Accelerating In-Page Logging with Non-Volatile Memory

Sang-Won Lee[†] Bongki Moon[‡] Chanik Park[§] Joo-Young Hwang[§] Kangnyeon Kim[†]

[†]School of Info. & Comm. Engr.
Sungkyunkwan University
Suwon 440-746, Korea
{*swlee,kangnuni*}*@skku.edu*

[‡]Dept. of Computer Science
University of Arizona
Tucson, AZ 85721, U.S.A.
bkmoon@cs.arizona.edu

[§]Samsung Electronics Co., Ltd.
San #16 Banwol-Ri
Hwasung-City 445-701, Korea
{*ci.park,jooyoung.hwang*}*@samsung.com*

Abstract

A great deal of research has been done on solid-state storage media such as flash memory and non-volatile memory in the past few years. While NAND-type flash memory is now being considered a top alternative to magnetic disk drives, non-volatile memory (also known as storage class memory) has begun to appear in the market recently. Although some advocates of non-volatile memory predict that flash memory will give way to non-volatile memory soon, we believe that they will co-exist, complementing each other, for a while until the hurdles in its manufacturing process are lifted and non-volatile memory becomes commercially competitive in both capacity and price. In this paper, we present an improved design of In-Page Logging (IPL) by augmenting it with phase change RAM (PCRAM) in its log area. IPL is a buffer and storage management strategy that has been proposed for flash memory database systems. Due to the byte-addressability of PCRAM and its faster speed for small reads and writes, the IPL scheme with PCRAM can improve the performance of flash memory database systems even further by storing frequent log records in PCRAM instead of flash memory. We report a few advantages of this new design that will make IPL more suitable for flash memory database systems.

1 Introduction

Flash memory has been the mainstream of solid state storage media for the past decade and is being considered a top alternative to magnetic disk drives [1, 5, 10, 11]. Recently, other types of non-volatile memory such as phase-change RAM (PCRAM) (also known as storage class memory) have also been actively studied, developed and commercialized by industry leading manufacturers [2, 3, 8, 14]. While NAND type flash memory is page addressable and used for block storage devices, PCRAM is byte addressable and can be used much like DRAM. Furthermore, PCRAM can be written (or programmed) in place without having to erase the previous state and exhibits much lower read latency than flash memory.

In this paper, we revisit the In-Page Logging (IPL) scheme that has been proposed as a new storage model for flash-based database systems [9] and elaborate how the IPL scheme can accelerate database systems further by utilizing PCRAM for storing log records. The byte-addressability of PCRAM allows for finer-grained writing of log records than flash memory, thereby reducing the time and space overhead for both page read and write

Bulletin of the IEEE Computer Society Technical Committee on Data Engineering

operations. The results of preliminary performance evaluation will be presented to illustrate the potential benefit of IPL adapted to hybrid storage systems equipped with both flash memory and PCRAM.

2 Flash Memory vs. Non-Volatile Memory

2.1 Flash Memory

Although there are two types of flash memory, namely NOR and NAND, we are interested in NAND-type flash memory, because NAND-type flash memory is the one that is used as a storage medium for a large quantity of data. NOR-type flash memory is byte addressable and is mainly used to store program codes. Hereinafter, we use the term flash memory to refer to NAND-type flash memory.

The unit of a read or write operation in flash memory is a page of typically 2K or 4K bytes. Since flash memory is a purely electronic device without any mechanical part, it provides much higher and more uniform random access speed than a magnetic disk drive, as shown in Table 1.

Media	Access time		
	Read	Write	Erase
Magnetic Disk†	12.7 ms (2KB)	13.7 ms (2KB)	N/A
NAND Flash‡	75 μs (2KB)	250 μs (2KB)	1.5 ms (128KB)
PCRAM¶	206 ns (32B)	7.1 μs (32B)	N/A
DRAM§	70 ns (32B)	70 ns (32B)	N/A

†Disk: Seagate Barracuda 7200.7 ST380011A;
‡NAND Flash: Samsung K9F8G08U0M 16Gbits SLC NAND [15];
¶PCRAM: Samsung 90nm 512Mb PRAM [8];
§DRAM: Samsung K4B4G0446A 4Gb DDR3 SDRAM [16]

Table 1: Access Speed: Magnetic disk vs. NAND Flash vs. PCRAM vs. DRAM

One of the unique characteristics of flash memory is that no data item can be updated by overwriting it in place. In order to update an existing data item stored in flash memory, a time-consuming erase operation must be performed before overwriting for an entire block of flash memory containing the data item, which is much larger (typically 128 or 256 KBytes) than a page. Dealing with the erase-before-write limitation of flash memory has been one of the major challenges in developing solid state storage devices based on flash memory.

