International Journal of Multimedia and Ubiquitous Engineering
Vol.10, No.8 (2015), pp.329-338
http://dx.doi.org/10.14257/ijmue.2015.10.8.32

Flash-aware Virtual Memory System for Consumer Electronics

Xiaobo Ji[1] and Fan Zeng[2]

[1] *Department of Information, Research Institute of Field Surgery, Daping Hospital, Third Military Medical University, Chongqing, China*
bati0716@126.com
Department of Information, Research Institute of Field Surgery, Daping Hospital, Third Military Medical University, Chongqing, China
zengfan215@163.com

Abstract

In this paper, an efficient flash-aware virtual memory system called FVMS is proposed for consumer electronics equipped with flash memory as secondary storage media. It introduces: 1) a redesigned read-ahead algorithm to delay the execution of page replacement algorithm, 2) a flash-aware page replacement algorithm considering access frequency, recent access time, and asymmetric I/O operation costs to reduce the number of flash page write operations and prevent serious degradation of page hit ratio, and 3) an efficient garbage collection policy taking into consideration the number of valid pages within each block and the number of erase operations on each block in order to reduce the cleaning cost and improve the degree of wear leveling. Experimental results show that the proposed FVMS outperforms existing virtual memory systems for flash memory in terms of energy consumption and the degree of wear leveling.

Keywords: *Flash memory, Virtual memory system, Consumer electronics*

1. Introduction

Flash memory becomes one of the most popular nonvolatile storages because of its superiority in fast data access speed, low power consumption, shock resistance, high reliability, small size, and light weight. Moreover, the price is decreasing and the capacity of single flash memory is increasing, many consumer electronics have been widely equipped with flash memory as their secondary storage media [1-2].

Current design practice for application execution on consumer electronics is to transfer operating system and application codes from flash memory to main memory, which is referred to as memory shadowing architecture. The memory shadowing architecture shows good performance at runtime since all the application codes are copied to main memory in advance, but it results in slow boot process. Moreover, a large amount of main memory is needed to hold both operating system and application codes even though not all of them are executed simultaneously.

As emerging embedded applications grow in code and data requirement, memory shadowing architecture is not available for use to consumer electronics due to its limited main memory. In order to utilize the limited main memory, consumer electronics currently exploit virtual memory systems, which allow application programs to be executed by loading code and data on demand from the secondary storage to the main memory. Thus, it requires a less main memory capacity and a shorter loading time than the memory shadowing architecture.

Traditional virtual memory systems have been optimized for decades under the assumption that the secondary storage consists of magnetic disk. Magnetic disk can be overwritten and the cost of page read operation is equal to that of page write operation.

ISSN: 1975-0080 IJMUE

Due to the limited main memory of consumer electronics, traditional virtual memory system, which is directly implemented on consumer electronics, will use up the limited main memory quickly since traditional virtual memory system performs read-ahead algorithm and reads eight pages including the requested one into main memory. In this case, page replacement algorithm will be triggered frequently to evict pages and obtain free page frames. Since the cost of flash page write operation to flash memory is higher than that of flash page read operation and intensive write operations could result in using up the free space of flash memory-based secondary storage due to out-of-place update scheme adopted to solve the erase-before-write constraint of flash memory, the read-ahead algorithm and page replacement algorithm designed for magnetic disk are not available for use to flash memory-based secondary storage. Due to the out-of-place update scheme, an efficient garbage collection policy should be included in virtual memory system to reclaim garbage and obtain free space.

In this paper, an efficient flash-aware virtual memory system called FVMS is proposed for consumer electronics. The main contributions can be summarized as follows:

(1) In order to delay using up the limited main memory and incurring the execution of the page replacement algorithm, a flash-aware read-ahead algorithm scheme is introduced.

(2) Since the cost of page write operation of flash memory is higher than that of page read operation and intensive write operations could use up the free space of flash memory-based secondary storage quickly due to out-of-place update scheme adopted to solve the erase-before-write constraint of flash memory, a flash-aware page replacement algorithm is designed to reduce the number of flash page write operations and prevent serious degradation of page hit ratio.

(3) Since the cleaning cost and the degree of wear leveling are two primary concerns of garbage collection policy, an efficient garbage collection policy considering the number of valid pages within each block is proposed to reduce the cleaning cost and the number of erase operations on each block to improve the degree of wear leveling.

