A Hybrid Flash Memory SSD Scheme for Enterprise Database Applications

Byung-Woo Nam, Gap-Joo Na and Sang-Won Lee
School of Information & Communications Engr.
Sungkyunkwan University
Suwon 440-746, Korea
{*moonbow81, factory, swlee*}*@skku.edu*

***Abstract*—Flash memory has many advantages such as high performance, low electronic power, non-volatile storage and physical stability, over hard-disks. For this reason, flash memory has been deployed as data storage for mobile devices, including PDAs, MP3players, laptop-computers and database systems. According to the cell type, flash memory can be divided into SLC(Single Level Chip) and MLC(Multi Level Chip). In general, SLC is known to have high performance and longer lifetime (i.e. more than 100K wear-leveling) while MLC is to offer larger capacity and with low price but have wear-leveling of not longer than 10K. In this paper, we show that it is possible to design a fast and cost-efficient storage by combining two types of flash memories in a hybrid fashion. Specifically, we propose a hybrid flash memory solid state disk(SSD) scheme using FAST FTL for enterprise applications, where SLC chip is used as the log space for FAST while MLC chips store the normal data blocks. SLC chips allow fast and durable performance for write while MLC chips provide the large capacity. And, this is mainly due to the FAST FTL algorithm's characteristics: it tends to direct the random writes to SLC chips and direct the other most random read to MLC chips. By taking the advantages of both chip types, we can find an economically desirable flash SSD design option. Experimental results show that our hybrid flash SSD scheme outperforms MLC-only flash scheme by far both in terms of performance and price.**

***Keywords*-NAND Flash Memory, OLTP, Database, FTL, Hybrid Flash Memory**

I. Introduction

NAND flash memory has been deployed as a non-volatile storage media for mobile devices including PDAs, MP3 players, mobile phones and digital cameras, mainly because of its characteristics such as high performance, low electronic power consumption and physical stability[1]. Moreover, as its density has increased and its price has decreased, flash-based storage devices are now considered to have tremendous potential as an alternative storage medium that can replace magnetic disk drives and achieve much higher performance.

Unlike hard-disk, however, flash memory does not allow any page to be updated in place just by overwriting it. Therefore, a write operation is available after a time-consuming erase operation is executed. Unfortunately, the basic unit of an erase operation is not a page but a block which is much larger than page. Hence, in order to overwrite any page in a block, other pages in the same block should be erased. Thus, it is necessary to preserve other valid pages in the block being erased, which incurs additional read/write operation. This characteristic of the flash memory is called 'erase-before-write' limitation and it is the main source of the performance bottlenecks in flash memory. Besides, every block allows the limited number of erase operations. If a block exceeds the threshold value of allowed erase operations, the data in the block becomes unreliable. This limitation is called 'wear-out' of the block. Therefore, we have to consider the wear-leveling as well as the 'erase-before-write' limitation[1].

In order to overcome these limitations and to use the flash memory as a storage device, flash memory storages are typically equipped with the so-called Flash Translation Layer (FTL) layer, a software module which emulates the function of the hard-disk by providing a block device interface. FTL handles the I/O operation with address mapping table, which can convert the logical addresses from host system into the physical addresses in flash memory. To minimize the erase operation, this scheme can delay the erase operations until the flash memory becomes full with this mapping table. Also, FTL tries to perform the erase operation over all blocks as uniformly as possible by maintaining the erase count for each block in the address mapping table. Therefore, flash memory based storages can overcome the major flash memory limitations.

According to the address mapping unit, FTL can be categorized into three types: block level, page level and log block(hybrid FTL) scheme[2]. Page level FTL can avoid overwriting in the same way as the log-structured file system [1]. Therefore, the write performance would be best, but it has two disadvantages. First, it requires the largest address mapping table. Second, it has to perform the garbage collection operation usually in a background manner, which incurs a heavy time-consuming operation. This, it is hard to expect the uniform write performance, which is a virtue of storage devices. In case of the block level scheme, the size of the address mapping table would be the smallest among the existing FTL schemes because the mapping granularity in the address mapping table is block. However, when updating a page in the block, all valid pages of the corresponding block must be copied to a free block, because the offset of the logical and physical block should always be kept equally.

978-0-7695-4012-2/10 $26.00 © 2010 IEEE
DOI 10.1109/APWeb.2010.70

In order to overcome the limitations of page level and block level FTLs, a few hybrid FTLs have been proposed. They adopt a block mapping table for the data blocks and a page mapping table for the log blocks, so that they can reduce 'erase-before-write' limitation with smaller memory space than the page level mapping table[1][2]. However, hybrid schemes also have a disadvantage. It requires the block merge operation which is induced by garbage collection, and a block merge operation results in many erase and write operations. Thus, it can deter the performance of flash memory. Thus, it is important to reduce the number of block merges. Our scheme uses one of the most popular hybrid FTLs, FAST, and we explain it in detail in the next section.