Most contemporary storage devices based on flash memory come with a software layer called *Flash Translation Layer (FTL)* [4]. The key role of an FTL is to redirect a write request from the host to an empty (or clean) area in flash memory and manage the address mapping from a logical address to a physical address for an updated data item. The use of an FTL hides the erase-before-write limitation of flash memory and makes a flash memory storage device look like a conventional disk drive to the host.

For the past few years, advances in the solid state drive (SSD) technology have made flash memory storage devices as a viable alternative to disk drives for large scale enterprise storage systems. Most enterprise class flash memory SSDs are equipped with parallel channels, a large overprovisioned capacity, and a large on-drive DRAM cache. Combined with advances in the FTL technology, this new SSD architecture overcomes huddles in small random write operations and provide significant performance advantages over disk drives [5, 10].

2.2 Byte-Addressable Non-Volatile Memory

Non-charge-based non-volatile memory technologies have recently been under active development and commercialization by industry leading manufacturers. Unlike charge storage devices such as flash memory, these non-

volatile memory technologies provide memory states without electric charges [6]. Examples of these memory technologies are ferroelectric RAM (FeRAM), magnetic RAM (MRAM) and phase-change RAM (PCRAM). Among those, PCRAM is considered a leading candidate for the next generation byte-addressable non-volatile memory.

PCRAM devices use phase change material for a cell to remember a bit. The phase change material can exist in two different states, amorphous and crystalline, which can be used to represent zero and one. Switching between the two states can be done by application of heat at different temperature ranges for different durations [6]. PCRAM devices can be programmed to any state without having to erase the previous state. Due to repeated heat stress applied to the phase change material, PCRAM has a limited number of programming cycles. However, PCRAM is considered to have greater write endurance than flash memory by a few orders of magnitude [7]. Furthermore, in contrast to NAND type flash memory, PCRAM need not operate in page mode and allows random accesses to individual bytes like DRAM does.

It is reported in the literature that the read and write latencies of PCRAM are approximately an order of magnitude greater than those of DRAM [7]. As is shown in Table 1, however, contemporary PCRAM products do not deliver the promised performance as yet particularly for write operations. While PCRAM takes only about 8 ns to read a two-byte word, it takes as long as one μs to write a two-byte word. A similar disparity in read and write speeds is observed in other PCRAM products as well [13].

Given the unique characteristics of PCRAM, we expect that PCRAM and flash memory will co-exist complementing each other for a while until the manufacturing cost of PCRAM is reduced to a level comparable to that of flash memory. As will be discussed in Section 3, we are interested in the potential of PCRAM that would make the In-Page Logging (IPL) scheme more efficient. Specifically, PCRAM allows for smaller units of writes, and the write time is approximately proportional to the amount of data to transfer.[1] These traits make PCRAM an excellent candidate for a storage medium for log data managed by the IPL scheme.

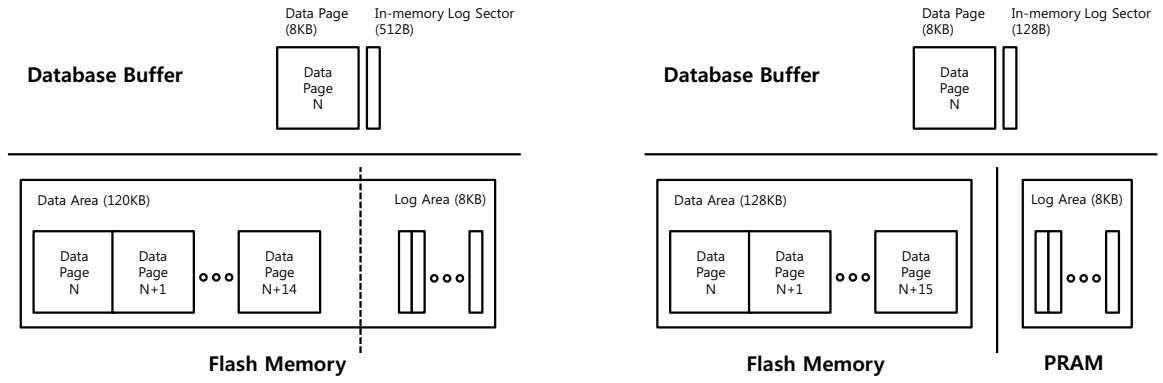
3 In-Page Logging with Non-Volatile Memory

In this section, we present a new implementation of IPL that utilizes non-volatile memory such as PCRAM to further improve its performance. For a hybrid storage device equipped with PCRAM as well as flash memory, IPL can boost its performance by taking advantage of higher access speed and byte-addressability of PCRAM for writing and reading small log records. We will review the IPL design and discuss the implications that the current technological trends of flash memory have on its performance. We will then present the new IPL implementation for hybrid storage devices.

