

¹Shweta¹P. K. Singh

MFGC: Minimal First Garbage Collection for Monitoring of Context- Aware Victim Selection



Abstract: - SSDs based on NAND flash are becoming a feasible secondary storage option for many computer systems. Because NAND flash memory is different from regular HDDs, flash-based SSDs frequently have an interface layer known as Flash Translation Layer (FTL). Garbage collection technique is required to free up space. Garbage collection policy is primarily concerned with cleaning costs and wear-leveling. To achieve the above design principles, we have proposed an effective minimal first garbage collection termed as MFGC. MFGC manages the blocks in the list, sorted in descending order of erase count. This list is divided into two regions: preference region and alternate region. The adaptive window for victim selection is used, which adapts dynamically as the cleaning cost changes. After the victim selection, effective data identification of valid hot page and valid cold page is done based on threshold and redistribution of them is done to free blocks. A Series of trace simulations is performed to evaluate the performance of proposed approach with other existing approaches. MFGC outperforms GR, CB, CAT, CATA and MCSGC by 85.5%, 77.1%, 61.7%, 56.7% and 43.5% respectively. The outcome of the simulation demonstrates that our approach is superior to others in terms of erase and copy operations as well as wear leveling of blocks.

Keywords: Garbage Collection, data identification, victim selection, lifetime, wear-leveling.

I. INTRODUCTION

Due to its favorable properties, such as shock to resistance, volatility in nature, compact size, and energy efficiency, flash memory gained popularity in consumer devices as a secondary storage device compared to magnetic disc [1]. With the introduction of flash memory chips capable of gigabytes, there is a much more significant trend for flash memory to replace magnetic discs as the storage media in electronic devices such as tablet PCs, PDAs, MP3 players, and smartphones [2][3][35]. When the available space in flash memory falls under a certain level, there is a need for an effective garbage collection policy to choose the victim block and migrate all valid pages of the chosen block to the specified available area, erasing the full victim block. The erase operation, which is performed on the block, is constrained [34]. In this instance, the garbage collection strategy should contain certain wear leveling-aware strategies to increase the extent of wear-leveling and uniformly distribute loads of all blocks in flash memory [4][5]. As a result, the cost of cleaning and the extent of wear-leveling are the most important considerations in garbage collection policies. An efficient minimal first garbage collection strategy termed MFGC is proposed to meet these two design concepts. In this paper, data are separated based on their lifetime for two different reasons.

The data lifespan is firstly appropriate for data separation [36][39]. A short lifespan indicates that data will be updated quickly, making them hot. Otherwise, the data is considered cold. Second, data lifetime seems more important than other metrics in representing access behavior since it may reveal data update trends [40].

For example, they are growing hotter given specific data as their data lifetime rapidly shortens. In comparison, the update frequency focuses on previous access times and does not accurately depict the current situation.

The following is a highlight of our contributions:

1. MFGC creates a list that connects all blocks in decreasing order of erase count and separates the list into two regions: Preference and Alternative.
2. MFGC uses an effective target block selection policy to select the target block for garbage collection (GC).
3. Data identification scheme is applied to cluster data on the basis of threshold into hot and cold.

Trace-driven simulations were used to test our proposed approach and it is figured out that it is more effective than existing garbage collection techniques in terms of copy operation, erase operation and the degree of wearleveling, according to the experimental results. The rest of this paper is laid out as follows. Background information on garbage collection is provided in Section 2. The purpose of the research is

¹Computer Science and Engineering Department, Madan Mohan Malaviya University Of Technology, Gorakhpur, 273010, Uttar Pradesh, India.

*Corresponding author(s). E-mail(s): Shweta20989@gmail.com; Contributing authors: topksingh@gmail.com;

explained in Section 3. Ongoing research in the field is the topic of Section 4. The proposed strategy MFGC is shown in Section 5. Section 6 provides the simulation result of the experiment. Section 7 finally concludes the paper.

II.BACKGROUND

NAND flash and NOR flash are two primary forms of flash memory. The bus interface is the most significant distinction between them. Although NOR memory, similar to SRAM, contains address and data buses that are directly coupled, NAND storage contains an interface for I/O which consist of control inputs[6][7][8][9]. Because of its increased density, cheaper cost, and quicker operations,

NAND memory is ideal for data storage. The random access speed and read operations of NAND is lower than NOR; therefore NOR flash is ideal execute-in-place (XIP) application[10][11]. The properties of flash memory vary in comparison to those of magnetic discs. To begin with, there is no delay connected to the head movement required to seek the right location to read or write data in flash memory[15][16][31]. Second, flash memory has asymmetric I/O operations as granularities of different operations are different, i.e. read and write operations are carried out on the page size whereas granularity for erase operation is block[24][27]. Third is write once policy in which data once written can not be updated until there is no privilege to update data unless the block residing it is erased [17][18]. Because block erases is time, it is usual to out-place update a page's data to an available page (i.e., an unused page that may be updated with data) to avoid deleting the block on each page update. Dead or invalid pages are those which are not updated frequently and are a better option for garbage collection[19][20][32][33].

