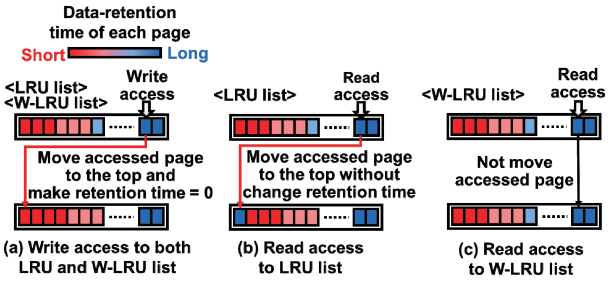


Fig. 12 (a) Write access to LRU and W-LRU list. Comparison of (b) conventional LRU and (c) proposed W-LRU.

Table 2 Conventional cache and proposed NV-cache.

	Cache algorithm	Data-retention monitor
Conventional cache	LRU (Read and Write)	--
Proposed NV-cache	LRU (Read and Write)	W-LRU (Only write request)

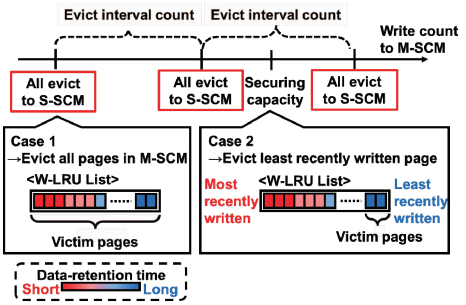


Fig. 13 Two Cases of proposed data-eviction. Case 1: Evict long retention data from M-SCM to S-SCM. Case 2: Evict all data when M-SCM becomes almost full.

(Case 1), and the other is when M-SCM capacity becomes almost full (Case 2). All pages in M-SCM are evicted with Case 1, and least recently written page is evicted with Case 2. Actually, most of triggered eviction is Case 1, and thus this paper optimizes Case 1 eviction with two proposals.

III. Evaluation Results

This paper uses system, circuit and device co-design (SCDCD) platform [22] that combines device measurement and transaction-level-modeling system emulator [1]. Characteristics of M-SCM and S-SCM are listed in Table 3 [6, 23]. For evaluation, M-SCM capacity is fixed at 10% of the whole storage capacity. Initial evict interval count is set at 1,000 write counts which is the minimum evict interval count. Then, N_{Adjust} is varied from 500 to 100,000. The evaluation results of proposals are compared with those of both conventional eviction (Conv.) and fixed interval eviction

Table 3 Memory characteristics [6, 23]

	M-SCM	S-SCM
Write latency	100 ns	10 us
Read latency	100 ns	10 us
I/O	1066 MHz	
V_{DD} (core, I/O)	1.8 V / 1.2 V	
Access unit	Sector (512 Bytes)	
Data movement unit	Page (16 KBytes)	
Storage capacity	User data size * 1.25 (Over provisioning : 25%)	
Capacity	10% of storage capacity	90% of storage capacity

(Fixed). With fixed interval eviction, the evict interval count is set to 1,000.

A. Results of Data-retention Time and System Performance

Fig. 14 shows the number of evicted valid pages and evicted interval count change when N_{Adjust} is set at 500. In Fig. 14(a), evict interval count continues to increase for prxy_0 workload. Fig. 15 plots all write requests to each logical address for one week. From Fig. 15(a), prxy_0 is dominated by write-hot access. Therefore, proposed system automatically increases evict interval count. In Fig. 14(b), for hm_0 workload, evict interval count is increasing during the first four days, and then evict interval count decreases rapidly

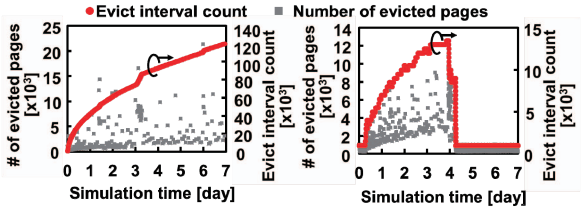


Fig. 14 Results of number of evicted pages for (a) prxy_0 write-hot workload and (b) hm_0 write-cold workload. Evict interval count for 1-week workloads when N_{Adjust} is set to 500. Evict interval count self-adjusts by monitoring hot/cold data in M-SCM.

