

## Intelligent Garbage Collection Policy Based on I/O Workload Prediction for NAND Flash-based Storage Devices

Xiaobo Ji<sup>1</sup>, Fan Zeng<sup>2</sup>, Mingwei Lin<sup>3</sup> and Fei Gao<sup>4</sup>

<sup>1</sup>Department of Information, Research Institute of Field Surgery, Daping Hospital, Third Military Medical University, Chongqing, China

<sup>2</sup>Department of Information, Research Institute of Field Surgery, Daping Hospital, Third Military Medical University, Chongqing, China

<sup>3</sup>Faculty of Software, Fujian Normal University, Fuzhou 350007, China

<sup>4</sup>Faculty of Software, Fujian Normal University, Fuzhou 350007, China

<sup>1</sup>bati0716@126.com, <sup>2</sup>zengfan215@163.com, <sup>3</sup>linmwcs@163.com,

<sup>4</sup>1432220483@qq.com

### Abstract

NAND flash memory exhibits a large number of advantages such as high random access performance, small size, high reliability, and strong shock resistance, it has been widely used as the dominant storage device for consumer electronics. However, NAND flash memory uses an out-of-place update scheme to solve its erase-before-write hardware constraint and then garbage collection operations should be performed to reclaim invalid pages that are incurred by its out-of-place update scheme. Garbage collection operations usually incur high garbage collection overhead and shorten the lifetime of NAND flash memory. In this case, this paper proposes an intelligent garbage collection policy based on the I/O workload prediction for NAND flash-based storage devices. The proposed intelligent garbage collection policy adopts an improved exponentially weighted moving average method to predict the I/O workload and determines the number of victim blocks that should be reclaimed. In order to lower the degree of wear leveling for NAND flash memory, the proposed intelligent garbage collection policy first reclaims young victim block candidates with low erase count. A series of experiments are conducted with real traces on a commercial smart phone and experimental results show that our proposed intelligent garbage collection policy is superior to previous garbage collection policies in terms of the number of copy operations, the number of erase operations, the number of erase operations, the energy consumption, and the degree of wear leveling.

**Keywords:** NAND flash memory, Garbage collection policy, I/O workload prediction, Out-of-place update scheme, Exponentially weighted moving average

### 1. Introduction

IBM designed the first commercial computer RAMAC, which introduced a moving-head hard disk drive for secondary storage in 1956 [1]. Since then, the hard disk drive has been the dominant data storage medium for personal computers and servers [2]. However, the hard disk drive shows some disadvantages of large volume, high power consumption, weak shock resistance, and loud noise. Therefore, the hard disk could not satisfy the requirements for data storage in mobile communication, data collection, and aeronautics and space fields [3]. At the same time, mechanical properties of hard disk restrict the improvement of I/O performance. In the past 20 years, the CPU speed increases 500 times, while the I/O speed of hard disk only increases 20 times. The speed gap between the CPU

and hard disk increases continuously, it has influenced the overall performance of storage systems significantly [4].

In this case, flash memory was introduced by Toshiba in 1984. It can effectively solve the disadvantages of hard disk drive and exhibits a large number of advantages in terms of small size, low power consumption, strong shock resistance, and high random access performance. It has been widely used in consumer electronics for data storage, such as smart phones, digital cameras, tablet devices, notebook computers, and so on [5]. As the semiconductor technology develops continuously, flash memory dramatically increases in storage capacity and decreases in cost-per-bit [6]. The Samsung Electronics company has produced 128GB flash memory. Therefore, it has started to be used as secondary storage for desktop computers and servers. For example, it has been produced in the form of solid state drives and the solid state drives have been used to store data for newly produced computers such as Mac computers [7].

However, flash memory shows an erase-before-write hardware constraint so that it cannot be overwritten [8]. Flash memory uses an out-of-place update scheme to solve this hardware constraint. The out-of-place update scheme writes the new data into the free space of flash memory and then invalidates the original data as invalid. As time goes by, the free space will reduce gradually and then the invalid pages will increase gradually. When the free space falls below a predefined threshold, garbage collection operations should be incurred to reclaim the garbage in terms of invalid pages. Garbage collection operations can not only generate high garbage collection overhead, but also raise the degree of wear leveling. Therefore, a number of garbage collection policies have been proposed to reduce the garbage collection overhead and lower the degree of wear leveling. However, current garbage collection policies focus on designing an efficient victim block selection algorithm for selecting victim blocks and none of them focuses on determining how many victim blocks should be selected for reclaiming during the garbage collection operation [9].

