An Efficient NAND Flash Garbage Collection Algorithm Based on Area and Block Operation

Wang Meng, Bu Kai, Xie Qiyou, Sun Zhaolin, Xu Xin, Xu Hui

National University of Defence Technology, Changsha, Hunan, 410072, China
wangmeng827@163.com

***Abstract*—Garbage collection as a key technology for solid-state storage (SSD), it's has critical influence for the SSD's performance. This article addressed the low efficiency, high cost problem of the presented Garbage Collection (GC) Algorithm, Proposed a new Efficient NAND Flash GC Algorithm Based on Area and Block (GCbAB) Operation. The algorithm is not only efficient, low cost, but took into account the Wear Leveling, greatly improving the performance of SSD, extending the life of SSD. Through the actual verification, this algorithm is effective, and compared to existing algorithms have a great improved.**

Keywords-SSD; NAND Flash; Garbage Collection; Wear Leveling

I. INTRODUCTION

Flash memory has advantages of non-volatile, small size, light weight, anti-vibration, high performance, low power consumption, wide operating temperature range. In embedded systems and SSDs, Flash has been widely applied. At the same time there are some different features compared with hard disk drive (HDD): (1) Flash does not support overwritten, it mean that, Before a writing operation in a area of Flash, you must first erase this area; (2) Flash's smallest written unit is page, the smallest erase unit is block; (3)The erase cycles of Flash is limited. The number of erase limited is 3K to 100K times according to Flash types and manufacturing technics. Based on the above characteristics, we need to collect the invalid data of Flash to free up storage space, which is GC technology. The process of GC needs to consider the efficiency, cost, wear-leveling and other factors

II. ADDRESS MAPPING TECHNOLOGY

Address mapping technology is the logical address to physical address mapping method. Generally the operating system write data in accordance with HDD's sector size (512Bits) , but the smallest read and write unit of Flash is page (typical 4KB), the minimum erase unit is block, a block usually contains 128 or 256 pages.

To solve this problem, increased Flash translation layer (FTL) between the operating system and storage medium. When the file system have data to write, FTL convert the data to physical address and write to free physical page of the storage medium, and update the mapping table, while the original data was marked as invalid data. Therefore, address mapping technology is the basis of the SSD technology. The performance of SSD is affected directly by the conversion rate of FTL.

There are three address mapping technologies: page-based address mapping, block-based address mapping, address mapping mixed. Page-based address mapping table, page is the smallest unit. It has high flexibility, good performance, but the cost and power consumption is high. Block-based address mapping table, block is the smallest unit. It has simple structure, low cost, but the utilization of storage medium is low. Mixed address mapping automatically select block or page as the smallest unit, according to the data updated frequency.

III. HIGH EFFICIENCY AND LOW COST ALGORITHM : GCBAB

Because Flash memory using remote update mechanism, with the system running, the free blocks is reducing and the invalid data is increasing. We need to erase the block contains invalid data to get a new free block, which is GC's function.

A. Start-time of GC

Start GC is triggered by the dynamic threshold, the threshold value is mainly according to the numbers of remaining free blocks. For the new SSD, all blocks of Flash are free, the threshold value $\mu = 0$, without the need for GC. With the increase of writing, the free blocks gradually reduced, the threshold value increases. When the Flash don't have free block, the threshold value $\mu = 0$. For the idle state of the SSD, the triggered level of GC increased correspondingly, the threshold value is:

$$\lambda = \mu + \phi$$

$$(0 \le \lambda \le 1,\ 0 \le \phi < 0.5;\ when\ \ \mu + \phi > 1,\ \ \lambda = 1)$$

ϕ is the level parameters of SSD's state, set according to the specific use environment. For consumer-grade SSD, write data is limited; it has long time and high probability in idle state, so the value of ϕ is less. For enterprise-class SSD, the pressure of writing data is high; it has low probability and short time in idle state, so the value of ϕ is larger. Dynamic threshold taking into account of SSD's using environment, improving the effectiveness of GC, while weakening GC's effect for the storage performance.

978-0-7695-4608-7/12 $26.00 © 2012 IEEE
DOI 10.1109/ISdea.2012.438

B. Select the area for GC

To improve the efficiency of GC and reduce the cost, according to the frequency of data update, the data is divided into three categories: hot data, warm data, and cold data; while the area of data storage is divided into hot area, warm area, and cold area correspondingly.

GC program once started, first step select the data area for GC. Different areas have different priority for GC. The area's priority value ρ for GC is:

$$\rho = \frac{N}{D+F} \times E$$

N is the number of invalid blocks in the area, D indicates the number of valid data blocks in the area, F was the number of free blocks. E is the priority parameters of the block. E_1, E_2 and E_3 was the priority parameter of hot area, warm area and cold area respectively，and

$$E_1 > E_2 > E_3$$

C. Determine the block for GC

A data area contains many blocks, just select the appropriate block to get high GC efficiency relatively. For GC shoud choose a block include more invalid pages. Also take into account the remaining erased count of the block. The block's GC priority value δ is:

$$\delta = \gamma(\frac{P}{L+W} - \frac{W}{U})$$

$$(\gamma = \frac{C}{T}, \quad U = L + W + P)$$

Where P is the number of invalid pages, L is the number of valid data pages, W is the number of free pages, γ is the ratio block of remaining erase times, C is the number of remaining erase times of the block, T is theoretical value of erase times.