On the other hand, NAND flash memory can be divided into single-level-cell flash memory chip (SLC) and multi-level-cell flash memory chip (MLC). SLC chips represent one bit with one memory cell while MLC chips does two or more bits with one memory cell. Therefore, SLC chips are superior to MLC chips both in performance and durability. On the other side, MLC chips provide the larger capacity with lower price than SLC chip[3].

In this paper, we propose a hybrid flash memory SSD scheme that is composed of one small SLC chip and a number of large MLC chips for enterprise database applications. Our flash memory SSD scheme can provide the benefits of high write performance, large capacity, and low price by taking advantages of both SLC and MLC. In addition, it applies FAST to hybrid flash memory SSD, to support the Online Transaction Processing(OLTP) workloads efficiently. In particular, FAST FTL which is known to be good at random writes would be suitable for OLTP workloads, one of the most popular enterprise applications, because they consist of random write patterns[4][5]. Our paper contributes three achievements : 1) to propose a design of hybrid flash memory SSD and its operations, 2) to compare the performance of our hybrid scheme with the MLC-only scheme, and to evaluate its cost efficiency, 3) to show that the lifetime of flash SSD can be significantly prolonged by our hybrid scheme, which separates log area from data area and uses SLC chip as log area.

Recently, flash memory storages including SSDs have employed the multi-channel architecture to exploit the parallelism among flash memory chips[6][7]. Even if there are multiple channels, it is reasonable to concern that hybrid flash memory scheme using single channel flash memory storage system, as all flash memories are attached to a data-bus in a channel. Therefore, this paper presents our hybrid scheme assuming in single channel, but we believe that the benefits from our scheme can be generalized to multiple channels.

The remainder of this paper is organized as follows. Section 2 introduces FAST FTL on which our scheme is based, and reviews several existing hybrid flash memory schemes. Section 3 describes the structure of our hybrid flash memory scheme and its operations. Section 4 presents experimental results and its analysis. Finally, Section 5 concludes this paper and suggests some future works.

II. Background

In this section, we introduce overview of the flash translation layer and explain the two types of NAND flash chips. We also describe the related work.

A. Flash Translation Layer

Most flash storage devices are equipped with flash translation layer(FTL) in order to use the flash memory as a storage medium instead of the hard-disk[2]. The FTL provides a block device interface to the host system by emulating the operations of the hard-disks. In our research, we adopt the FAST FTL, which is a kind of the hybrid FTL and is known to be superior in random writes[8]. In FAST, the flash memory is logically divided into the data area and the log area. The data area is managed by block level address mapping and stores the original data. On the other hand, the log area is managed by page level address mapping and stores be stored the up-to-date data of the previously written data in flash memory. Meanwhile, log area is composed of a sequential log block and the multiple random log blocks. The key idea of FAST is to allow each page from original data block to be mapped any log block in a fully associative manner, and thus FAST can enhance the utilization of log block and improve the random write performance.

Since the main targets of our hybrid SSD are OLTP workloads and FAST has several advantages for OLTP workloads, we choose FAST for our hybrid flash SSD. First, as mentioned earlier, FAST shows good performance for random write patterns of OLTP workloads[5][4]. Second, the FAST scheme has the smaller overhead in scanning the address mapping table than page level FTLs, because of its much smaller address information to be maintained. Finally, considering characteristics of log block FTL, a lot of the I/O operations are mostly performed in the log area, rather than the data area. Because the random write performance and the longer lifetime of log area are important in our hybrid flash SSD, we decided to use SLC chip for the log area, which is faster and has longer lifetime.

B. Other Hybrid Flash Memory Schemes

In general, hybrid flash memory SSD schemes, which use both SCL and MLC flash memory, can improve the cost-efficiency and the total lifetime as well as better I/O performance. Recently, several hybrid flash memory schemes have been proposed[9][10].