In this paper, an intelligent garbage collection policy based on I/O workload prediction is proposed for NAND flash-based storage devices. The proposed intelligent garbage collection policy first introduces an improved exponentially weighted moving average method to predict the future I/O workload and then determines the number of victim blocks. Then, the proposed intelligent garbage collection policy introduces an efficient victim block selection algorithm for selecting victim blocks. Finally, the proposed intelligent garbage collection policy adopts an efficient data migration algorithm for migrating valid pages within victim blocks. A series of experiments are conducted with real traces on a commercial smart phone and experimental results evaluate the effectiveness of the proposed intelligent garbage collection policy.

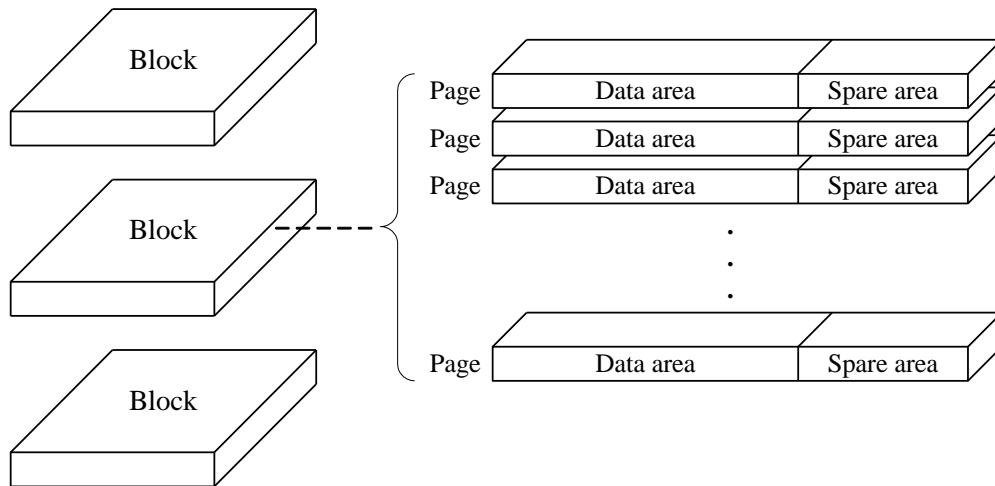
The remainder of the paper is organized as follows. Section 2 gives the overview of NAND flash memory. Section 3 briefly reviews existing works on garbage collection policy. Section 4 presents the typical system architecture of NAND flash-based storage system. Section 5 describes the proposed intelligent garbage collection policy in detail. Experimental results are presented in Section 6. Finally, conclusions are drawn in Section 7.

## **2. Overview of NAND Flash Memory**

Flash memory was designed from Electrically Erasable Programmable Read-only Memory (EEPROM) in 1984 and inherited features from EEPROM [10]. Hence, flash memory is a kind of electronic non-volatile memory storage medium that can be electrically erased and reprogrammed. There are two main kinds of flash memory, which are NOR flash memory and NAND flash memory, respectively [11]. They have different physical characteristics and they are often used for different purposes.

NOR flash memory shows small storage capacity. Moreover, it is very costly. However, it offers an execute-in-place (XIP) capability that allows applications to be executed directly in NOR flash memory instead of loading them into the main memory. Therefore, it is often used to replace ROM (Read Only Memory) for storing the BIOS of computers and the firmware of consumer electronics [12].

NAND flash memory has large storage capacity, low cost-per-bit, and high write and erase performance. Therefore, it is usually used as data storage medium for desktop computers and consumer electronics. As NAND flash memory increases in storage capacity and decreases in cost-per-bit continuously, NAND flash memory is introduced by enterprises to store data for servers, especially for read intensive applications [13].



**Figure 1. Architecture of a NAND Flash Memory Chip**

As shown in Figure 1, a NAND flash memory chip is composed of a fixed number of blocks. Each block also consists of a fixed number of pages. A page is usually divided into two parts, which are data area and spare area, respectively. The data area is adopted for storing user data, while the spare data is adopted to store the error correction code. NAND flash memory offers three basic I/O operations, which are the read, write, and erase operations, respectively. Read operation is performed to read data from a target page, while write operation is often incurred to write data to a target page. Hence, the basic unit of read and write operations is a page. The erase operation is usually triggered during garbage collection to erase a victim block after valid pages in the victim block are copied to free space. Hence, the basic unit of erase operation is a block [14-16].

NAND flash memory also shows a number of unique physical characteristics. First, it has an erase-before-write hardware constraint, which requires that a block must be erased before it is written. In this case, NAND flash memory usually adopts an out-of-place update scheme to solve this hardware constraint. This out-of-place update scheme writes the new data to the free space of NAND flash memory and then invalidates the original pages containing old data as invalid. Invalid pages are usually referred to as garbage. Second, NAND flash memory has asymmetric I/O costs. The cost of write operation is much higher than that of read operation, while the cost of erase operation is higher than that of write operation. Third, the maximum erase count allowed to each block within NAND flash memory is limited, typically 10,000 ~ 100,000. In order to avoid wear out the blocks within NAND flash memory, a large number of wear leveling algorithms have been proposed to erase all blocks as evenly as possible [17].