3.1 IPL for Flash Only

It is quite common in the OLTP applications that most page frames become dirty in the buffer pool by small changes because updates are usually fairly small and distributed across a large address space. When a dirty page is about to be evicted from the buffer pool, a conventional buffer manager would write the entire dirty page back to a secondary storage device, even though the actual change amounts to only a small fraction of the page. The key idea of IPL is, as depicted in Figure 1(a), that only the changes made to a dirty page be written to a flash memory storage device in the form of log records without writing the dirty page itself [9]. Since the amount of changes is likely to be much smaller than the page itself, the IPL scheme can reduce the absolute amount of writes substantially, improve the overall write latency, and lengthen the lifespan of flash memory storage devices.

[1] For both read and write, the access time of PCRAM is not entirely proportional to the number of bytes to access, as there is an initial latency of about 78 ns for each operation [8]. Similarly, there is an initial latency of about 30 ns for DRAM [16]. The access times of PCRAM and DRAM shown in Table 1 include the latencies.



(a) Flash-only IPL

(b) IPL with PCRAM

Figure 1: IPL with Flash-only vs. IPL with Flash and PCRAM

The effectiveness of the IPL scheme, however, is being countered by the current trend in NAND flash memory technology towards a larger unit of I/O. For MLC-type NAND flash memory whose smallest unit of write is a page, it is impossible for IPL to reduce the absolute amount of writes because propagating even a few log records will require writing a full flash memory page. The situation becomes much easier for IPL when it comes to SLC-type flash memory. SLC-type NAND flash memory supports *partial programming* that allows a limited number of partial writes to be performed on the same page without erasing it. Thus, for example, a 2KB page can be written by four separate (512 byte) sector write operations instead of a single page write operation [12, 15].

While the effectiveness of IPL may not be compromised too much with SLC-type flash memory devices that are popular in enterprise database applications, there still remains a concern about the granularity of writes performed by the IPL scheme. As is discussed above, when a dirty page is about to be evicted, the amount of log records that need to be propagated for the page is usually very small. (It was in the range of 50 to 150 bytes in the TPC-C workloads we had observed.) This implies that write amplification could be further reduced if log records were written in a granularity smaller than even a sector. Obviously such a fine-grained writing is not feasible with any contemporary flash memory device, but it can be done very efficiently with byte-addressable non-volatile memory like PCRAM. Moreover, with PCRAM, the time taken to perform a write operation is almost linearly proportional to the amount of data to write. In contrast, this is not the case with NAND flash memory, because programming a sector takes the same amount of time as programming a full page, and they differ only in time taken to transfer a different number of bytes over the channel.

3.2 IPL for Flash and Non-Volatile Memory

Under the new design of IPL with PCRAM, as shown in Figure 1(b), regular data pages are stored in flash memory, while the corresponding log records are stored in PCRAM. Unlike the *flash-only* IPL, log records are not physically co-located with their corresponding data pages any longer, because data pages and log records reside in two separate storage components. Nonetheless, a flash memory block storing data pages must be associated with its log area in PCRAM efficiently. For simplicity, we assume that each data block in flash memory has a fixed length log area in PCRAM such that the address of a log area for a given data block can be determined by a direct mapping in constant time. Similar to the flash-only IPL, a merge operation needs to be carried out when the log area of a data block runs out of free space. Each log record in PCRAM log area is applied to its corresponding data page, and all the current data pages in the data block are written to a different clean data block.

In comparison, the new IPL design has the following advantages over the flash-only IPL. This is essentially attributed to the fact that the way PCRAM is used by the new IPL design for storing log records matches well with the characteristics of PCRAM.

- With byte-addressable PCRAM, log records can be flushed from the buffer pool in a finer grained fashion without increasing the write latency. In fact, this allows us to utilize the log area more efficiently, because the amount of log data a dirty page carries before being evicted is typically much smaller than a 512 byte sector (in the range of 50 to 150 bytes). This will in turn lead to much less frequent merge operations. Note that the size of an in-memory log sector is also reduced to 128 bytes in Figure 1(b).

- With PCRAM storing log data, the average latency of flushing log records from the buffer pool will be reduced considerably. This is because the write speed of PCRAM is (or will be) higher than that of flash memory. This effect will be amplified by the fact that the PCRAM write time is almost linearly proportional to the amount of data to write and the size of a write will be approximately one fourth of a 512 byte sector on average.

- PCRAM is faster than flash memory for small reads by at least an order of magnitude. Besides, when the current version of a data page needs to be computed, its log records can be fetched from PCRAM simultaneously while the old version of the data page is read from flash memory. Thus, the read overhead by the new IPL can be reduced to a negligible extent.

