IEICE TRANS. INF. & SYST., VOL.E100–D, NO.5 MAY 2017

LETTER

RRWL: Round Robin-Based Wear Leveling Using Block Erase Table for Flash Memory*

Seon Hwan KIM[†], **Ju Hee CHOI**[††], ***Nonmembers***, ***and*** **Jong Wook KWAK**[†a)], ***Member***

SUMMARY In this letter, we propose a round robin-based wear leveling (RRWL) for flash memory systems. RRWL uses a block erase table (BET), which is composed of a bit array and saves the erasure histories of blocks. BET can use one-to-one mode to increase the performance of wear leveling or one-to-many mode to reduce memory consumption. However, one-to-many mode decreases the accuracy of cold block information, which results in the lifetime degradation of flash memory. To solve this problem, RRWL consistently uses one-to-one mode based on round robin method to increase the accuracy of cold block identification, with reduced memory size of BET, like in one-to-many mode. Experiments show that RRWL increases the lifetime of flash memory by up to 47% and 14%, compared with BET and HaWL, respectively.

key words: *flash memory, wear leveling, hidden cold block problem, bit array table, block erase table*

1. Introduction

Recently, flash memory has been in a great demand from server computer sectors to embedded systems for storage devices. However, it has a weak point which is the limited number of erase operations. To manage this disadvantage, wear leveling techniques have been proposed. Wear leveling increases the lifetime of flash memory by equalizing the number of erases for each block and uses various information tables [1].

The information tables save the data for wear leveling such as elapsed time of access, erase count of blocks, bit error rate, and so on. However, these information tables require additional memory space and they have to be minimized in limited memory space requirements. Therefore, to reduce the memory consumption of information tables, block erase table (BET) techniques have been studied [2], [3].

BET is composed of a bit array and saves the histories of erase operations in each block [2]. Subsequently, during a given time period, if there are no erased blocks, BET techniques set the unerased blocks to cold blocks and migrates the cold blocks for wear leveling. The k of BET techniques indicates how many blocks are mapped into a bit of BET. For one-to-one mode, k is 0, while k is more than 1, 2^k blocks are mapped to one bit, called one-to-many mode.

However, a conventional BET technique has a hidden cold block problem. The hidden cold block problem is a phenomenon that information of cold blocks disappears with the erase operation of hot blocks in one-to-many mode [3]. The performance of wear leveling is degraded by the hidden cold block problem in one-to-many mode of BET. To solve the hidden cold block problem, HaWL uses a bit-set threshold, which is based on a maximum deviation compared with the average invalid page count of blocks related to each bit in BET [3]. However, identifying cold blocks based on the number of invalid pages has limitations in particular cases. For example, some cold blocks may have more invalid pages than hot blocks. Therefore, wear leveling techniques using BET need to improve the accuracy of cold block identification further.

In this letter, to solve this problem, we propose a novel wear leveling technique, called round robin-based wear leveling (RRWL). RRWL consistently uses one-to-one mode based on the round robin method to increase the accuracy of cold block identification, with reduced memory size of BET, like in one-to-many mode.

2. Round Robin-Based Wear Leveling

To increase the accuracy of cold block identification in RRWL, each bit of BET is related to one block as one-to-one mode, even though the value of k is more than 1. Subsequently, RRWL fulfills wear leveling using the history of block erasures like BET techniques.

RRWL uses two values for one-to-one mode, when the value of k is more than 1. First, round robin index (RR_{index}) is used for selecting target blocks of BET. In each round for wear leveling, RRWL changes target blocks, which participate in wear leveling. RR_{index} is calculated by Eq. (1) and has 0 as the initial value.

$$RR_{index} = (RR_{index-1} + 1) \bmod 2^k,$$
$$0 \le RR_{index} \le (2^k - 1) \tag{1}$$

Second, the addresses of all blocks are converted into internal addresses of each group (G_{addr}) by Eq. (2) and G_{addr} identifies the target blocks compared with RR_{index}. b_{index} means the physical address of each block, which is erased by garbage collection.

$$G_{addr} = b_{index} \bmod 2^k,$$

Manuscript received November 28, 2016.
Manuscript publicized January 30, 2017.
[†]The authors are with the Department of Computer Engineering, Yeungnam University, Gyeongsan, Korea.
[††]The author is with the Department of Computer Science and Engineering, Seoul National University, Seoul, Korea.
*This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2014R1A1A2057146).
a) E-mail: kwak@yu.ac.kr
DOI: 10.1587/transinf.2016EDL8228