Consequently, as the frequency of writes increases, so does the number of erases. The cost disparity would be substantially larger if erase operations was included. When we examine the possible erase cost associated with flash write actions, we can see that a flash write process is more than 8 times more expensive than a read. Finally, flash memory blocks are restricted for certain number of erase operations, after which blocks wear out and will be of no use[21][22][23]. As a result, a well-designed garbage-collecting strategy is required to wear down the flash memory area uniformly.

III.MOTIVATION

Issues of garbage collection:

Several points should be taken into consideration for designing effective cleaning policies, which are as follows:

When: Cleaning initiated and terminated?

Which: Block is picked for cleaning? Most garbage collection technique selects a block as a victim based on various information such as age, frequency, erase cost, etc.

How many: blocks must be brought to the cleaning process at once?

The amount of valid data that can be rearranged depends on how many blocks are cleaned at once. Nevertheless, cleaning of block requires a significant buffer to store valid pages of blocks. This also causes a considerable delay in the availability of free memory. After cleaning, blocks in cleaning could be removed or modified, which causes pointless migration.

How and where: where the pages of the victim block should be migrated? This is known as data redistribution, in which valid pages are migrated to some other blocks on the basis of certain conditions such as age, frequency, etc.

IV.RELATED WORK

Garbage collection occurs when a flash drive's available memory goes below a specific threshold. Garbage collection is commonly divided into the following stages, as seen in fig. 1. The first task is choosing target block. Second is to find the valid pages within the target block and then copy them to some free block. Finally, erase the selected blocks for further use[28][29][30].

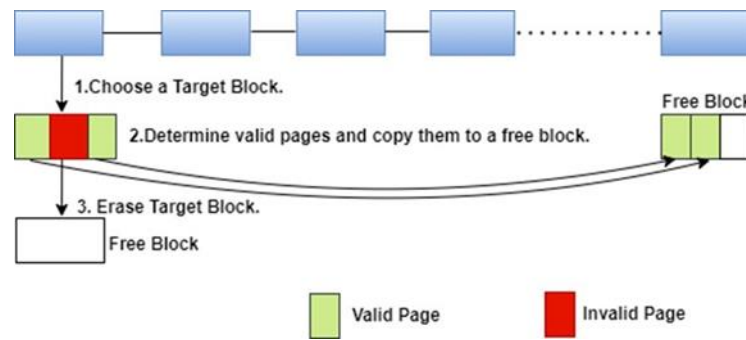


Figure 1 Garbage Collection Procedure

The garbage collection policy is primarily concerned with the expense of cleaning and the extent of wear leveling. As a result, an effective garbage collection approach should reduce the cleaning cost per valid page while wearing down all blocks as equally as feasible. A lot of cleaning policies have been presented to meet these two aims.

Random policy: Naively, one can choose a random block to clean. This straightforward solution may need minimum bookkeeping information, but it provides no way to limit the copy operation of valid pages in GC. Therefore, a consistently random selection approach can produce excellent wear-leveling results.

Wu et al suggested a NAND flash garbage collecting method [12] known as the greedy strategy. The amount of valid pages inside a block is the garbage collection overhead. It chooses the block as the victim who has the least valid pages, reducing garbage collection costs, and maximizing free space. In the GR approach, choosing a victim block ignores the block's erase count. Thus, it displays excessive wear leveling and reduces NAND flash memory lifespan. This approach requires more bookkeeping for counting invalid pages.

As its name suggests, the Least Recent Write (LRW) policy chooses the least recently used block for cleaning first, in the order in which they are written [37][38]. when a block is entirely written, it joins a FIFO queue called the LRW queue, and when it reaches the end, it is picked for cleaning. The LRW approach can guarantee perfect wear leveling across all physical blocks. Since there is no assurance that we have truly selected the block with the fewest active pages, the LRW approach may need more migrations than the greedy scheme.

A combination of the LRW and greedy approaches is Window-Greedy [40]. Similar to the LRW policy, it builds a block queue and only uses the greedy policy on the first B blocks in the queue. This increases the possibility of discovering a block with few active pages in relation to the LRW policy and avoids the possibly CPU-expensive greedy algorithm search.

The d-Choice strategy offers a trade-off between system performance and usability. It uses a selection window of size d and draws inspiration from the load-balancing selection method [32]. It determines a random subset of d from all blocks between the block with the most invalid pages to be selected.