### 3. Related Works

NAND flash memory has an erase-before-write hardware constraint, which requires each block within NAND flash memory to be erased before being written. In this case, it adopts an out-of-place update scheme to solve this hardware constraint. The out-of-place update scheme first writes the new data to the free space of NAND flash memory and then invalidates the pages containing the original data as invalid. As time goes by, the free space will be reduced gradually and the invalid pages will be increased gradually. When the free space falls below a predefined threshold, garbage collection operations should be triggered to reclaim garbage in terms of invalid pages. Garbage collection operations often incur very high garbage collection overhead and also shorten the lifetime of NAND flash memory. Hence, reducing the garbage collection overhead and lowering the degree of wear leveling of NAND flash memory are the design principles of garbage collection policies. In order to achieve these design principles, a large number of garbage collection policies have been proposed.

Wu *et al.* proposed an efficient garbage collection policy for a non-volatile storage system [18]. The proposed efficient garbage collection policy is called GR policy and it is the first garbage collection policy proposed for NAND flash-based storage devices. The GR policy defines the garbage collection overhead as the number of valid pages within each block and then selects the blocks with the least number of valid pages as victim blocks. The GR policy can minimize the garbage collection overhead, but it does not consider the erase count of each block when selecting victim blocks. Therefore, the GR policy is very likely to increase the degree of wear leveling and then shorten the lifetime of NAND flash memory.

In order to solve the problems of GR policy and then lower the degree of wear leveling of NAND flash memory, Kawaguchi *et al.* proposed an efficient garbage collection policy called cost-benefit (CB) policy for NAND flash file systems [19]. The CB policy computes the cost-benefit ratio of each block and chooses the block with the highest cost-benefit ratio as a victim block. The cost-benefit ratio of each block is computed as

$$\frac{a \times (1-u)}{1+u} \quad (1)$$

where the terms  $a$  and  $u$  are the elapsed time since the last data invalidation on this block and the percentage of valid pages within this block. The term  $(1-u)$  represents the free space that can be obtained during garbage collection operation, while the term  $(1+u)$  represents the garbage collection overhead that is incurred. The CB policy uses the elapsed time since the last data invalidation to compute the cost-benefit ratio. In this case, the blocks that have been invalidated for a long time have a relatively high chance to be selected as victim blocks and the degree of wear leveling of NAND flash memory will be reduced. However, the CB policy also does not consider the erase count of each block when selecting victim blocks as the GR policy.

In order to further lower the degree of wear leveling of NAND flash memory, Chiang *et al.* proposed an efficient garbage collection policy called cost-age-times (CAT) for NAND flash memory storage systems [20]. The CAT policy considers the number of valid pages, elapsed time since the last data invalidation, and erase count of each block when selecting victim blocks. The CAT policy evicts the block that minimizes the following formula.

$$\frac{u}{1-u} \times \frac{1}{a} \times \text{erase\_count} \quad (2)$$

where the term  $u$  represents the number of valid pages, which is also considered as the garbage collection overhead. The term  $1-u$  represents the free space that is obtained.

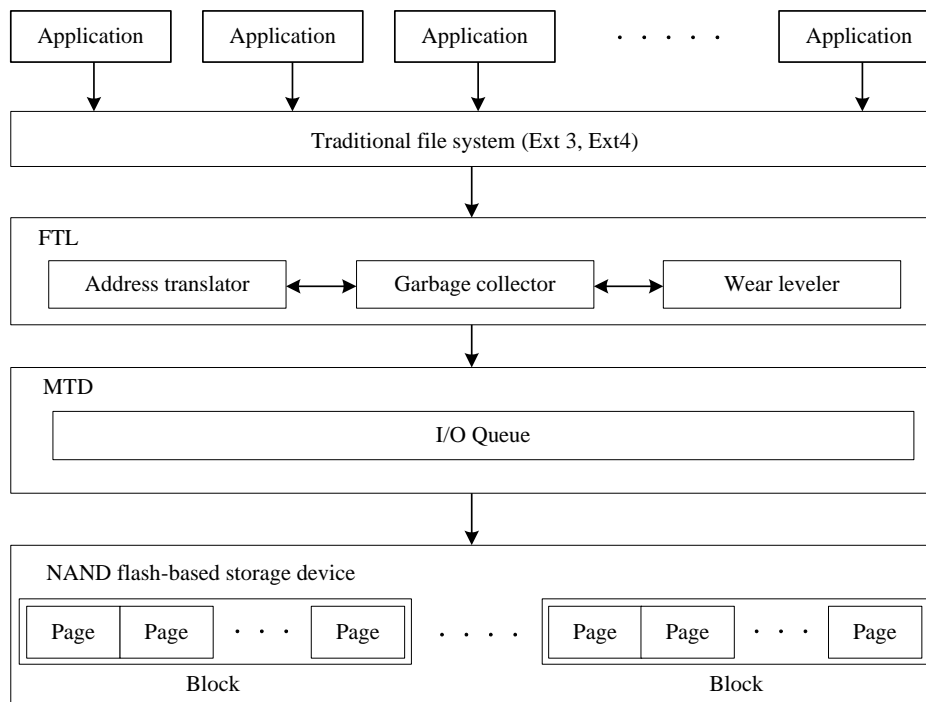
The term  $a$  is the elapsed time since the last data invalidation on this block. The term *erase\_count* represents the erase count of the block. The CAT policy considers the erase count of each block when selecting victim blocks. In this case, the blocks with low erase count will be reclaimed before the blocks with high erase count and then the degree of wear leveling will be lowered.