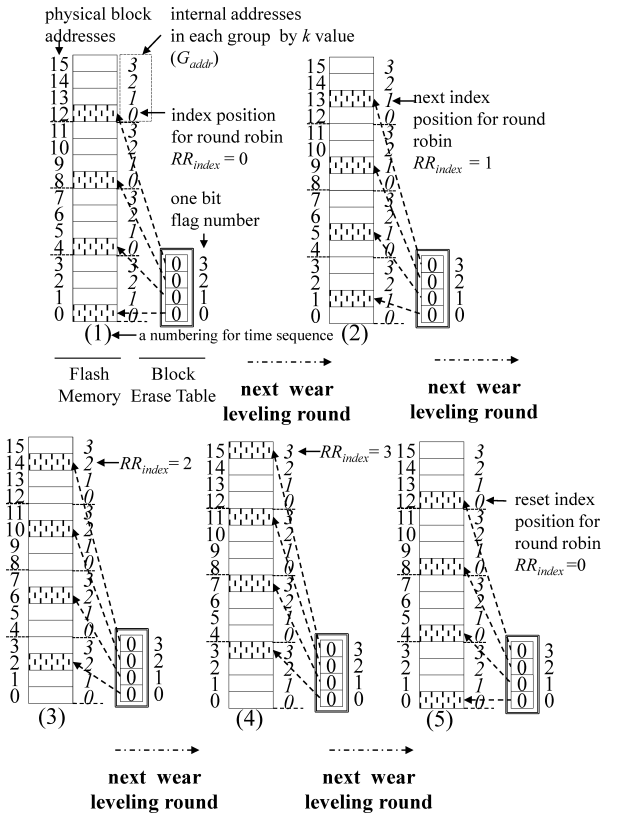


Fig. 1 The mapping method of RRWL. (k=2)

$$0 \le G_{addr} \le (2^k - 1) \tag{2}$$

When a block is erased by garbage collection, G_{addr} of the block is compared with RR_{index}. If two values are equal, a bit of BET, which is related the block, is set to 1; otherwise, the bit does not changed. Therefore, each bit of BET is related to one block by RR_{index} and G_{addr} despite the value $k \ge 1$ in RRWL. In consequence, RRWL solves the hidden cold block problem and improves the accuracy of cold block identification, because the main cause of the hidden cold block problem is induced by the condition that each bit of BET is related to two or more blocks.

Figure 1 is an example using RR_{index} in RRWL, when the value of k is 2, and thus each group is composed of four blocks in flash memory by the value of 2^k. The number of BET bits is 4 because the number of blocks is 16 in flash memory. In Fig. 1, we show the status of BET and RR_{index} based on a numbering for time sequence. (1) of Fig. 1 is initial status of BET and RR_{index}. RR_{index} is 0 in first round of RRWL. If the round of RRWL, which migrates cold blocks, is completed, next RR_{index} is recalculated by Eq. (1), and (2), (3), and (4) of Fig. 1 increases RR_{index}, continuously. In (5) of Fig. 1, RR_{index} become 0 again by Eq. (1) as round robin method.

Algorithm 1 shows a procedure of RRWL. In Algorithm 1, the triggering condition of wear leveling is same as the BET technique [2]. If wear leveling is triggered in line 2, a victim block is selected by k, BET, and RR_{index} in line

Algorithm 1: Procedure of RRWL

input : e_{cnt}(total erase count), f_{cnt}(total 1s bit count of BET), b_{index}, f_{index}, k, T, RR_{index}, and BET
output: *null*

1 **if** $f_{cnt} = 0$ **then return** ;
2 **while** $e_{cnt}/f_{cnt} = T$ **do** // Check the trigger of RRWL
3 **if** $f_{cnt} \ge size(BET)$ **then** // for next round of RRWL
4 $e_{cnt} \leftarrow 0$, $f_{cnt} \leftarrow 0$;
5 $f_{index} \leftarrow$ RANDOM(0, $size(BET)$-1);
6 reset all flags in the BET;
7 $RR_{index} \leftarrow (RR_{index} + 1) \bmod 2^k$;
8 **return** ;
9 **end if**
10 **while** $BET[f_{index}] = 1$ **do** // Select a victim
11 $f_{index} \leftarrow (f_{index} + 1) \bmod size(BET)$;
12 **end while**
13 $victimblock \leftarrow (f_{index} \times 2^k) + RR_{index}$;
14 **call** EraseBlock($victimblock$);
15 $f_{index} \leftarrow (f_{index} + 1) \bmod size(BET)$;
16 **end while**