In case of [9], the flash storage device(Hybrid-SSD) consists of one SLC chip and multiple MLC chips. The Hybrid SSD uses the page-level address mapping table and the write requests are directed to either SLC chip or MLC chips. The selection of the SLC or MLC chip is determined

Table I
CHARACTERISTICS OF GENERAL FLASH MEMORY AND FUSION FLASH MEMORY

	General flash		Fusion flash	
	SLC	MLC	SLC	MLC
Page Size	4 Kb	4 Kb	4 Kb	4 Kb
Block Size	256 Kb	512 Kb	256 Kb	512 Kb
Read Speed	25 μs	60 μs	45 μs	50 μs
Write Speed	200 μs	800 μs	240 μs	1000 μs
Erase Speed	1.5 ms	1.5 ms	0.5 ms	0.5 ms
Erase Cycle	100 k	10 k	50 k	10 k

by the locality of the data. The main idea of [9] is to store the hot data, which is updated frequently, in SLC chip, and to store the cold data, which is not updated frequently, in MLC chips, and thus the total I/O performance can be improved. In order to locate each data in the proper chip, however, data migration should be performed through periodical garbage collection, which is very time-consuming. Also, this scheme assume the page level mapping so that it should maintain large address mapping table. Consequently, whenever I/O operation, data migration, and wear-leveling are executed. it would be very time consuming to search the large address mapping table. So the expensive searching cost would be the main performance burden[9]. Besides, it is very difficulty to decide the hotness of a data item, especially for random I/O pattern.

Another work on hybrid flash memory schemes is [10]. In the [10], the authors use the new type of NAND flash memory, called 'fusion flash memory' (Flex-OneNAND[11], mobileLBA-NAND[12]). In fact, since the fusion flash memory is another type of NAND flash memory for mobile devices, it is not proper to be adopted in case of the enterprise systems. The main characteristic of the new flash chip is that every block can be changeable to a SLC block or a MLC block. In order to manage all blocks, they use the log block-based FTL and additional meta-data including the locality of each block, which are maintained in the address mapping table. When performing merge operation, FTL checks the locality of a data page, and decides whether it is evicted from SLC to MLC. In this scheme, the capacity of the hybrid flash memory is sensitive to the number of SLC blocks, since SLC and MLC blocks can be composed using the fixed blocks and the size of an SLC block is half of an MLC block. For example, if a MLC block is changed to a SLC block in order to increase the number of SLC block, the entire capacity will be decreased by the size of a SLC block. For these reasons, this scheme can use only a few SLC blocks to reserve large capacity. Moreover, as is shown in Table I[13][14][15], the performance of the fusion flash memory chip is worse than general SLC chip and MLC chip.

III. HYBRID FLASH MEMORY SSD SCHEME

A. The Structure of Hybrid Flash Memory SSD

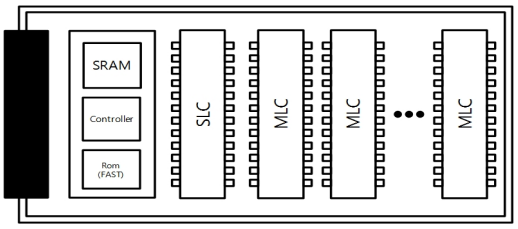

Figure 1. The Structure of hybrid flash memory SSD

Figure 1 shows the physical structure of the proposed hybrid flash memory SSD. The storage device is composed of controller, ROM, SRAM(or DRAM), an SLC NAND flash memory chip and several MLC NAND flash memory chips. Generally, accustomed flash memory storage devices consist of one kind of flash memory. But, in our purposed scheme, hybrid flash memory storage device is equipped with heterogeneous flash memory to take advantage of both SLC and MLC.

A flash controller is responsible for processing I/O operations requested by host applications with the FAST's algorithm stored in a ROM. In addition, it manages wear-leveling of all the blocks for improving device's lifetime and recovers data errors in flash memory chips. For instance, if a host application requests the flash controller to read data, the flash controller scans the address mapping table in a SRAM to translate logical address of this data into physical address and then the controller decides whether the controller read corresponding data from SLC or MLC with physical address. Finally, the controller reads the data in the selected flash memory chip.

We use the SLC for the log blocks because of the following two reasons. First, FAST is designed that it allocates most of the data being written by I/O operations to log area, because FAST considers log area to write buffer for avoiding erase operation. So data processing occurs in log area more frequently than the data area. For these reason, adopting SLC as log area improves whole I/O performance, due to the fast latency of SLC chip[3].

Second, under FAST's algorithm, A few of erase operations should flock to the log area, because the more write operations are executed in log area and the size of the log area is smaller than the data area. This phenomenon decays the wear-level of the log area more rapidly than that of the data area. But, SLC blocks are, in general, 10 times more durable than MLC blocks. Therefore, using SLC chip for the log area can keep a balance between wear-level of the SLC and the MLC blocks and improve the life-time of our hybrid flash SSD. Consequently, by using the SLC chip instead of the MLC chip as log area, we can improve the I/O performance and the durability.