We can find that all the existing garbage collection policies focus on selecting a suitable victim block in order to reduce the garbage collection overhead and lower the degree of wear leveling of NAND flash memory chips. None of them focus on determining how many victim blocks should be selected during garbage collection operation.

#### 4. System Architecture of NAND Flash-Based Storage Systems

In this section, a typical system architecture of NAND flash-based storage systems will be described.

NAND flash memory shows very different hardware characteristics from hard disk drives, so NAND flash-based storage systems often introduce an additional software layer to operate NAND flash-based storage devices. This additional software layer is called flash translation layer (FTL). As shown in Figure 2, the typical system architecture of NAND flash-based storage systems is composed of the traditional file system, flash translation layer, memory technology device (MTD), and NAND flash-based storage device, respectively.



**Figure 2. System Architecture of NAND Flash-Based Storage Systems**

The traditional file system can be Ext2, Ext3, Ext4. The flash translation layer is usually adopted to deploy between traditional file system and memory technology device. The flash translation layer is responsible for hiding the unique hardware characteristics of NAND flash memory and emulating the NAND flash-based storage devices as block devices. In this case, traditional file system can operate NAND flash-based storage devices like hard disk drives. The memory technology devices is a generic subsystem in

Linux operating system that is used to access non-volatile memory devices such as NAND flash memory.

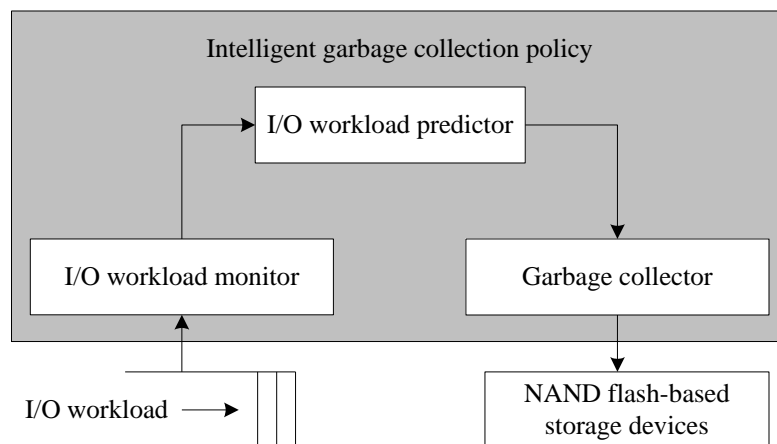
The flash translation layer is composed of address translator, garbage collector, and wear leveler. Address translator translates the logical block addresses from traditional file systems into the physical block addresses in the NAND flash-based storage devices. Garbage collector uses a garbage collection policy to reclaim garbage in terms of invalid pages. Wear leveler is a module used to wear out each block as evenly as possible in order to extend the lifetime of NAND flash-based storage devices. In this paper, we focus on designing an efficient garbage collection policy for NAND flash-based storage systems.

## 5. Intelligent Garbage Collection Policy Based on I/O Workload Prediction

In this section, the proposed intelligent garbage collection policy based on I/O workload prediction is described in detail.

### 5.1. Architecture

As shown in Figure 3, the proposed intelligent garbage collection policy is composed of three components, which are I/O workload monitor, I/O workload predictor, and garbage collector. The I/O workload monitor is in charge of measuring and saving the I/O workload. The I/O workload predictor is responsible of extracting the measurements of I/O workload from the I/O workload monitor to predict the I/O workload in the near future. The garbage collector is a module that selects and reclaims victim blocks.



**Figure 3. Architecture of the Proposed Intelligent Garbage Collection Policy**

In this paper, the I/O workload predictor introduces an improved exponentially weighted moving average method to predict the I/O workload in the near future and then determines the number of victim blocks that should be selected. The garbage collector uses a window-based victim block selection algorithm to select victim blocks and an efficient valid data migration algorithm to migrate valid data to free space.