The cost- benefit algorithm (CB) introduced by Kawaguchi et al [13] picks the block having the highest value as a victim block as calculated in equation 1.

$$age * (1 - v) \tag{1}$$

Where v is the proportion of valid pages inside the block and 2v reflects the reading cost of valid blocks and migrating them to the space available. The expression 1- v denotes the gain in free space gained during garbage collection. When selecting a target block, the cost-benefit analysis does not consider the block's erase quantity; as a result, wear leveling is still high in this case. A cleaning policy for flash-based consumer electronics devices was introduced by Chiang et al. known as Cost-age-time garbage collection (CAT)[14].It evolved from the CB method by incorporating each block's erase count. The CAT scheme selects a block using equation 2 as a target block.

$$\frac{v}{1 - v} * \frac{1}{age} * EC \tag{2}$$

Where v is the fraction of valid pages in the block and age is the period since the block's previous data invalidation. The EC refers to the block's erase count. The CAT technique differentiates hot valid pages from cold invalid pages and distributes them to various free blocks. Han et al proposed a garbage collection technique based on cost- age-time with age-sort (CATA) [25]. As a victim block, the CATA system chooses the block that maximises the value in equation 3.

$$\frac{1 - v}{1 + v} * age * \frac{1}{EC} \quad (3)$$

where v is the proportion of valid pages inside the block and age denotes the duration since the block's previous data invalidation. The word $erase\ count$ refers to the block's erase count. Hwang et al. devised the Migration Cost algorithm, MCSGC (Migration Cost Sensitive Garbage Collection) [26]. The MCSGC algorithm considers both recency and frequency for identifying hot and cold data, subsequently lowering the garbage collection overhead.

4 Minimal First Garbage Collection

An efficient garbage collecting method aims to increase block wear and reduce the price of cleaning policies compared to the amount of valid pages to extend the lifespan of flash memory. We present MFGC, a flash memory garbage collection technique to achieve these goals. As shown in Fig. 2 MFGC arranges the block in descending order of block erase count. Victim block selection is made based on number of valid pages. The blocks are divided into preference region and alternative region. The window size w represents the size of preference region.

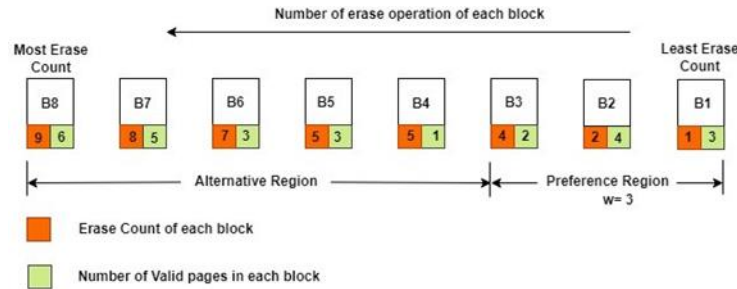


Fig. 2 Working of MFGC

To begin, MFGC selects an appropriate block as a victim using a target block selection algorithm described below. Selected Victim block might contain both hot and cold valid pages. MFGC uses data identification for differentiating between hot and cold valid pages. This process of identification and distribution may significantly delay garbage collection execution owing to fewer dirty blocks and lowers garbage collection cleaning costs.

Cleaning cost:-We begin by determining the cleaning cost; suppose for x^{th} garbage collection M_{Cx} and E_{Cx} represent the migration and erasure costs, respectively. The migration cost refers to cost of copying valid pages from the victim block to some other free space. The E_{Cx} represents the block erase count. Garbage collection cleaning cost GC_c may therefore be ted as follows:

$$GC_c = M_{Cx} + E_{Cx} \quad (4)$$

4.1 Target Block Selection(TBS)

The TBS (Target Block Selection) method efficiently lowers flash memory's garbage collection cleaning cost. The method works as follows:

In MFGC approach, an effective method for victim selection is used, which picks blocks from preference regions with few valid pages. If no such blocks are in the preference region, other blocks having lowest erase counts among the alternative region is selected as the victim. For example, as shown in Fig. 2, TBS selectively chooses block B3 as a target to decrease garbage collection cleaning costs, even though block B1 has a lower erase count than block B3. MFGC tries to maintain the performance trade-off between erase count and the amount of valid pages. After choosing the target blocks, the MFGC policy clusters the cold valid pages and hot valid pages separately based on a threshold, and the block is erased for further use. In summary, TBS with MFGC effectively reduces garbage collection's cleaning cost and improves flash memory's performance and lifespan.