(4) Trace-driven simulations are conducted to evaluate the performance effectiveness of the proposed FVMS in terms of energy consumption and the degree of wear leveling.

The rest of this paper is organized as follows. Section 2 briefly introduces the overview of flash memory and also shows the motivation of this work. Section 3 briefly reviews existing works on virtual memory system for flash memory, page replacement algorithm, and garbage collection policy. Section 4 presents the design and implementation of the proposed FVMS. Section 5 shows the performance evaluation. Finally, in the Section 6, the conclusions are drawn.

2. Background and Motivation

This section briefly introduces the overview of flash memory. By showing the unique characteristics of NAND flash memory, the potential problems of traditional virtual memory systems are addressed as the motivation of this work.

2.1. Flash Memory

There are two major types of flash memory. One is NOR flash memory and the other is NAND flash memory [3]. NOR flash memory provides fast data access speed and execute-in-place (XIP) functionality, so it is usually used in place of ROM (Read Only Memory) to store and execute application code. In contrast, NAND flash memory is usually employed as high capacity storage media since it provides high density with low cost. NAND flash memory provides three basic operations: read, write, and erase [4]. The read operation fetches data from a target page, while the write operation writes data to a page. The erase operation resets all the values of a target block to 1. Namely, the

granularity of erase operation is a block, while the granularity of read and write operation is a page.

However, NAND flash memory exhibits some unique characteristics, which might have a significant influence on the efficiency and performance of traditional virtual memory system directly implemented over flash memory [5]. Firstly, flash memory adopts out-of-place update scheme to solve the erase-before-write constraint of flash memory. The erase-before-write constraint works as follows. Once a page is written, it should be erased in advance before the subsequent write operation is performed on the same page. Secondly, each block has a limited cycle count on the erase operation, typically 10,000~100,000 times. A block will be worn out when this number is exceeded. A worn-out block could suffer from frequent write errors. Finally, the cost of flash page write operation is much higher than that of flash page read operation in terms of latency and energy consumption.

2.2. Motivation

Traditional design practice for application execution in consumer electronics adopts memory shadowing architecture to transfer all operating system and application codes from the secondary storage to main memory during boot process. Since the main memory of consumer electronics is limited and emerging embedded applications grow in code and data requirement, consumer electronics exploit virtual memory system to replace memory shadowing architecture in order to improve boot time in case of without increasing main memory.

Traditional virtual memory systems have been optimized for decades under the assumption that the secondary storage consists of magnetic disk. Because of the limited main memory, traditional virtual memory system will use up the limited main memory quickly since traditional virtual memory system performs read-ahead algorithm and reads eight pages including the requested one into the main memory. In this case, page replacement algorithm will be triggered frequently to evict pages and obtain free page frames. However, flash memory shows several very different characteristics compared to magnetic disk. The out-of-place update scheme is adopted to solve the erase-before-write constraint of flash memory and intensive write operations will use up the free space of flash memory quickly. In this case, garbage collection policy with high energy consumption will be incurred frequently to reclaim garbage. Moreover, the cost of flash page write operation to flash memory is much higher than that of flash page read operation. Therefore, the read-ahead algorithm and the page replacement algorithm should be redesigned in light of the unique characteristics of flash memory. Due to the out-of-place update scheme, an efficient garbage collection policy should be included in virtual memory system to reclaim garbage and obtain free space. In a word, an efficient virtual memory system for flash memory should focus on optimizing three important parts, which are flash-aware read-ahead algorithm, flash-aware page replacement algorithm, and garbage collection policy.

3. Related Work

3.1. Existing Works on Virtual Memory System

In order to utilize the limited main memory, Park *et al.* [6] proposed an energy-aware virtual memory system for NAND flash memory based embedded storages with the assistance of operating system and memory management unit (MMU) hardware. They characterize the interactive applications with large memory footprints in terms of memory access pattern and CPU (Central Processing Unit) utilization. And they also analyze the performance and energy consumption bottleneck in traditional virtual memory system. Finally, they proposed an improved page replacement algorithm called CFLRU, which is

optimized for NAND flash memory. Although this virtual memory system could efficiently reduce the energy consumption of consumer electronics, it does not optimize the read-ahead algorithm in light of the unique characteristics of flash memory and also not design an efficient garbage collection policy for virtual memory system.