### 5.2. I/O Workload Prediction

The exponentially weighted moving average method has been introduced to predict the I/O workload. The exponentially weighted moving average method predicts the I/O workload as follows [21].

$$E(t) = \alpha * E(t-1) + (1-\alpha) * O(t) \quad (3)$$

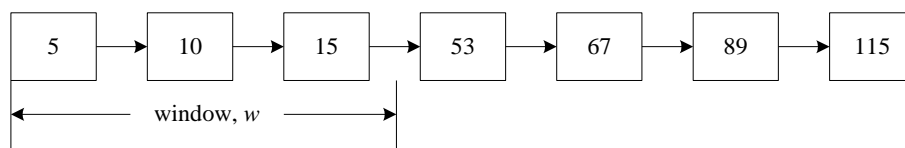
where the term  $E(t-1)$  is the predicted value of I/O workload at time  $(t-1)$ . The terms  $E(t)$  and  $O(t)$  are the predicted value of I/O workload and the observed value of I/O workload at time  $t$ , respectively. The coefficient  $\alpha$  represents the degree of weighting decrease and it is also a constant smoothing factor between 0 and 1.

However, the exponentially weighted moving average (EWMA) method cannot capture the rising trends of I/O workload. For instance, there is a time series of  $O(t)$   $\{50, 60, 70, 80\}$ , it is reasonable to predict the next value of I/O workload to be 90. But the coefficient  $\alpha$  is between 0 and 1, the predicted value of I/O workload is always between 50 and 80. It is impossible equal to 90. In order to capture the rising trends of I/O workload, we modify the original exponentially weighted moving average method by setting the coefficient  $\alpha$  to be between -1 and 0. Then, the formula (3) can be transformed as follows.

$$\begin{aligned} E(t) &= -|\alpha| * E(t-1) + (1+|\alpha|) * O(t) \\ &= O(t) + |\alpha| * [O(t) - E(t-1)] \end{aligned} \quad (4)$$

### 5.3. Window-Based Victim Block Selection Algorithm

The window-based victim block selection algorithm is in charge of selecting victim blocks. As shown in Figure 4, it maintains a list of victim block candidates in ascending order of erase count. The leftmost victim block candidate has lowest erase count, while the rightmost victim block candidate has highest erase count. The block list has a window that contains the victim block candidates with low erase count. The size of the window is  $w$ .



**Figure 4. An Example of Window-Based Victim Block Selection Algorithm**

The victim block candidates within the window are considered as young blocks, while the rest ones out of the window are called old blocks. The design principal of the window-based victim block selection algorithm is to first select the young victim block candidates as victim blocks. Therefore, the window-based victim block selection algorithm first checks the victim block candidates in the window and then selects the block with the lowest eviction value as a victim block. The eviction value of each block is calculated as

$$ev = \sum_{i=1}^n \frac{age_i - age_{\min}}{age_{\max} - age_{\min}} \quad (5)$$

where the term  $ev$  represents the eviction value of this block. The term  $n$  is the number of invalid pages within the block. The term  $age_i$  represents the elapsed time since the last invalidation on the  $i$ th invalid page. The terms  $age_{max}$  and  $age_{min}$  are the maximum elapsed time since the last invalidation on all the invalid pages within the block and the minimum elapsed time since the last invalidation on all the invalid pages in the block.

#### 5.4. Valid Data Migration Algorithm

After the victim blocks are selected, a valid data migration algorithm should be introduced to migrate valid data within the victim blocks. The victim blocks usually contain hot and cold valid data at the same time. Because hot valid data are often invalidated before cold valid data, migrating hot and cold valid data into the same block incurs new garbage collection overhead in terms of copy operations that migrating cold valid data into free space. In order to reduce the garbage collection overhead of NAND flash memory, the valid data migration algorithm separates the hot valid data from the cold valid data and then distributes them to different free blocks. Copying hot valid data to old blocks with high erase count will cause that old blocks will be erased frequently and the degree of wear leveling will be increased. Hence, the valid data migration algorithm distributes hot valid data to young blocks with low erase count and cold valid data to old blocks in order to avoid increasing the degree of wear leveling and then extend the lifetime of NAND flash-based storage device.

### 6. Experimental Results

In this section, experimental results are presented to assess the performance effectiveness of the proposed intelligent garbage collection policy based I/O workload prediction.

In order to evaluate the performance of the proposed intelligent garbage collection policy based on I/O workload prediction, a series of experiments are conducted with a real trace that is collected when executing a music player on the Linux operating system. The real trace of the music player shows a high locality of references in that 83.6% of write operations are made to metadata. Out proposed intelligent garbage collection policy based I/O workload prediction is compared with the greedy (GR) policy, cost-benefit (CB) policy, and cost-age-times (CAT) policy, respectively.

Experiments are conducted on a real smart phone. The smart phone is equipped with a 1 GHz microprocessor, 512 RAM, and 1 GB NAND flash memory.