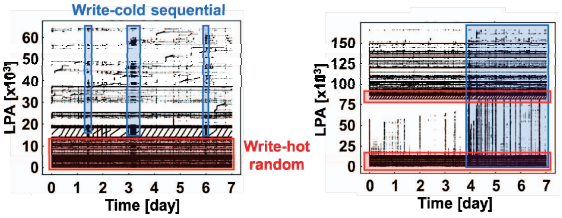


Fig. 15 Plots of write requests in one-week workloads for (a) prxy_0 and (b) hm_0.

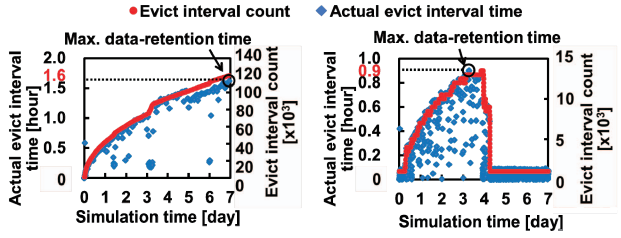


Fig. 16 Results of actual interval time for (a) prxy_0 and (b) hm_0. Maximum evict interval time is defined as maximum data-retention time.

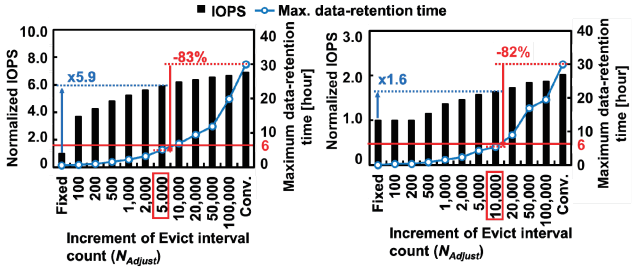


Fig. 17 Storage system performance and maximum data-retention time of SCM for (a) prxy_0 and (b) hm_0.

on the fourth day. Fig. 15(b) indicates that write-hot random accesses are dominant for the first four days. However, after the fourth day, write-cold sequential request appears and becomes dominant. Thus, proposed system reduces evict interval count after the fourth day to frequently evict cold data. In this way, the proposed system can self-adjust the data-eviction considering the temporal localities of data. However, if workloads with few write requests (such as proj_3) are used, eviction is not triggered in Multi-SCM and M-SCM needs refresh to remove long retention data. Fig. 16 shows the relationship between evict interval count and actual evict interval time. Actual evict interval time harmoniously changes similar to evict interval count. Maximum data-retention time is defined as the maximum evict interval time. Therefore, maximum data-retention time for prxy_0 and hm_0 is 1.6 hours and 0.9 hour, respectively.

Fig. 17 indicates the storage system performance, Input/Output per Second (IOPS) and maximum data-retention time of M-SCM with each N_{Adjust}. IOPS results are normalized by that of fixed interval eviction. The storage system performance of conventional eviction is 6.67 times higher than fixed interval eviction as shown in Fig. 17(a). With fixed interval eviction, frequent eviction largely decreases system performance. On the other hand, maximum data-retention time of fixed interval eviction is reduced to 0.57 hour while that of conventional eviction is 30 hours. For hm_0 in Fig. 17(b), conventional eviction increases system performance by 2.0 times of fixed interval eviction. Performance improvement of hm_0 is smaller than that of prxy_0 because evict interval count of hm_0 reduces from the fourth day, and data-eviction is triggered more frequently. Meanwhile, maximum data-retention time of fixed interval eviction decreases from 30 hours, that is the results of conventional eviction, to 72 seconds.

If the limit of data-retention time is assumed as 6 hours, N_{Adjust} that produces the highest performance is 5,000 for prxy_0. In the case, the storage performance improves 5.9 times compared with fixed interval eviction. The maximum data-retention time reduces by 83%, compared with conventional eviction. For hm_0, the system performance becomes the highest with 6 hours or less data-retention time when N_{Adjust} is set at 10,000. The performance enhances 1.6 times compared with fixed interval eviction. Additionally, the maximum data-retention time reduces 83% from conventional eviction.