3.2. Existing Works on Page Replacement Algorithm

Existing page replacement algorithms for magnetic disk are customized to improve the page hit ratio under the assumption that the costs of page read and write operation of magnetic disk are equal. Because the costs of flash page read and write operation of flash memory are asymmetric and the out-of-place update scheme is used to solve the erase-before-write constraint of flash memory, existing page replacement algorithms designed for magnetic disk are not available for use to flash memory and a number of page replacement algorithms have been studied for consumer electronics by modifying the LRU (Least Recently Used) algorithm.

Park *et al*. proposed the first flash-aware page replacement algorithm called CFLRU [7], which considers asymmetric I/O costs of read and write operation of flash memory. CFLRU links all the pages by the LRU order and divides the page list into two regions, which are the working region and clean-first region. The working region contains the recently referenced pages and most of page hits are generated in this region. The pages in clean-first region are victim candidates and the number of pages within the clean-first region is determined by the size of the window. In order to reduce the replacement cost, CFLRU first evicts clean pages in the clean-first region by the LRU order under the assumption that the cost of flash page write operation is much higher than that of flash page read operation. If there are no clean pages in the clean-first region, it evicts the rest pages by the LRU order. CFLRU can significantly reduce the number of write operations to flash memory by delaying the flush of dirty pages in the buffer.

Based on CFLRU, Yoo *et al.* [8] proposed three improved algorithms called CFLRU/C, CFLRU/E, and DL-CFLRU/E. CFLRU/C, which considers the access frequency of dirty pages, preferentially selects the least recently referenced clean page as the victim within the pre-specified window. If there is no clean page within the window, CFLRU/C evicts the dirty page with the lowest access frequency.

CFLRU/E considers the erase count of the block, which the victim page candidate belongs to. It preferentially selects the clean page as victim like CFLRU and CFLRU/C. If there is no clean page within the window, CFLRU/E evicts the dirty page belonging to the block with the lowest erase counts. DL-CFLRU/E maintains two LRU lists, which are the clean page list and the dirty page list, respectively. DL-CFLRU/E checks the clean page list first for selecting a victim page. If there is a clean page in the list, the least recently referenced page is evicted. Otherwise, it scans the dirty page list and evicts the page belonging to the block with the lowest erase counts within the window. If there is no dirty page within the window, it scans the rest of the dirty page list and evicts the dirty page with the least recently referenced time. CFLRU/C, CFLRU/E, and DL-CFLRU/E reduce the number of write operations to flash memory and improve the degree of wear leveling.

Jung *et al.* proposed another flash-aware page replacement algorithm called LRU-WSR, which enhances the normal LRU algorithm with add-on page replacement strategy, namely Write Sequence Reordering (WSR) [9], [10]. LRU-WSR delays the flush of dirty pages, which are not regarded as cold, to reduce the number of write operations to flash memory and prevent the serious degradation of page hit ratio. In order to identify cold dirty page, cold-detection algorithm is introduced and each page in the LRU list has an additional flag called cold flag. Dirty page with setting cold flag is regarded as cold and cold flag of dirty page is cleared when the page is referenced again. LRU-WSR scans the LRU list and checks the least recently referenced page. If the checked page is clean, LRU-WSR evicts it regardless of the status of its cold flag. If dirty, LRU-WSR checks the cold flag of this dirty page. If it is not set, this dirty page is moved to the MRU position of the

LRU list with setting the cold flag and another candidate page is selected from the LRU position of the LRU list. If the candidate page is a dirty page and its cold flag is set, the dirty page is evicted and flushed into the flash memory.

3.3. Existing Works on Garbage Collection Policy

The cleaning cost and the degree of wear leveling are two primary concerns of garbage collection policy. To achieve these two goals, a number of garbage collection polices have been studied.

The greedy algorithm (GR) [11-12] evicts the block with the largest amount of garbage for erasure, hoping to obtain free space as much as possible with the least copy cost. The greedy algorithm tends to select a block in a FIFO order irrespective of data access patterns. It is known that the greedy policy works well for uniform access, but does not perform well for high localities of access.