4.2 Page Lifetime and threshold

Consider the lifetime of a page as the time period between the current time and the time when that particular page was written firstly. When such an incident happens, the previous entry stored in the mapping table is invalidated with the current one. In Fig. 3 it is depicted how the lifetime of pages is calculated. The lifetime is evaluated as the time between the instant time and when the page was written for the first time.

$$Lifetime = CT(i) - IT(i) \quad (5)$$

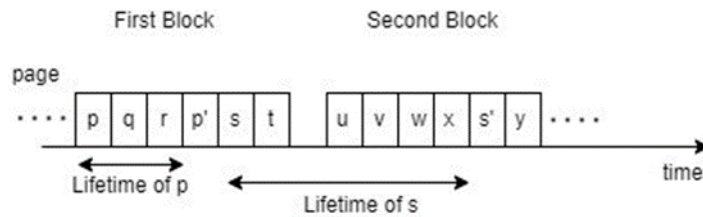


Figure. 3 Lifetime of page

4.3 Data Identification

Data detection for flash memory significantly influences garbage collection performance in flash memory. Placing data into separate blocks based on threshold can reduce the overhead of extra valid page copy operation. As shown in the fig. 4, data is placed without identification, so in the selected victim block, there are two pages for reclamation, and the other two pages had to be copied somewhere to erase the block. With the data identification scheme, one block is reclaimed as a target and no extra page copy operation is needed. Hence, the data identification scheme is reasonable and adequate compared to those without data identification approaches.

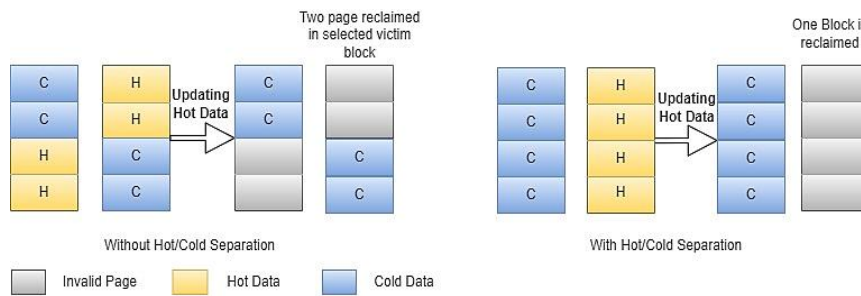


Figure 4 Data distribution with and without hot/cold separation.

This study recommends utilizing a lifetime as a threshold to identify data hotness. A page write request from a host is made up of one or more page writes. The requested page data are classified as hot data if the lifetime of page is less than the threshold. Otherwise, it is considered as cold.

When garbage collection occurs, the selected victim blocks data is identified for clustering into cold and hot blocks based on the abovementioned threshold. If the data is considered hot, it is transferred to a hot block; else, it is considered cold and migrated to a cold block. The cluster of the hot data block will soon be regarded as a victim as there will be frequent updation in the block data. As per hardware constraint, the erase operations per block are restricted; therefore, hot data are migrated to younger blocks with lower erasure counts. The cold data will be added to older blocks with higher erasure counts. This distribution tries to regulate the amount of wear leveling. As there will be little or no updation in cold block data, there will be no unnecessary page migration which will lessen the garbage collection in system and improves system performance. The data separation approach is depicted in Fig 5.

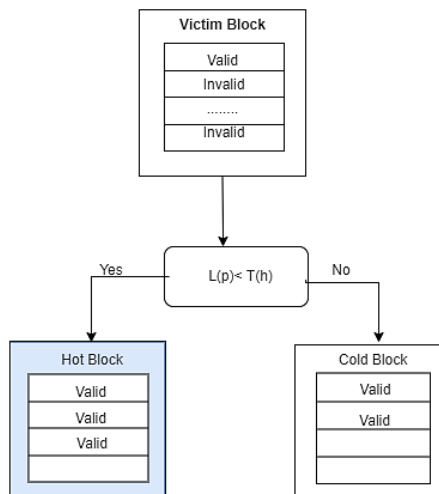


Figure. 5 Data identification method of MFGC.

V. EXPERIMENTAL RESULTS

5.1 Simulation Environment

We utilize an open-source simulator Disksim3.0 with some modifications [42], to analyze the flash memory performance since it is a commonly used simulation platform. A 32GB NAND flash memory is used in our experiment for simulation, and the parameters are shown in Table 1.

Parameter Used	
Memory capacity	32GB
Pages per Block	64
Size of Page	2KB
Latency of Page Read	29 μ s
Latency of Page Write	220 μ s
Latency of Block Erase	2000 μ s

The three traces used in the trace-driven trials were collected from Syllab of FIU [43] and span three weeks. These were obtained from a virtual machine that was operating a storage server (Homes), an email server (Mail), and two web servers (Web-vm). Table 2 [41][43] shows the features of the three traces.

Table 2 Characteristics of workload

Traces	Avg. Request Size	Write Ratio
Web-vm	40.8KB	78.5%
Mail	14.8KB	69.8%
Homes	13.1KB	80.5%

existing policies: GR stands for greedy policy ; CB stands for cost benefit policy ; CAT stands for cost age time policy ; CATA stands for cost-agetime with age-sort policy and MCSGC is migration cost sensitive garbage collection policy.