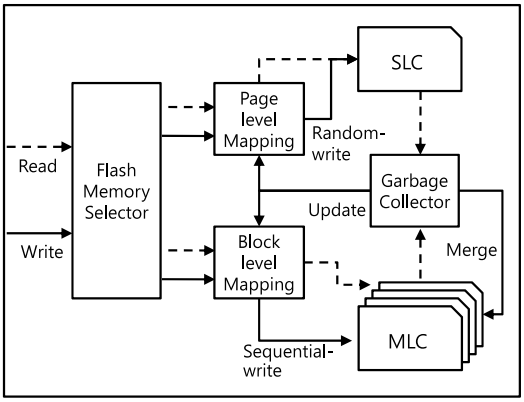

Figure 2. Operation diagram

Figure 2 describes the process of the I/O operation in our hybrid flash storage device. If a host application requests sequential write operation, the write operation is performed in MLC. However, if the write operation is not for massive data, that is, the overwrite operation is requested, the write operation should be executed in the SLC chip. If there is not clean blocks in SLC, the FAST FTL perform the garbage collection. In the next subsection, we explain the write operation and the garbage collection in detail.

B. The Operation in Data area

As is explained above, MLC chips are responsible for the data area in the FAST FTL. The operations of the data area can be divided into followings.

Case 1) Sequential write operation(Figure 3)
In case of the original FAST, the sequential write operations are executed in the log block and after bulk write operation, the FAST algorithm changes that log block into the data block. This operation can easily be done since every flash memory chips are the same type. However, in our design, since the block size is different depending on the type of NAND flash chips, and the log block and the data block are located in different flash chips, it is impossible to change the mode (log block to data block and vice-versa). Therefore, the sequential write operations are performed in data area (MLC chips). In addition, if the number of sequential writes is over one quarter of a MLC block, the data in old data blocks and new data to be written should be merged into a new data block.

Case 2) Block merge operation caused by SLC(Figure 4)
As is shown in Figure 4, this operation is performed during the garbage collection. When a log block is full, an erase operation should be performed in order to make a log block empty. However, there might be valid pages in the original data block so that the valid pages should be preserved in a victim block. This process is called the block merge operation. To complete the block merge operation, the valid pages in victim log block are merged with the corresponding

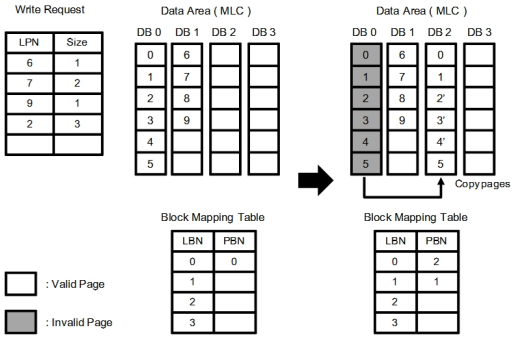

Figure 3. Sequential write operation

data blocks and then the up-to-date data pages are written to new data blocks. At this point, the old data blocks become invalid.

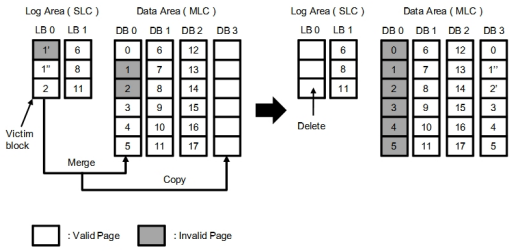

Figure 4. Merge operation

C. Operation of Log area

SLC chip is used for the log area, and the overwrite operation is performed in the log area. As explained in previous section, flash memory does not allow the overwrite operation. In most hybrid log block FTL scheme including FAST, the overwrite operation is performed by appending the data in the log area.

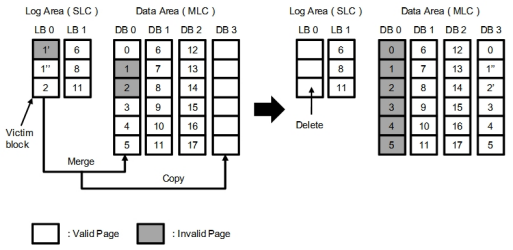

Figure 5. Write log data

Figure 5 shows the process of write operation in the log area. In order to execute the overwrite operations, the two address mapping table are updated that the corresponding data is invalid and the new data is written in the next empty pages of log blocks sequentially. Since the write operation in the log area is conducted without any erase

operation, the response time of write operation can improve. Meanwhile, since the capacity of log area is smaller than that of the data area, the log area becomes full in the course of time. Therefore, the garbage collection should be performed to make empty log block. To perform the garbage collection, the victim block has to be chosen in 'round-robin fashion'[8]. This method allows to improve the durability of the log area, because the erase operations can be distributed equally over all log blocks.