Algorithm 2: BET update for RRWL

input : e_{cnt}, f_{cnt}, b_{index}, k, G_{addr}, and BET
output: e_{cnt}, f_{cnt}, and BET are updated based on the erased block address b_{index} related from k and RR_{index} in the BET mapping

1 $e_{cnt} \leftarrow e_{cnt}$+1; // Increase the total erase count.
2 $G_{addr} \leftarrow (b_{index} + 1) \bmod 2^k$;
3 **if** $RR_{index} = G_{addr}$ **then** // Update the BET if needed
4 **if** $BET[\lfloor b_{index}/2^k \rfloor] = 0$ **then**
5 $BET[\lfloor b_{index}/2^k \rfloor] \leftarrow 1$;
 // Increase the total 1's count of BET.
6 $f_{cnt} \leftarrow f_{cnt}$+1;
7 **end if**
8 **end if**

10~14. The victim block means that it has to be migrated to a free block for wear leveling. At this point, RRWL has a difference compared with the conventional BET technique. RRWL migrates one victim block by Eq. (3), whereas the conventional BET technique migrates one or more blocks according to k, in each round of wear leveling. f_{index} means the index of a bit, which has zero bit in BET.

$$victimblock = (f_{index} \times 2^k) + RR_{index} \tag{3}$$

Therefore, RRWL has an advantage that the overhead of wear leveling is low, compared with the conventional BET technique. If all bits of BET are 1, RR_{index} is changed by Eq. (1) and whole BET entries are reset in line 3~9.

Algorithm 2 shows an update of BET in RRWL. The update of BET is similar to the conventional BET technique. However, the update condition is different in line 2~3. The condition is based on the comparison of RR_{index} and G_{addr}. Each bit of BET is changed to 1, when G_{addr} is equal to RR_{index}.

RRWL uses one-to-one mode so it can efficiently solve the hidden cold block problem which is induced in one-to-many mode. Figure 2 is a process to solve the hidden cold

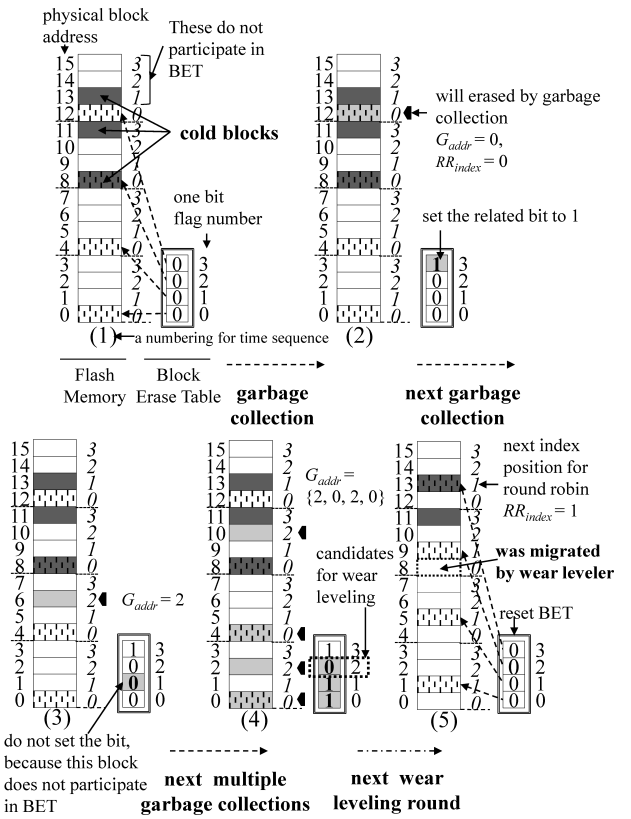


Fig. 2 A process of RRWL to indentify and migrate hidden cold blocks. (k=2)

block problem in RRWL. We assume that the number of cold blocks is 3 in Fig. 2. If the block 12 is erased by garbage collection, the corresponding BET entry is set to 1 by Algorithm 2 in (2) of Fig. 2. In case of (3), the BET bit is not changed because RR_{index} and G_{addr} are not equal, whereas in the conventional BET technique, the bit is set to 1 because it is according to erase operations of block 4, 5, 6 and 7.