5.2 Experimental Result

Experiments are conducted and the metrics taken for performance evaluations are number of copy operation, erase operation and extent of wearing blocks.

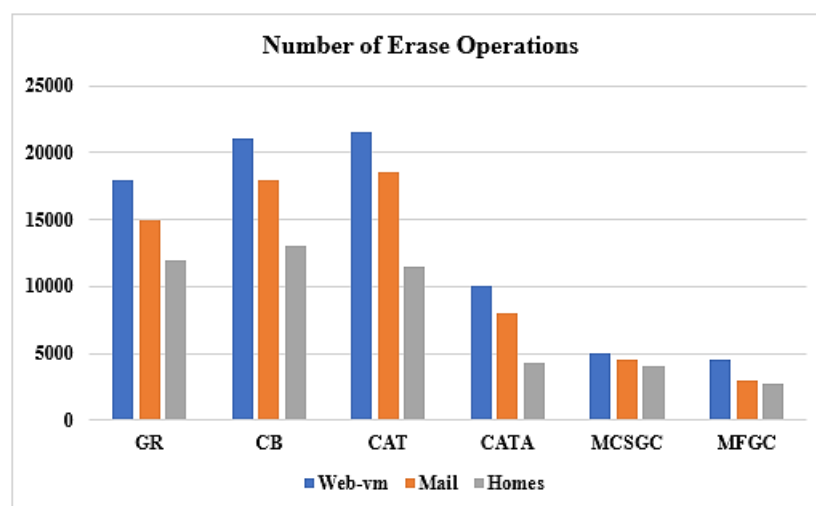


Figure 6 Erase Operation for various garbage collection policies.

Fig. 6 represents the performance evaluation of different garbage-collection algorithms with the MFGC in the context of the erase operation performed. MFGC picks the block as the target block with the least amount of valid pages to reduce the garbage collection overhead. Furthermore, it differentiates hot valid pages from cold valid ones inside the target block, then allocates them to young and old blocks respectively. As a result, the

proposed MFGC scheme generates significantly high storage space and requires significantly least cleaning operations compared to existing garbage collection techniques. Hence in comparison to other garbage collection policies, our proposed approach MFGC produces the least erase operation.

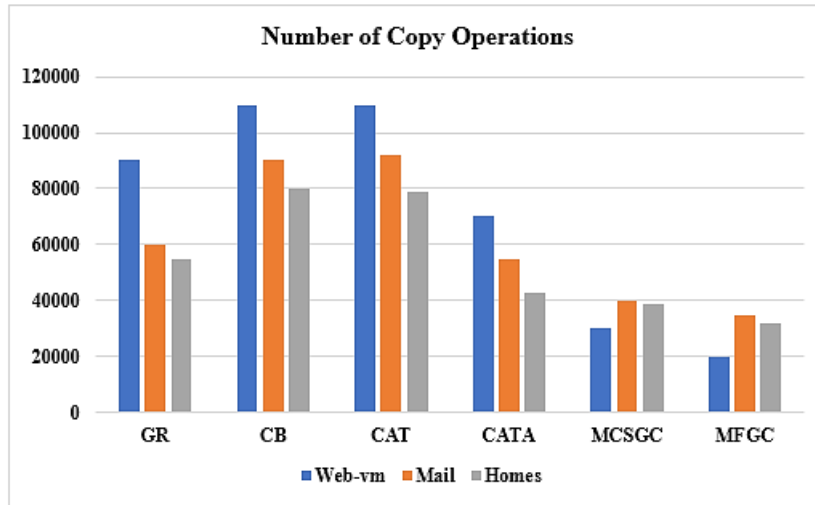


Figure. 7 Copy Operation for various garbage collection policies.

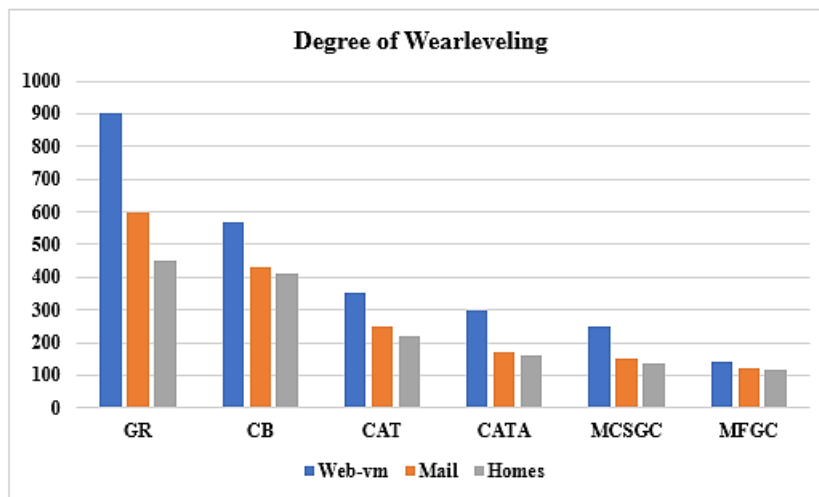


Figure. 8 Wear-leveling for various garbage collection policies.