Identified the Block for GC, then according to the situation associated of blocks, finish a GC using exchange merge, or part merge, or all merge.

D. Free blocks redistribution

Getting the block from GC reallocation, hot area，warm area and cold area have the right amount of free blocks for use. During redistribution of free blocks, we need to consider the following factors: the ratio of free block in each area, the forecast demand for free blocks in each area, the priority of reallocate for each area, the value between the remaining erase number of this block and the average of all blocks. Each area's priority of reallocation is:

$$\psi = (1 - \frac{F}{N+D+F})\omega \bullet \varphi$$

Where ω is the demand forecast for free blocks of each area, the forecast is based on the consumes of free blocks in the recent period. φ is the priority parameters of distributing free blocks; $\varphi_1, \varphi_2, \varphi_3$ are the priority parameters of hot area, warm area and cold area, there is:

$$\varphi_1 > \varphi_2 > \varphi_3$$

When the needs of each area's priority for the free blocks determined, we must to determine the specific allocation area according to the free block's wear leveling θ.

$$\theta = \frac{A - C}{T}$$

Where A is the average number of erase times of all blocks.

When the priority ψ in all area under the same conditions, a block with high degree of wear leveling have priority assigned to the cold area; When the priority ψ in each area is not same, considering the value and weight of ψ, θ.

IV. VERIFY ALGORITHM PERFORMANCE AND COMPARISON

This part is through contrasting with the existing GC algorithms to complete the verification work. The existing GC algorithms are FIFO, Cost Age Time (CAT), Garbage Collection based Temporal Locality (GCbTL).

A. Experimental setting

Verification algorithm uses FPGA-based hardware platform for the IDE interface SSD, FPGA model is Xilinx XC3S5000-FG900, Flash model is Samsung K9HCG08U1M-PCB0. The Device, Block, Page size is as follows:

1 Page = (4K+128) Bytes,

1 Block = (4K+128)B × 128 Pages,

1 Device = (4K+128)B × 128 Pages × 8,192 Blocks

=33,792 Mbits

Verification platform was shown in Figure 1.

Figure 1. The platform for verification

Before the test, fill the SSD with random data. The data size and composition are shown in Table 1.

Table 1. The write data size and composition

NO.	Random write data size	Composition
1	512 bytes (0.5k)	4%
2	1,024 bytes (1k)	1%
3	1,536 bytes (1.5k)	1%
4	2,048 bytes (2k)	1%
5	2,560 bytes (2.5k)	1%
6	3,072 bytes (3k)	1%
7	3,584 bytes (3.5k)	1%
8	4,096 bytes (4k)	67%
9	8,192 bytes (8k)	10%
10	16,384 bytes (16k)	7%
11	32,768 bytes (32k)	3%
12	65,536 bytes (64k)	3%

B. Experiment results

The SSD using in the experiment has been full with random data, and continues random writing at full speed with the data as shown in Table 1. The results shown in Figure 2, Figure 3. In Figure 2, the abscissa is the number of the system actually writes data operations to the SSD; the vertical coordinate is the number of GC operations. In Figure 3, the abscissa is the block's ID number of the SSD; the vertical axis is the number of block erased.

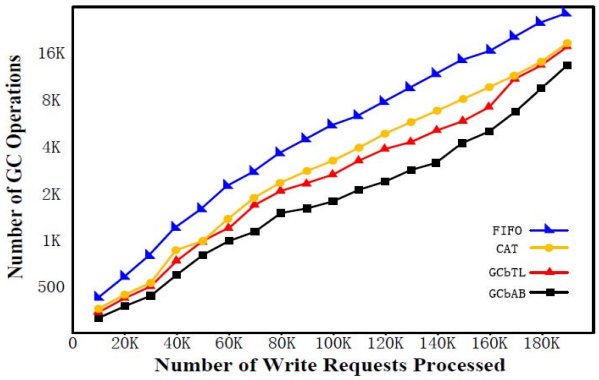

Figure 2. Number of GC Operations at Different Write Requests

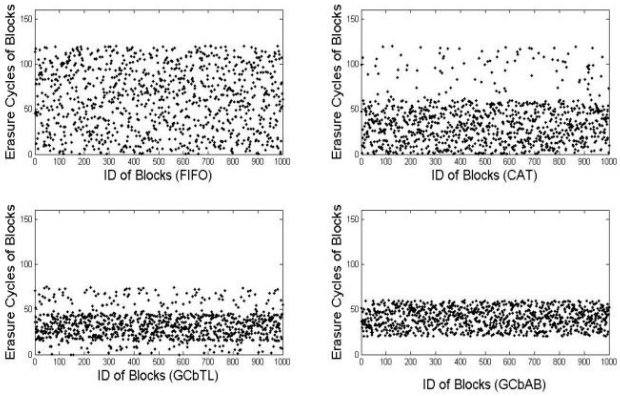

Figure 3. Distribution of Block's Erasure Cycles

Finally, test the recovery performance of the SSD. The SSD's 4KB data average writing speed is 45MB/s in the initial state, the state of its Flash shown in Figure 4. With the data shown in Table 1, the SSD take an hour of data write stress test, the write speed reduced to about 8MB/s. Then the SSD get into stable state, the state of its flash shown in Figure 5.