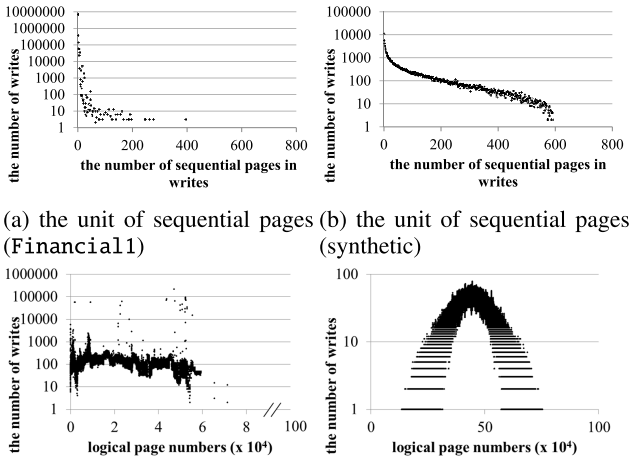
Subsequently, block 10, 4, 2, and 0 are erased by garbage collection and all bits of BET are set to 1 except for bit 2 of BET in (4) of Fig. 2. The block 8, which is related to bit 2 of BET is migrated by wear leveler, which is a hidden cold block. In (5) of Fig. 2, if next round of wear leveling is started, RR_{index} becomes 1 by Eq. (1). Remaining hidden cold blocks will be migrated by next rounds of wear leveling as in the prior progression.

3. Performance Evaluation

We implemented a simulator for our experiments. In our experiments, we compare the performance of RRWL with BET and HaWL [2], [3]. Table 1 shows details of experiment parameters used. To evaluate the performance of our proposal under various conditions, we used a financial trace and a synthetically generated trace. The financial trace is traced by an online transaction process application (OLTP), named `Financial1` [4]. Note that this trace is widely used

Table 1 Details of experiments

Items	Values and Descriptions
block count	4096
page count per block	256
page size	8KB
each file size in synthetic	593 pages (4744KB)
total file count in synthetic	1500
garbage collection trigger	the proportion of free block is under 2%.
wear leveling trigger	$e_{cnt}/f_{cnt} < 10$ (T = 10)
initial free space	15%
bad block	erase operation count > 3000 times.



(a) the unit of sequential pages (`Financial1`)
(b) the unit of sequential pages (synthetic)
(c) the distribution of logical page numbers (`Financial1`)
(d) the distribution of logical page numbers (synthetic)

Fig. 3 The unit of sequential pages and the distribution of logical page numbers in writes of experiments. (10^7 writes)

for the performance measurements of storage systems.

In addition, we also used a synthetic trace. The update probability of the synthetic trace exploits a normal distribution for the general cases, because Financial1 has high locality of reference. The update probability with normal distribution (p) and the calibrated update probability (cp_x) for maximum probability use Eq. (4) and Eq. (5), respectively.

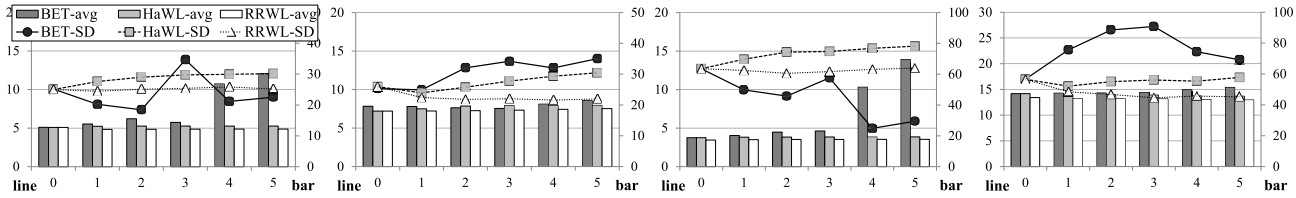
$$p = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \tag{4}$$

$$cp_x = p_x \frac{\max(p) - \min(p)}{100}, p_x \in p \tag{5}$$

Additionally, our experiments use two types of flash translation layer (FTL), which are the FTL using page-level mapping (pFTL) and the NAND flash translation layer using block-level mapping (NFTL), in mapping algorithms [5].