In relation to the copy operations, fig.7 displays the effectiveness of evaluating the MFGC with different garbage collection policies. The proposed MFGC approach outperforms various policies, as illustrated in the graph. The MFGC policy expells the blocks with the lowest valid pages in the preference area; if no such block is available in the preference region, the MFGC strategy selects the block having the lowest erase counts from the remaining blocks. Fig.8 represents the comparison graph for wear leveling. The improvement in the degree of wear leveling is calculated using equation 6.

$$dw_i(\%) = \frac{dw_p - dw_e}{dw_e} * 100 \tag{6}$$

where dw_i =Improvement in the degree of wear-leveling, dw_p : wear-leveling in the proposed approach, dw_e : wear-leveling in the existing approach. Fig. 9 depicts the improvement ratio for wear-leveling; the MFGC policy out- performs GB, CB, CAT, CATA, and MCSGC by 85.5%,77.1%,61.7%,56.7% and 43.5%, respectively.

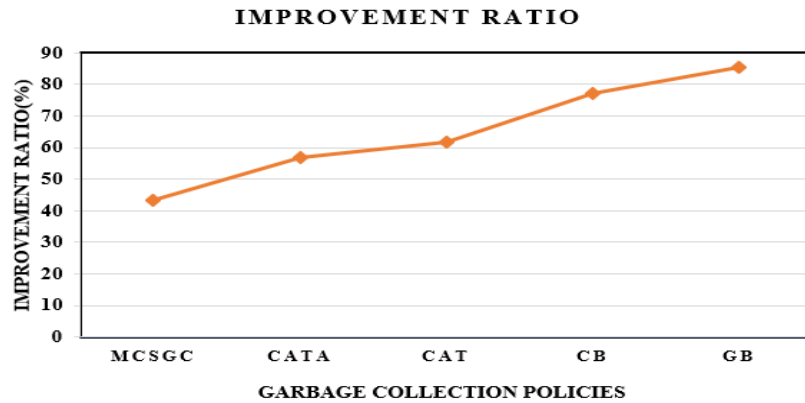


Figure. 9 Improvement Ratio(%).

VI.CONCLUSION

This paper presents a garbage collection technique known as MFGC to enhance the longevity of flash memory. MFGC connects the blocks in the order of their erase count and splits them into preference region and alternative region. MFGC first presents an effective target selection mechanism for victim blocks that first evicts the block in the preference area with the fewest valid ges.If no blocks are in the preference area, the remaining blocks with the lowest erase count are chosen as a target. During the garbage collection procedure, the migration of valid pages is led by an efficient data identification and redistribution mechanism to differentiate hot valid and cold valid pages; further, it distributes them to young/old blocks. We carried out a series of tests and found that the results were promising as compared to existing approaches. The MFGC policy surpasses the GB, CB, CAT, CATA, and MMSGC policies in terms of wear leveling by 85.5 %, 77.1%, 61.7 %, 56.7%, and 43.5%, respectively.

Conflict of Interest:

The author certify that there is no conflict of interest for this paper.

Compliance with Ethical Standards:

Yes

REFERENCES

- [1] A. Leventhal, "Flash storage memory," *Communications of the ACM*, vol. 51, no. 7, pp. 47–51, 2008.
- [2] O. Kwon and K. Koh, "Swap space management technique for portable consumer electronics with nand flash memory," *IEEE Transactions on Consumer Electronics*, vol. 56, no. 3, pp. 1524–1531, 2010.
- [3] H.-L. Li, C.-L. Yang, and H.-W. Tseng, "Energy-aware flash memory management in virtual memory system," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 16, no. 8, pp. 952–964, 2008.
- [4] P. Yang, N. Xue, Y. Zhang, Y. Zhou, L. Sun, W. Chen, Z. Chen, W. Xia, J. Li, and K. Kwon, "Reducing garbage collection overhead in SSD based on workload prediction," in *11th USENIX Workshop on Hot Topics in Storage and File Systems (HotStorage 19)*, 2019.
- [5] C. Du, Y. Yao, J. Zhou, and X. Xu, "Vbms: A novel buffer management strategy for nand flash storage devices," *IEEE Transactions on Consumer Electronics*, vol. 65, no. 2, pp. 134–141, 2019.
- [6] Narayan, Vipul, and A. K. Daniel. "IOT based sensor monitoring system for smart complex and shopping malls." *Mobile Networks and Management: 11th EAI International Conference, MONAMI 2021, Virtual Event, October 27-29, 2021, Proceedings*. Cham: Springer International Publishing, 2022.
- [7] Narayan, Vipul, and A. K. Daniel. "FBCHS: Fuzzy Based Cluster Head Selection Protocol to Enhance Network Lifetime of WSN." *ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal* 11.3 (2022): 285-307.
- [8] Narayan, Vipul, and A. K. Daniel. "RBCHS: Region-based cluster head selection protocol in wireless sensor network." *Proceedings of Integrated Intelligence Enable Networks and Computing: IIENC 2020*. Springer Singapore, 2021.
- [9] Narayan, Vipul, and A. K. Daniel. "A novel approach for cluster head selection using trust function in WSN." *Scalable Computing: Practice and Experience* 22.1 (2021): 1-13.