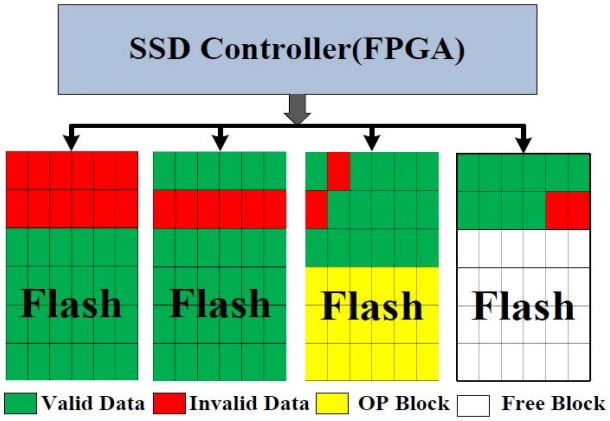

Figure 4. The Data Distributing in the Flash of Normal State

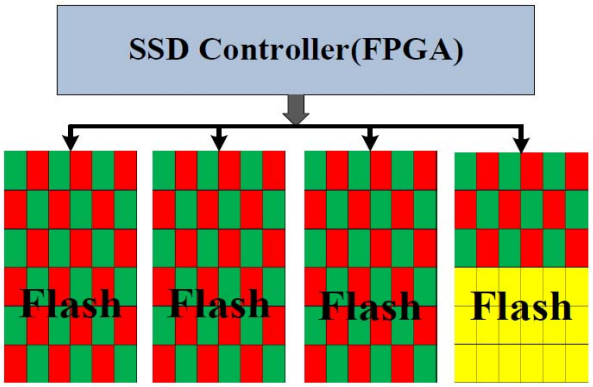

Figure 5. The Data Distributing in the Flash of Steady State

Sequential writing 4KB data to the steady-state SSD in order to observe the GC's efficiency through writing process; recording the recovery time of writing performance, the results shown in Figure 6.

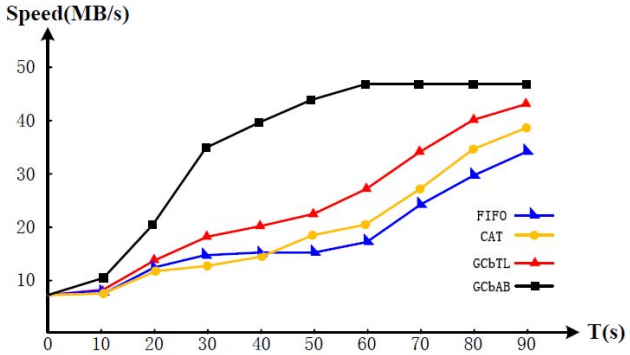

Figure 6. SSD's Performance Recovery Time

V. CONCLUSIONS

This article studies one of the key technologies of SSD: GC, put forward the GCbAB algorithm, which simplifies the GC process, improve the efficiency of GC; taking into account the factors wear leveling, reducing the cost. The algorithm is proved to be efficient and feasible through practical verification. Furthermore, we need to further deepen the study of GC and verify GCbAB's performance in the SATA SSD in the future.

REFERENCES

[1] Zheng Wenjing, Flash Storage Technology, Journal of Computer Research and Development ,2010,47(4).(in Chinese)

[2] Kee-Hoon Jang, Efficient Garbage Collection Policy and Block Management Method for NAND Flash Memory. 2010 2nd International Conference on Mechanical and Electronics Engineering (ICMEE 2010)

[3] Youngjae Kim, FlashSim: A Simulator for NAND Flash-based Solid-State Drives. 2009 First International Conference on Advances in System Simulation.

[4] SHI Zheng. A Garbage Collection Algorithm for Flash File System Based on Differential Evolution. ACTA ELECTRONICA SINICA, 2011.2, VOL.39, NO.2. (in Chinese)

[5] YUE Li hua, Efficient Space Allocation and Reclamation Mechanism for Flash Memory. Journal of Chinese Computer Systems. 2011.5, VOL.31, NO.5. (in Chinese)

[6] HU Zhi－gang, Garbage Block Collection Algorithm for NAND Flash-memory Taking in to Consideration 0peration Temporal Locality. Journal of Chinese Computer Systems. 2008.10, VOL.29, NO.10. (in Chinese)

[7] JEDEC STANDAR, JESD219, Solid-State Drive (SSD) Endurance Workloads, SEPTEMBER 2010.