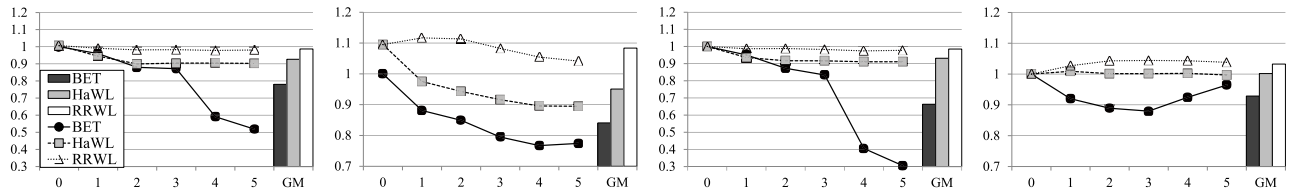
Figure 3 indicates the unit of sequential pages and the distribution of logical page numbers, when the number of page writes is 10^7 in `Financial1` and the synthetic trace. The writes of `Financial1` have high locality of reference and use smaller page units compared with the synthetic trace.

Figure 4 shows the standard deviation of erased blocks and the average number of erased blocks in the experiments. The standard deviation of RRWL is lower than that



(a) pFTL (Financial1) (b) pFTL (synthetic) (c) NFTL (Financial1) (d) NFTL (synthetic)

Fig. 4 The standard deviation of the number of erased blocks (SD) and the average number of block erases (avg) in the experiments. (x axis shows values of k and y axis is the number of erase blocks in 10^7 writes.)



(a) pFTL (Financial1) (b) pFTL (synthetic) (c) NFTL (Financial1) (d) NFTL (synthetic)

Fig. 5 The lifetime of flash memory and geometric means (GM) in the experiments. (x axis shows values of k and y axis is normalized lifetime by BET (k=0).)

of HaWL, and RRWL has steadily balanced performance, that is a uniform distribution of standard deviation in Fig. 4. There are the cases that the standard deviation of BET is lower than that of RRWL and HaWL in particular cases, such as Fig. 4 (a) and Fig. 4 (c). However, the average number of erased blocks in BET is higher than that of RRWL and HaWL. Therefore, in given consideration to the performance of wear leveling, RRWL is better than BET and HaWL in the final outcome.

Figure 5 represents the lifetime of flash memory and geometric means in the experiments. Lifetime prolongation is the most important performance indicator in wear leveling. The lifetime of flash memory is based on the first time to occur any bad blocks [2]. In all cases of Fig. 5, the lifetime of RRWL is increased, compared to BET and HaWL. According to k, RRWL increases the lifetime of flash memory by up to 3.18 times compared with BET, when the value of k is 5 in Fig. 5 (c). In geometric means, the lifetime of RRWL is increased by up to 47% and by up to 14% BET and HaWL, respectively.

4. Conclusion

To increase the lifetime of flash memory, we proposed a round robin-based wear leveling (RRWL) technique. RRWL uses a block erase table (BET), which is composed of a bit array. BET has two modes: one-to-one mode and one-to-many mode. The one-to-many mode has an advantage of low memory consumption, while the performance of wear leveling is decreased compared with one-to-one mode. To solve this problem, RRWL uses one-to-one mode with a round robin method to increase the lifetime of flash memory, consuming the reduced size of BET like one-to-many mode. Our experimental results showed that RRWL prolonged the lifetime of flash memory by up to 47% and 14%, compared with BET and HaWL, respectively.

References

[1] M.-C. Yang, Y.-M. Chang, C.-W. Tsao, P.-C. Huang, Y.-H. Chang, and T.-W. Kuo, "Garbage collection and wear leveling for flash memory: past and future," 2014 International Conference on Smart Computing (SMARTCOMP), pp.66–73, Nov. 2014.

[2] Y.-H. Chang, J.-W. Hsieh, and T.-W. Kuo, "Improving flash wear-leveling by proactively moving static data," IEEE Trans. Comput., vol.59, no.1, pp.53–65, Jan. 2010.

[3] S.H. Kim, J.H. Choi, and J.W. Kwak, "HaWL: Hidden Cold Block-Aware Wear Leveling using Bit-Set Threshold for NAND Flash Memory," IEICE Trans. Inf. & Syst., vol.E99-D, no.4, pp.1242–1245, April 2016.

[4] Storage Performance Council, http://traces.cs.umass.edu/index.php/Storage/Storage, 2002.

[5] D. Ma, J. Feng, and G. Li, "A survey of address translation technologies for flash memories," ACM Computing Surveys (CSUR), vol.46, no.3, pp.1–39, Jan. 2014.