- [10] M. Santarini, "Nand versus nor-which flash is best for bootin'your next system?," *Edn*, vol. 50, no. 21, pp. 41–49, 2005.
- [11] S. L. Min and E. H. Nam, "Current trends in flash memory technology," in *Proceedings of the 2006 Asia and South Pacific Design Automation Conference*, pp. 332–333, 2006.
- [12] W. Bux and I. Iliadis, "Performance of greedy garbage collection in flash-based solid-state drives," *Performance Evaluation*, vol. 67, no. 11, pp. 1172–1186, 2010.
- [13] A. Kawaguchi, S. Nishioka, and H. Motoda, "A flash-memory based file system.," in *USENIX*, pp. 155–164, 1995.
- [14] M.-L. Chiang and R.-C. Chang, "Cleaning policies in mobile computers using flash memory," *Journal of Systems and Software*, vol. 48, no. 3, pp. 213–231, 1999.
- [15] Kuo, Tei-Wei, et al. "Special issues in flash." *2008 IEEE/ACM International Conference on Computer-Aided Design*. IEEE, 2008.
- [16] Ma, Dongzhe, Jianhua Feng, and Guoliang Li. "A survey of address translation technologies for flash memories." *ACM Computing Surveys (CSUR)* 46.3 (2014): 1-39.
- [17] Chang, Li-Pin. "On efficient wear leveling for large-scale flash-memory storage systems." *Proceedings of the 2007 ACM symposium on Applied computing*. 2007.
- [18] Hu, Xiao-Yu, Robert Haas, and Eleftheriou Evangelos. "Container marking: Combining data placement, garbage collection and wear levelling for flash." *2011 IEEE 19th Annual International Symposium on Modelling, Analysis, and Simulation of Computer and Telecommunication Systems*. IEEE, 2011.
- [19] Wang, Chundong, and Weng-Fai Wong. "Observational wear leveling: an efficient algorithm for flash memory management." *Proceedings of the 49th Annual Design Automation Conference*. 2012.
- [20] Lee, Sungjin, Dongkun Shin, and Jihong Kim. "BAGC: Buffer-aware garbage collection for flash-based storage systems." *IEEE Transactions on Computers* 62.11 (2012): 2141-2154.
- [21] Ji, Seunggu, and Dongkun Shin. "An efficient garbage collection for flash memory-based virtual memory systems." *IEEE Transactions on Consumer Electronics* 56.4 (2010): 2355-2363.
- [22] Liu, Jiakun, and Wonyong Sung. "COPR: a cost-oriented recycling policy for flash translation layer." *IEEE Transactions on Consumer Electronics* 56.2 (2010): 673-681.
- [23] Wang, Meng, et al. "An efficient NAND flash garbage collection algorithm based on area and block operation." *2012 Second International Conference on Intelligent System Design and Engineering Application*. IEEE, 2012.
- [24] Murugan, Muthukumar, and David HC Du. "Rejuvenator: A static wear leveling algorithm for NAND flash memory with minimized overhead." *2011 IEEE 27th Symposium on Mass Storage Systems and Technologies (MSST)*. IEEE, 2011.
- [25] L. Han, Y. Ryu, and K. Yim, "Cata: A garbage collection scheme for flash memory file systems," in *International Conference on Ubiquitous Intelligence and Computing*, pp. 103–112, Springer, 2006.
- [26] S.-H. Hwang, J. H. Choi, and J. W. Kwak, "Migration cost sensitive garbage collection technique for non-volatile memory systems," *IEICE TRANSACTIONS on Information and Systems*, vol. 99, no. 12, pp. 3177–3180, 2016.
- [27] Jung, Sanghyuk, and Yong Ho Song. "LINK-GC: a preemptive approach for garbage collection in NAND flash storages." *Proceedings of the 28th Annual ACM Symposium on Applied Computing*. 2013.
- [28] Tsao, Che-Wei, Yuan-Hao Chang, and Ming-Chang Yang. "Performance enhancement of garbage collection for flash storage devices: An efficient victim block selection design." *Proceedings of the 50th Annual Design Automation Conference*. 2013.
- [29] Gao, Congming, et al. "Exploiting parallelism in I/O scheduling for access conflict minimization in flash-based solid state drives." *2014 30th Symposium on Mass Storage Systems and Technologies (MSST)*. IEEE, 2014.
- [30] Kwon, Ohhoon, et al. "FeGC: An efficient garbage collection scheme for flash memory based storage systems." *Journal of Systems and Software* 84.9 (2011): 1507-1523.
- [31] Mitzenmacher, Michael. "The power of two choices in randomized load balancing." *IEEE Transactions on Parallel and Distributed Systems* 12.10 (2001): 1094-1104.
- [32] Kwon, Ohhoon, Yeonseung Ryu, and Kern Koh. "An efficient garbage collection policy for flash memory based swap systems." *Computational Science and Its Applications—ICCSA 2007: International Conference, Kuala Lumpur, Malaysia, August 26-29, 2007. Proceedings, Part I* 7. Springer Berlin Heidelberg, 2007.