B. Measured Device Reliability

Fig. 18 shows measured trade-off of acceptable data-retention time and endurance of ReRAM at accelerated measurement condition to intentionally increase errors [10]. Note that the reliability of LRS is solely analyzed in Fig. 18 because HRS is highly reliable [10]. Measured data-retention time with conventional eviction allows only 10^2 Set/Reset cycles. On the other hand, in the proposed self-adjusting system, the maximum data-retention time decreases by 83% for prxy_0 workload. Reduced maximum data-retention time enhances measured acceptable endurance to 10^5 Set/Reset cycles. Moreover, measured BER decreases by 79% as shown in Fig. 19 because data-retention time deceases by 83%. As a result, required strength of error-correcting code (ECC) is relaxed and ECC code rate increases by 5.7% as described in

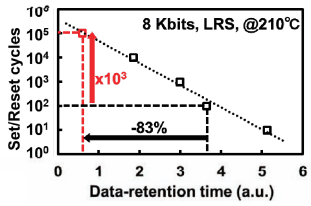


Fig. 18 Measured trade-off between data-retention time and endurance of ReRAM [10].

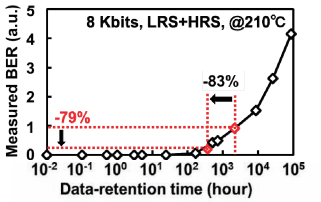


Fig. 19 Measured BER of ReRAM as a function of data-retention time.

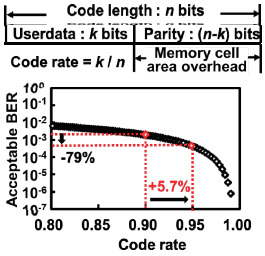


Fig. 20 Acceptable BER dependence on ECC code rate.

Fig. 20. Because parity bits of ECC corresponds to memory cell area overheads, memory cell area decreases by 5.7%.

IV. Conclusion

This paper proposes data-eviction self-adjusting system of Multi-SCM storage to resolve the trade-off of device reliability and system performance. When conventional eviction is applied to Multi-SCM storage, long retention data remain in M-SCM and that causes data-retention errors. If data eviction is performed at short intervals to reduce long retention data in M-SCM, frequently triggered data-eviction degrades the storage system performance. Proposed eviction is composed of Access Frequency Monitor and Evict Interval Adjustment. By using write count to M-SCM as pseudo time, pseudo retention time is successfully estimated. By self-adjusting data-eviction frequency, the storage system performance is improved by 5.9 times from fixed interval eviction. Additionally, acceptable data-retention time decreases by 83% from conventional eviction (Table 4). As a result, measured acceptable endurance increases from 10^2 to 10^5 cycles and measured BER decreases by 79%. Finally, memory cell area decreases by 5.7%.

Acknowledgement

This paper is based on results obtained from a project commissioned by New Energy and Industrial Technology Development Organization (NEDO). The authors thank S. Matsuda, H. Kinoshita and Y. Kakuta for their support.

Table 4 Summary of this work

	Conventional eviction [7]	Fixed evict interval count	Proposed eviction
Max retention time	30 hours	1 hours -83%	5.1 hours
System performance of storage	1.00	0.14 x5.9	0.86
Acceptable endurance	10^2 cycles	2.2 x 10^5 cycles x10^3	10^5 cycles
ECC code rate	0.90	0.96	0.95
		Memory cell area	

References

[1] H. Fujii *et al.*, "x11 performance increase, x6.9 endurance enhancement, 93% energy reduction of 3D TSV-integrated hybrid ReRAM/MLC NAND SSDs by data fragmentation suppression," *Symp. VLSI Circ.*, 2012, pp. 134-135.

[2] G. Sun *et al.*, "A hybrid Solid-State Storage Architecture for the Performance, Energy Consumption, and Lifetime Improvement," *HPCA*, 2010, pp. 1-12.

[3] H. Naeimi *et al.*, "STTRAM Scaling and Retention Failure," *Intel Technology Journal*, vol. 17, issue 1, pp. 54-75, 2013.