The cost-benefit algorithm (CB) [13] evicts the block, which maximizes (1):

$$\frac{age \cdot (1-u)}{1+u} \tag{1}$$

Where u is the utilization of a block (the fraction of space occupied by valid data) and *age* is the time since the most recent modification. The cost is derived from migration and erasure, which are expressed as the denominator (u+1). The benefit is given as the space-time product form. The term (1-u) reflects how much free space it acquires. Because of the term *age*, cold blocks can be cleaned at a higher priority than hot blocks.

Performance of garbage collection policy depends on the combination of victim block selection policy and data migration policy. CB policy generally performs better than GR policy. But it does not perform well for high localities of access without combining efficient data migration policy.

In order to solve this problem, Han *et al.*. proposed the Cost-Age-Time with Age-sort algorithm (CATA). It evicts several blocks, which maximize (2):

$$\frac{1-u}{1+u} \cdot age \cdot \frac{1}{erase\_count} \tag{2}$$

Where u is the utilization of a block (the fraction of space occupied by valid data), *age* is the time since the most recent modification, and *erase_count* is the number of times a block has been erased, respectively.

The CATA [14] selects several victim blocks for better separation of cold and non-cold data. When it collects valid data from many victim blocks, cold and non-cold data are separated and distributed to different free blocks. The CATA uses the predicted workload to determine the number of victim blocks. If the CATA predicts that the number I/O request arrivals during the next garbage collection is high, it selects one or none victim block. Otherwise, it selects two or three victim blocks according to the predicted number of I/O request arrivals.

The Cost-Benefit with Age-sort algorithm (CBA) [15] is a combination of CB and CATA. When selecting victim blocks, it uses the victim block selection method like that of CB policy. It selects several victim blocks and reorganizes the clean data on victim blocks like CATA.

4. Proposed Flash-Aware Virtual Memory System

An efficient flash-aware virtual memory system called FVMS is proposed for consumer electronics. As described in introduction, the proposed FVMS optimizes the read-ahead algorithm and the page replacement algorithm, as well as proposes an efficient garbage collection policy to reduce the cleaning cost and improve the degree of wear leveling. They are described in detail as follows.

4.1. Flash-Aware Read-Ahead Algorithm

The read-ahead algorithm used in traditional virtual memory system reads eight pages including the requested one into the main memory at a time when page fault happens. Because the consumer electronics are equipped with the limited main memory due to the high cost of RAM (Random Access Memory), reading eight pages into the main memory at a time will result in using up the free page frames of main memory quickly. Therefore, the read-ahead algorithm is redesigned and only the requested page is loaded from flash memory to main memory. In this case, seven free page frames are saved and page replacement algorithm is delayed to run.

4.2. Flash-Aware Page Replacement Algorithm

Traditional page replacement algorithms have been designed for magnetic disk and focus on improving the page hit ratio. Flash memory adopts out-of-place update scheme to solve the erase-before-write constraint, so intensive write operations could use up the free space of flash memory quickly and garbage collection with high energy consumption will be incurred frequently to reclaim garbage. Moreover, the cost of flash page write operation is much higher than that of flash page read operation. Therefore, an efficient flash-aware page replacement algorithm called FAP is proposed to reduce the number of write operations to flash memory as well as prevent serious degradation of page hit ratio.

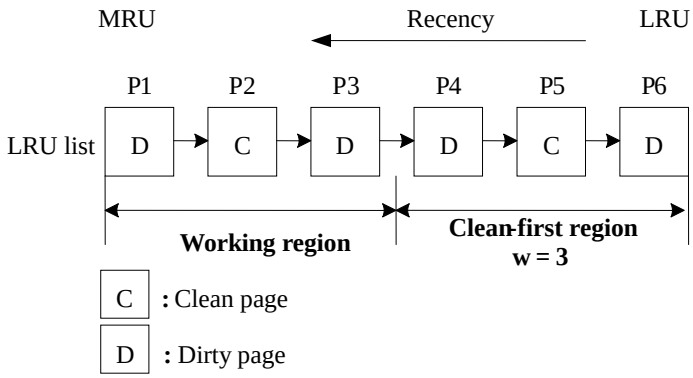


Figure 1. The Example of the FAP Algorithm

As depicted in Figure 1, FAP algorithm is modified from LRU policy and divides the LRU list into two regions, which are the working region and the clean-first region. FAP algorithm works as follows. FAP algorithm preferentially evicts the clean page, which maximizes (3) within the clean-first region.

$$\frac{1}{f} \times (T_c - T_i) \tag{3}$$

Where f is the access count of each page during recent 1 ns, T_c is current time and T_i is the time when page i is referenced last time.