- [33] Ahn, Sungyong, et al. "A compressed file system manager for flash memory based consumer electronics devices." *IEEE Transactions on Consumer Electronics* 59.3 (2013): 544-549.
- [34] Lin, Wen-Huei, and Li-Pin Chang. "Dual greedy: Adaptive garbage collection for page-mapping solid-state disks." *2012 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2012.
- [35] Liu, Jun, et al. "A novel hot data identification mechanism for NAND flash memory." *IEEE Transactions on Consumer Electronics* 61.4 (2015): 463-469.
- [36] Luo, Xiang, and Brian M. Kurkoski. "An improved analytic expression for write amplification in NAND flash." *2012 International Conference on Computing, Networking and Communications (ICNC)*. IEEE, 2012.
- [37] Desnoyers, Peter. "Analytic modeling of SSD write performance." *Proceedings of the 5th Annual International Systems and Storage Conference*. 2012.
- [38] Xu, Guangxia, Manman Wang, and Yanbing Liu. "Swap-aware garbage collection algorithm for NAND flash-based consumer electronics." *IEEE Transactions on Consumer Electronics* 60.1 (2014): 60-65.
- [39] Yan, Hua, and Qian Yao. "An efficient file-aware garbage collection algorithm for NAND flash-based consumer electronics." *IEEE Transactions on Consumer Electronics* 60.4 (2014): 623-627.
- [40] Wu, Suzhen, et al. "Cagc: A content-aware garbage collection scheme for ultra-low latency flash-based ssds." *2021 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*. IEEE, 2021.
- [41] Koller, Ricardo, and Raju Rangaswami. "I/O deduplication: Utilizing content similarity to improve I/O performance." *ACM Transactions on Storage (TOS)* 6.3 (2010): 1-26.
- [42] Bucy, John S., and Gregory R. Ganger. *The DiskSim simulation environment version 3.0 reference manual*. School of Computer Science, Carnegie Mellon University, 2003.
- [43] FIU IODedup traces. <http://iotta.snia.org/traces/391>.
- [44] Narayan, Vipul, et al. "7 Extracting business methodology: using artificial intelligence-based method." *Semantic Intelligent Computing and Applications* 16 (2023): 123.
- [45] Narayan, Vipul, et al. "A Comprehensive Review of Various Approach for Medical Image Segmentation and Disease Prediction." *Wireless Personal Communications* 132.3 (2023): 1819-1848.
- [46] Mall, Pawan Kumar, et al. "Rank Based Two Stage Semi-Supervised Deep Learning Model for X-Ray Images Classification: AN APPROACH TOWARD TAGGING UNLABELED MEDICAL DATASET." *Journal of Scientific & Industrial Research (JSIR)* 82.08 (2023): 818-830.
- [47] Narayan, Vipul, et al. "Severity of Lumpy Disease detection based on Deep Learning Technique." *2023 International Conference on Disruptive Technologies (ICDT)*. IEEE, 2023.
- [48] Saxena, Aditya, et al. "Comparative Analysis Of AI Regression And Classification Models For Predicting House Damages In Nepal: Proposed Architectures And Techniques." *Journal of Pharmaceutical Negative Results* (2022): 6203-6215.
- [49] Kumar, Vaibhav, et al. "A Machine Learning Approach For Predicting Onset And Progression" "Towards Early Detection Of Chronic Diseases " " *Journal of Pharmaceutical Negative Results* (2022): 6195-6202.
- [50] Chaturvedi, Pooja, Ajai Kumar Daniel, and Vipul Narayan. "Coverage Prediction for Target Coverage in WSN Using Machine Learning Approaches." (2021).
- [51] Chaturvedi, Pooja, A. K. Daniel, and Vipul Narayan. "A Novel Heuristic for Maximizing Lifetime of Target Coverage in Wireless Sensor Networks." *Advanced Wireless Communication and Sensor Networks*. Chapman and Hall/CRC 227-242.