The performance metrics used in the experiments are the number of copy operations, the number of erase operations, the energy consumption, and the degree of wear leveling.

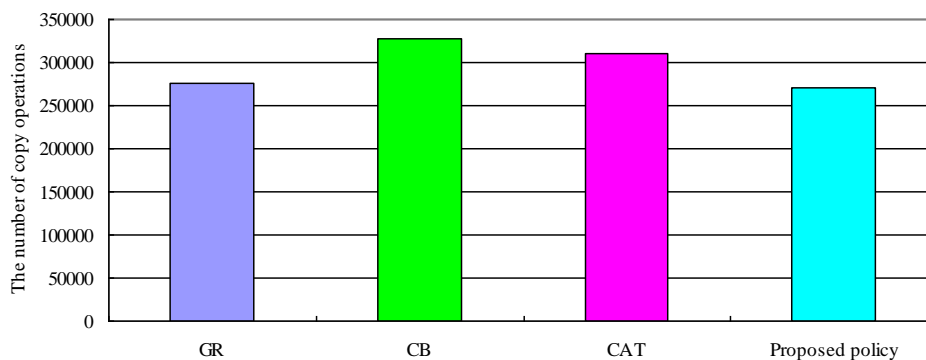
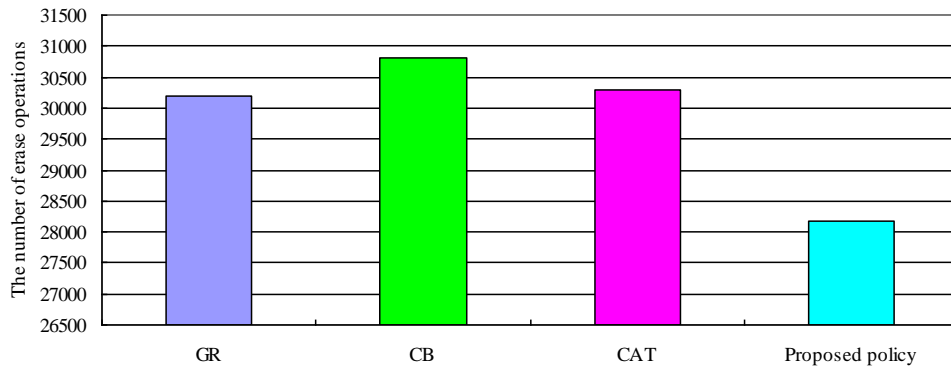


Figure 5. The Number of Copy Operations for Various Garbage Collection Policies

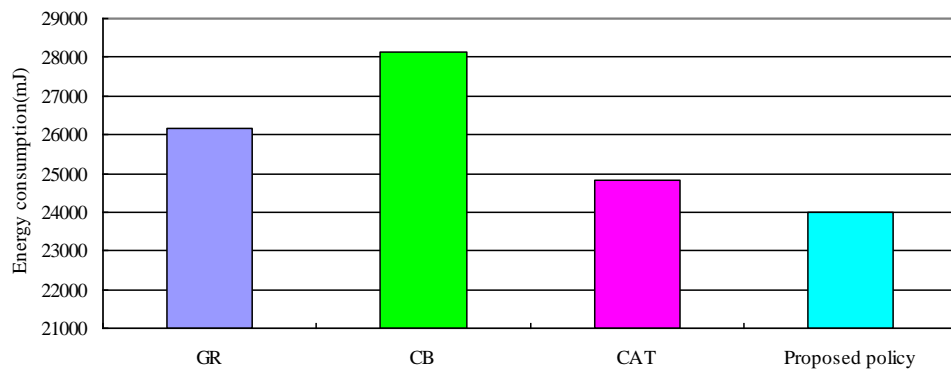


Figure 5 illustrates the experimental results of four garbage collection policies in terms of the number of copy operations. The GR policy selects the block with the lowest erase count as a victim block, so it incurs less copy operations than the CB and CAT policies. Our proposed garbage collection policy not only selects the block with the highest eviction value as a victim block, but also separates hot valid data from cold valid data within victim blocks and copies them to different free blocks. In this case, copy operations are reduced significantly. Hence, our proposed garbage collection policy incurs less copy operations than the GR policy.