If there is no clean page within the clean-first region, FAP algorithm evicts the dirty page, which maximizes (4) within the clean-first region.

$$\frac{1}{f} \times (T_c - T_i) \times \frac{\varepsilon_{\max} - \varepsilon_{\min}}{\varepsilon} \tag{4}$$

Where ε is the number of erase operations on block that page i belongs to. $\varepsilon_{\max}$ is the maximum erasure count of all blocks while $\varepsilon_{\min}$ is the minimum erasure count of all blocks.

4.3. Wear Leveling-Aware Garbage Collection Policy

In order to reduce the cleaning cost and improve the degree of wear leveling, an efficient wear leveling-aware garbage collection policy called WLGC is proposed. It evicts the block, which maximizes (5):

$$\frac{1}{x_i} \times \frac{1}{\max(y_1, y_2, \cdots, y_{i-1}, \cdots y_n) - \min(y_1, y_2, \cdots, y_{i-1}, \cdots y_n)} \quad (5)$$

Where x_i is the number of valid pages within block i and y_i is the number of erase operations on block i. When a victim block is selected, WLGC separates the cold valid pages from hot valid pages and distributes them to different free blocks like CATA policy.

5. Performance Evaluation

To verify the effectiveness of the proposed FVMS, trace-driven simulations are conducted to compare FVMS with EADP proposed by Park *et al.*. The synthesized traces are illustrated in Table 1. A read/write ratio "10%/90%" in Table 1 means that the read and write operations in the trace are of 10% and 90%, respectively. The locality "80%/20%" shows that 80% of total references are intensively performed on the 20% of data pages.

Table 1. The Characteristics of Synthesized Traces

Type	Total references	Read/write ratio	locality
T1	300,000	90%/10%	80%/20%
T2	300,000	50%/50%	80%/20%
T3	300,000	10%/90%	80%/20%
T4	300,000	90%/10%	50%/50%
T5	300,000	50%/50%	50%/50%
T6	300,000	10%/90%	50%/50%

To evaluate the replacement cost, the weighted count of read and write operations on flash memory is used. The write count is weighted with eight times higher than the read count. This is based on the access latency and the energy consumption considering the cost of potential erase operations.

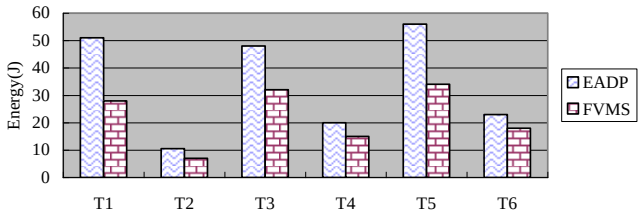


Figure 2. The Energy Consumption for EADP and FVMS System

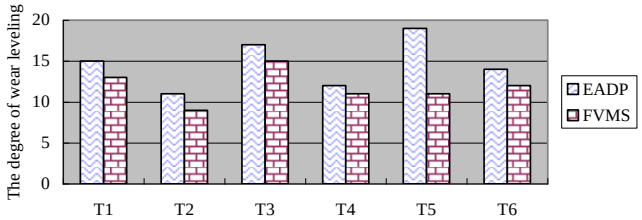


Figure 3. The Degree of Wear Leveling for EADP and FVMS System

Figure 2 and Figure 3 show the energy consumption and the degree of wear leveling for EADP and FVMS system. FVMS redesigns the read-ahead algorithm to only load the requested page from flash memory-based secondary storage media to main memory. In this case, the execution of page replacement algorithm is delayed. In order to reduce the number of write operations to flash memory, FVMS also introduces an efficient flash-aware page replacement algorithm taking the asymmetric costs of flash page read and write operation into consideration. And then the number of garbage collection operations with high energy consumption is reduced. Therefore, FVMS consumes less energy compared with EADP as shown in Figure 2. In order to wear out all the blocks as evenly as possible, FVMS also introduces an efficient wear leveling-aware garbage collection policy taking into account the erase count on each block when a victim block is selected. Figure 3 shows that FVMS outperforms EADP in terms of the degree of wear leveling.