IV. Experiment

In order to prove the main advantages of our hybrid scheme, in this section, we evaluate the performance of our scheme and compare it with the MLC only scheme. For the experiment, we use a trace-driven simulator, which can calculate the number of read, write and erase operations of the flash storage schemes with the I/O trace files. Therefore, the total elapsed time can be calculated with the each I/O unit operation time from flash data-sheet[13][14]. Table II shows the characteristics of each type of NAND chips.

Table II
Characteristics of SLC and MLC

	SLC	MLC
Page Size	4 Kb	4 Kb
Block Size	256 Kb	512 Kb
Read Speed	25 μs	60 μs
Write Speed	200 μs	800 μs
Erase Speed	1.5 ms	1.5 ms
Erase Cycle	100 k	10 k
Price(1 Gbyte)	5.8 $	2.6 $

In our experiment, we set the total size of MLC chips as 8Gbyte(each MLC is 2Gbyte, and there are four MLCs) and then we change the size of log area from 5% to 25% of the data area size (8Gbyte), respectively. In order to compare the performance, we first use the MLC chip for the log area and then use the SLC chip for the log area. The I/O trace of workload for the experiment is extracted from the 'new-order', which is a part of the TPC-C benchmark. TPC-C benchmark is the standard benchmark for OLTP applications[4]. Figure 6 shows the performance results of the MLC-only scheme and our hybrid flash SSD scheme along with different size of the log area. As is shown in Figure 6, our hybrid storage scheme outperforms the MLC-based scheme(In our scheme, performance is improved up to 36%). The main reason of the result is that the most I/O operations are performed in SLC chip which has faster than MLC chip.

Now, let us think about the cost efficiency of our approach. As is shown in Figure 7, by exchanging MLC to SLC within 15% of full capacity, the storage cost increases by 16%, but the performance improve up to 27%. Moreover, as is shown in Figure 10, the lifetime of hybrid flash memory SSD can be prolonged because the wear-level of data area can increase

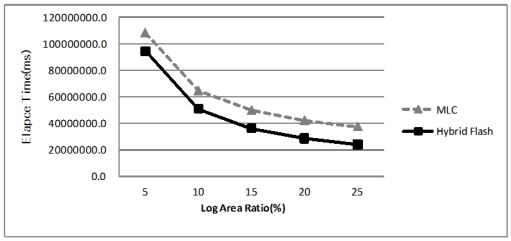

Figure 6. Execution Time

by our scheme Consequently, although the price of flash memory storage is raised by our hybrid scheme, the storage can achieve high performance and reliability.

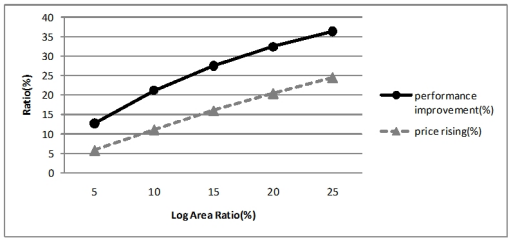

Figure 7. Price Raising and Performance Improvement

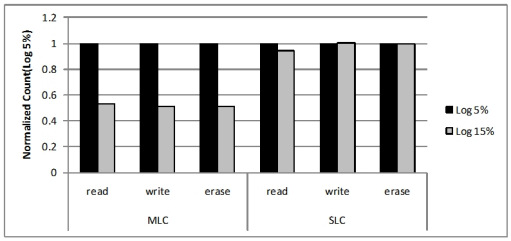

Figure 8. Variations in Flash Operations

Figure 8 describes the variation of the elapsed time in the IO operations in SLC and MLC as the size of the log area changes. As is shown in Figure 8, because the larger log area can perform the more write operation and reduce the block merge operation in data area, the number of I/O operation of the MLC decreases as the size of SLC increases.

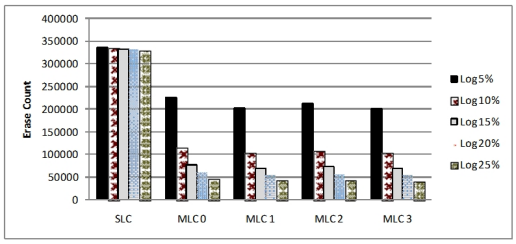

Figure 9. Erase Count in Hybrid Flash Memory SSD Scheme

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS
Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS
Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS
Sync your system to PACER to automate legal marketing.