**Figure 6. The Number of Erase Operations for Various Garbage Collection Policies**

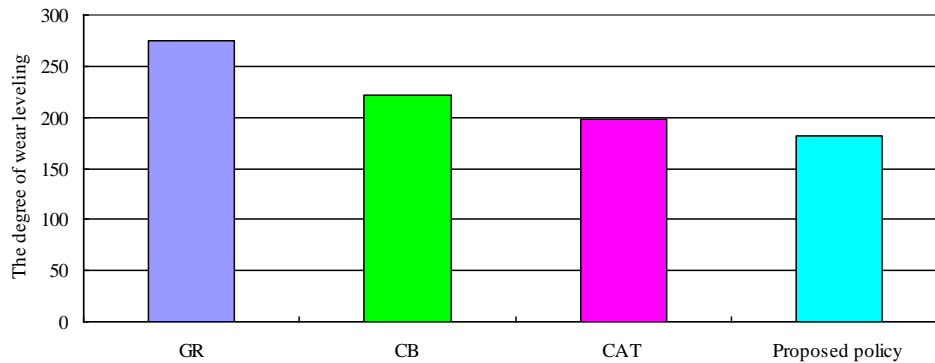
Figure 6 shows the experimental results of various garbage collection policies in terms of the number of erase operations. The GR policy obtains more free space than the CB and CAT policies, the GR policy incurs less erase operations for reclaiming garbage than the CB and CAT policies. Our proposed garbage collection policy introduces an improved exponentially weighted moving average method to predict the I/O workload in the near future and controls the number of erase operations. Moreover, it separates hot valid data from cold valid data in victim blocks and distributes them to different free blocks, victim blocks will be reduced and the number of erase operations will be decreased. Therefore, the proposed garbage collection policy incurs the least erase operations.



**Figure 7. Energy Consumption for Various Garbage Collection Policies**

Figure 7 illustrates the experimental results of various garbage collection policies in terms of the energy consumption. The energy consumption of a garbage collection policy mainly rely on the number of copy operations and the number of erase operations.

Because the proposed garbage collection policy incurs the least number of copy operations and the number of erase operations, the proposed garbage collection policy also consumes the least energy.



**Figure 8. The Degree of Wear Leveling for Various Garbage Collection Policies**

Figure 8 shows the experimental results of various garbage collection policies in terms of the degree of wear leveling. The GR policy does not consider the elapsed time since the last data invalidation on each block and the erase count of each block when selecting victim blocks, so it shows the highest degree of wear leveling. The CB policy introduces the elapsed time since the last data invalidation on each block when selecting victim blocks and the CAT policy also considers the erase count of each block when selecting victim blocks, so they show the lower degree of wear leveling. The proposed garbage collection policy first selects the young blocks with low erase count as victim blocks. Moreover, it distributes hot valid data in victim blocks into young free blocks with low erase count and cold valid data into old free blocks with high erase count. Therefore, the proposed garbage collection policy has the lowest degree of wear leveling.

## 7. Conclusions

In this paper, an intelligent garbage collection policy based I/O workload prediction is proposed for NAND flash-based storage devices. This policy first adopts an improved exponentially weighted moving average method to predict the I/O workload in the near future in order to control the number of garbage collection operations. Then, this policy introduces an efficient window-based victim block selection algorithm to first select the young blocks with low erase count as victim blocks. Finally, this policy adopts a valid data migration algorithm that separates hot valid data from cold valid data within victim blocks and then distributes them to different free blocks. A series of experiments have been conducted with a real trace and experimental results assess the effectiveness of the proposed garbage collection policy.

## Acknowledgments

The work of this paper is supported by the Natural Science Foundation Project of CQ CSTC under Grant No. cstc2013jcyjA40059, National Natural Science Foundation of China under Grant No. 61502102, No. 61370078, and No. 61402109, Fujian Province Education Scientific Research Projects for Young and Middle-aged Teachers under Grant No. JA15122, National Undergraduate Training Programs for Innovation and Entrepreneurship under Grant No. 201510394021, and Fujian Normal University

Undergraduate Training Programs for Innovation and Entrepreneurship under Grant No. cxxl-2015163.