6. Conclusion

In this paper, an efficient flash-aware virtual memory system called FVMS is proposed for consumer electronics, which are equipped flash memory as secondary storage media. In order to delay the execution of page replacement algorithm, FVMS redesigns the read-ahead algorithm and only load the requested page from flash memory-based secondary storage media to main memory. Because of the out-of-place update scheme adopted to solve the erase-before-write constraint of flash memory and the much higher cost of flash page write operation than that of flash page read operation, FVMS introduces an efficient flash-aware page replacement algorithm taking into account recent access time, access frequency, and the erase count on each block. In order to reduce the cleaning cost and improve the degree of wear leveling, an efficient wear leveling-aware garbage collection policy is also proposed to select a reasonable victim block for erasure. A series of experiments are conducted and experimental results showed that the proposed FVMS outperforms EADP in terms of energy consumption and the degree of wear leveling.

Acknowledgements

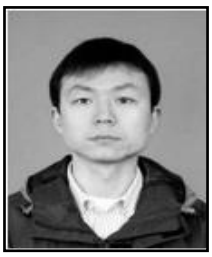
This work is supported by Natural Science Foundation Project of CQ CSTC under Grant No. cstc2013jcyjA40059.

References

[1] D. Woodhouse, “JFFS: The journaling flash file system,” Proc. of Ottawa Linux Symposium, **(2001)**.
[2] Mingwei L., Shuyu C. and Guiping W., “Greedy page replacement algorithm for flash-aware swap system,” IEEE Transactions on Consumer Electronics, vol. 58 no. 2, **(2012)**, pp. 435-440.
[3] Mingwei L. and Shuyu C., “Efficient and intelligent garbage collection policy for NAND flash-based consumer electronics,” IEEE Transactions on Consumer Electronics, vol. 59 no. 3, **(2013)**, pp. 538-543.
[4] Mingwei L. and Shuyu C., “Flash-aware Linux swap system for portable consumer electronics,” IEEE Transactions on Consumer Electronics, vol. 58 no. 2, **(2012)**, pp. 419-427.
[5] Mingwei L., Shuyu C., Guiping W. and Tianshu W., “HDC: An adaptive buffer replacement algorithm for NAND flash memory-based databases,” OPTIK, vol. 125 no. 3, **(2014)**, pp. 1167-1173.
[6] Chanik P., Jeong U. K., Seon Y. P. and Jin S. K., “Energy-aware demand paging on NAND flash-based embedded storages,” Proc. of the 2004 International Symposium on Low Power Electronics and Design, **(2004)**, pp. 338-343.
[7] S. Y. Park and D. Jung, “CFLRU: A replacement algorithm for flash memory,” Proc. of the 2006 International Conference on Compilers, Architecture and Synthesis for Embedded Systems, **(2006)**, pp. 234-241.
[8] Y. S. Yoo, H. Lee, Y. Ryu and H. Bahn, “Page replacement algorithms for NAND flash memory Storages,” Proc. of International Conference on Computational Science and its Applications, **(2007)**, pp. 201-212.
[9] H. Jung, H.Shim, S. Park, S. Kang and J. Cha, “LRU-WSR: Integration of LRU and writes sequence reordering for flash memory,” IEEE Transactions on Consumer Electronics, vol. 54 no. 3, **(2008)**, pp. 1215-1223.

[10] Z. Li, P. Jin, X. Su, K. Cui and L. Y, "CCF-LRU: A new buffer replacement algorithm for flash memory," IEEE Transactions on Consumer Electronics, vol. 55 no. 3, **(2009)**, pp. 1351-1359.
[11] M. Wu and W. Zwanepoel, "eNvy: A non-volatile main memory storage system," Proc. of the 6th International Conference on Architectural Support for Programming Language and Operating systems, **(1994)**.
[12] A. Kawaguchi, S. Nishioka and H. Motoda, "A flash memory based file system," Proc. of USENIX 1995 Technical Conference, **(1995)**, pp. 155-164.
[13] M. Rosenblum and J. K. Ousterhout, "The design and implementation of a log-structured file system," ACM Transactions on Computer Systems, vol. 10 no. 1, **(1992)**, pp. 26-52.
[14] L. Han, Y. Ryu, and K. Lim, "CATA: A garbage collection scheme for flash memory file systems," Lecture Notes in Computer Science, vol. 4159, **(2006)**, pp. 103-112.
[15] L. Han, Y. Ryu, T. 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.

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.