[4] S. Shiratake, "STT-MRAM Design and Device Requirement," *IEDM Tutorial*, 2018. (unpublished)

[5] Intel Optain Technology, https://ark.intel.com/content/www/us/en/ark/products/97159/intel-optane-ssd-dc-p4800x-series-1-5tb-1-2-height-pcie-x4-3d-xpoint.html

[6] R. Kinoshita *et al.*, "Maximizing Performance/cost Figure of Merit of Storage-type SCM based SSD by Adding Small Capacity of Memory-type SCM," *NVMTS*, 2018, pp. 1-6.

[7] S. Okamoto *et al.*, "Application driven SCM and NAND flash hybrid SSD design for data-centric computation system," *IMW,* 2015, pp. 157-160.

[8] Z. Wei *et al.*, "Highly Reliable TaOx ReRAM and Direct Evidence of Redox Reaction Mechanism," *IEDM*, 2008, pp. 293-296.

[9] S. Mittal and J. S. Vetter, "A Survey of Software Techniques for Using Non-Volatile Memories for Storage and Main Memory Systems," *TPDS*, vol. 27, no.5, pp. 1537-1550, 2016.

[10] S. Fukuyama *et al.*, "Comprehensive Analysis of Data-retention and Endurance Trade-off of 40nm TaOx-based ReRAM," *IRPS*, 2019, pp. 7C.3-1-7C.3-6.

[11] Y. Y. Chen *et al.*, "Understanding of the Endurance Failure in Scaled HgO2-based 1T1R RRAM through Vacancy Mobility degradation," *IEDM,* 2012, pp. 482-485.

[12] S. Ryu *et al.*, "Overcoming the Reliability Limitation in the Ultimately Scaled DRAM Using Silicon Migration Technique by Hydrogen Annealing," *IEDM*, 2017, pp. 21.6.1-21.6.4.

[13] E. Choi and S. Park, "Device Considerations for High Density and Highly Reliable 3D NAND Flash Cell in Near Future," *IEDM,* 2012, pp. 9.4.1-9.4.4.

[14] J. Liu *et al.*, "RAIDR: Retention-aware intelligent DRAM refresh," *ISCA,* 2012, pp. 1-12.

[15] Y. Cai *et al.*, "Flash correct-and refresh: Retention-aware error management for increased flash memory lifetime," *ICCD,* 2012, pp. 94-101.

[16] T. Ninomiya *et al.*, "Improvement of Data Retention During Long-Term Use by Suppressing Conductive Filament Expansion in TaOx Bipolar-ReRAM," *Electron Device Letters,* vol. 34, no. 6, pp. 762-764, 2013.

[17] M. F. Chang *et al.*, "A 130nm 1Mb HgOx Embedded RRAM Macro Using Sef-Adaptive Peripheral Circuit System Techniques for 1.6X Work Temperature Range," *A-SSCC*, 2017, pp. 173-176.

[18] A. Suzuki *et al.*, "Periodic Data Eviction Algorithm of SCM/NAND Flash Hybrid SSD with SCM Retention Time Constraint Capabilities at Extremely High Temperature," *NVMTS*, 2018, pp. 1-5.

[19] Micron Technology, Inc, 4Gb DDR3 SDRAM Component Data Sheet: MT41J512M8, 2009.

[20] Y. Yamaga *et al.*, "Application Optimized Adaptive ECC with Advanced LDPCs to Resolve Trade-off among Reliability, Performance, and Cost of Solid-State Drives," *IMW,* 2016, pp. 129-132.

[21] MSR Cambridge Traces, http://iotta.snia.org/traces/388.

[22] C. Matsui *et al.*, "Application-Induced Cell Reliability Variability-Aware Approximate Computing in TaOx-based ReRAM Data Center Storage for Machine Learning," *Symp. VLSI Tech.,* 2019, pp. T234-T235.

[23] C. Matsui and K. Takeuchi, "22% Higher Performance, 2x SCM Write Endurance Heterogeneous Storage with Dual Storage Class memory and NAND Flash," *ESSDERC,* 2017, pp. 6-9.