## References

- [1] M. L. Lesser and J. W. Haanstra, "Random-access memory accounting machine. I. system organization of the IBM 305", IBM Journal Research and Development, vol. 44, no. 1, (2000), pp. 6-15.
- [2] J. Elerath, "Hard-disk drives: The good, the bad, and the ugly", Communications of the ACM, vol. 52, no. 6, (2009), pp. 38-45.
- [3] G. MacGillivray, "Hard-disk drives vs. flash", Electronic Engineering Times, no. 1408, (2006), pp. 42-44.
- [4] Q. Wei, J. Chen and C. Chen, "Accelerating file system metadata access with byte-addressable nonvolatile memory", ACM Transactions on Storage, vol. 11, no. 3, (2015).
- [5] Z. Qin, Y. Wang, D. Liu and Z. Shao, "Real-time flash translation layer for NAND flash memory storage systems", Proceedings of 18th IEEE Real Time and Embedded Technology and Applications Symposium, (2012), pp. 35-44.
- [6] S. Jung and Y. H. Song, "Data loss recovery for power failure in flash memory storage systems", Journal of Systems Architecture, vol. 61, no. 1, (2015), pp. 12-27.
- [7] J.-W. Hsieh, C.-H. Wu and G.-M. Chiu, "MFTL: A design and implementation for MLC flash memory storage systems", ACM Transactions on Storage, vol. 8, no. 2, (2012).
- [8] C.-H. Wu, C.-K. Jan and T.-W. Kuo, "A low-memory address translation mechanism for flash-memory storage systems", Journal of Information Science and Engineering, vol. 27, no. 5, (2011), pp. 1713-1727.
- [9] L.-Z. Han, Y. Ryu, T.-S. Chung, M. Lee and S. Hong, "An intelligent garbage collection algorithm for flash memory storages", Lecture Notes in Computer Science, vol. 3980, (2006), pp. 1019-1027.
- [10] G. Lawton, "Improved flash memory grows in popularity", Computer, vol. 39, no. 1, (2006), pp. 16-18.
- [11] Y.-H. Chang, J.-W. Hsieh, J.-H. Lin and T.-W. Kuo, "A strategy to emulate NOR flash with NAND flash", ACM Transactions on Storage, vol. 6, no. 2, (2010).
- [12] D. M. Han, "Fast erase algorithm using flash translation layer in NAND-type flash memory", IEEE Transactions on Consumer Electronics, vol. 57, no. 4, (2011), pp. 1749-1755.
- [13] M. Fabiano and G. Furano, "NAND flash storage technology for mission-critical space applications", IEEE Aerospace and Electronic Systems Magazine, vol. 28, no. 9, (2013), pp. 30-36.
- [14] S. Jung and Y. H. Song, "Garbage collection for low performance variation in NAND flash storage systems", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 34, no. 1, (2015), pp. 16-28.
- [15] C. Zhang, Y. Wang, T. Wang, R. Chen, D. Liu and Z. Shao, "Deterministic crash recovery for NAND flash based storage systems", Proceedings of the 51st Design Automation Conference, (2014), pp. 148:1-148:6.
- [16] M. A. A. Sanvido, F. R. Chu, A. Kulkarni and R. Selinger, "NAND flash memory and its role in storage architectures", Proceedings of the IEEE, vol. 96, no. 11, (2008), pp. 1864-1874.
- [17] K.-S. We, C.-G. Lee, K. Yi, K.-J. Lin and Y. S. Lee, "HRT-RLRU: A new paging scheme for executing hard real-time programs on NAND flash memory", IEEE Transactions on Computers, vol. 63, no. 4, (2014), pp. 927-940.
- [18] M. Wu and W. Zwaenepoel, "eNVy: A non-volatile, main memory storage system", Proceedings of the Sixth International Conference on Architecture Support for Programming Languages and Operating Systems, (1994), pp. 86-97.
- [19] A. Kawaguchi, S. Nishioka and H. Motoda, "A flash memory based file system", Proceedings of the 1995 USENIX Technical Conference, (1995), pp. 155-164.
- [20] M.-L. Chiang and R.-C. Chang, "Cleaning policies in mobile computers using flash memory", Journal of Systems and Software, vol. 48, no. 3, (1999), pp. 213-231.
- [21] Z. Xiao, W. Song and Q. Chen, "Dynamic resource allocation using virtual machines for cloud computing environment", IEEE Transactions on Parallel and Distributed Systems, vol. 24, no. 6, (2013), pp. 1107-1117.

## Authors



**Xiaobo Ji**, received his B.S. degree in Command Automation and the M.S. degree in Computer Science from Logistical Engineering University of China, Chongqing, China, in 2002 and 2005, respectively. He also received his Ph.D. degree in Computer Science from Chongqing University, Chongqing, China, in 2011. Currently, he is with the Department of Information, Research Institute of Field Surgery, Daping Hospital, Third Military Medical University,

Chongqing, China. His current research interests include distributed system and flash memory.



**Fan Zeng**, received the B.S. degree in Computer Science from Chongqing Communication Institute, Chongqing, China, in 1997 and the M.S. degree in Science of Educational Management from Southwest University, Chongqing, China, in 1999. Currently, she is with the Department of Information, Research Institute of Field Surgery, Daping Hospital, Third Military Medical University, Chongqing, China. Her research interests include distributed system and flash memory.



**Mingwei Lin**, received his B.S. and Ph.D. degrees from Chongqing University, China, in July 2009 and December 2014. Currently, he is a lecturer in the Faculty of Software, Fujian Normal University, China. His research interests include anomaly detection, NAND flash memory, Linux operating system, and cloud computing. He got the CSC-IBM Chinese Excellent Student Scholarship in 2012.



**Fei Gao**, is a junior student in the Faculty of Software, Fujian Normal University, Fuzhou, China. He majors in Software Engineering. He has applied for a National Undergraduate Training Program for Innovation and Entrepreneurship successfully. His current research interests include cloud computing, android application development, and flash memory.