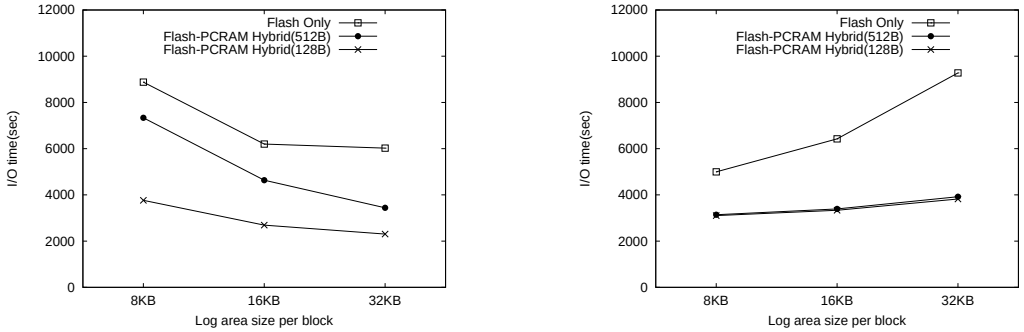
In addition to the performance aspects stated above, this new design makes IPL more pragmatic with existing flash memory solid state drives. With PCRAM being used to store log records, there is no need for partial programming or writing in a unit smaller than a page with flash memory SSDs any longer. Therefore, regardless of its type - SLC or MLC - any flash memory SSD can be used as a storage device for a database system with the IPL capability.

4 Preliminary Performance Evaluations

This section reports preliminary experimental results from a trace-driven simulation to demonstrate the effectiveness of the flash-PCRAM IPL design. We implemented an IPL module to the B^+-tree based Berkeley DB and ran it to obtain trace data for the simulation. Five million key-value records were inserted in random order, and the key-value records were retrieved again in random order. The size of each record was in the range of 40 to 50 bytes, and the database size was approximately 360 MB. The page size was 8 KB, and the buffer size was set to 20 MB. The trace obtained from the run included all the IPL related operations such as page reads, page writes, log sector writes, block merges and erasures.

In the experiment, we used the performance parameters of flash memory and PCRAM given in Table 1. Note that, in the case of flash-only IPL, the latency of a 512 byte sector write was set to 213 μs instead of 250 μs, because the amount of data to transfer over the channel was one fourth of what a full page write would do. When the size of an in-memory log sector was set to 512 bytes, the amount of valid log data flushed by a single write operation was 135 bytes on average, which was a little larger than the average size of a log write observed in the TPC-C benchmark (a little less than 100 bytes) [9]. Note that the size of a physical log write was still 512 bytes by flash-only IPL (as well as flash-PCRAM IPL (512B) that will be shown below).

Figure 2 shows the simulated runtime of the flash-only IPL and the flash-PCRAM IPL for the trace. In the figure, we plotted two cases of the flash-PCRAM IPL with an in-memory log sector of different sizes: 512 bytes and 128 bytes. Thus, in the latter case, log data were flushed to PCRAM more often in finer grained manner. In Figure 2(a), for the random insertion workload, the flash-PCRAM IPL (512B) outperformed the flash-only IPL 17 to 43 percent. This performance improvement was gained mostly by PCRAM whose latency for reading log data was shorter than that of flash memory. On top of that, by changing the size of an in-memory log sector from



(a) Random Insertion (b) Random Search

Figure 2: IPL Performance: Flash-only vs. Hybrid (5M records)

512 bytes to 128 bytes, the simulated runtime of flash-PCRAM IPL was reduced further up to almost 50 percent. Such a high percentage of reduction was due to the higher utilization of log areas achieved by finer-grained log writes, which in turn reduced block merge operations by more than a factor of two.

In Figure 2(b), for the random retrieval workload, the two cases of flash-PCRAM IPL yielded almost identical performance because the high speed of PCRAM read operations made the overhead of reading log data insignificant. For the same reason, the flash-PCRAM IPL outperformed the flash-only IPL with by a significant margin, because the overhead of reading log data from flash memory was not negligible and increased as the size of a log area increased.

5 Conclusions

There are many promising non-volatile memory technologies under active development. They are being considered yet another serious alternative to disk drives as much as NAND type flash memory is. In this paper, we propose a new In-Page Logging (IPL) design that uses PCRAM, a leading candidate of non-volatile memory technologies, as a storage medium for log records. The low latency and byte-addressability of PCRAM can make up for the limitations of flash-only IPL. The preliminary evaluation results show that the proposed design can accelerate the performance of IPL significantly even with the contemporary PCRAM products that do not deliver the promised performance for write operations as yet. This work provides a model case of hybrid storage design based on flash memory and PCRAM.

Acknowledgment

This work was partly supported by MEST, Korea under NRF Grant (No.2010-0025649) and NRF Grant (No.2010-0026511). This work was also sponsored in part by the U.S. National Science Foundation Grant IIS-0848503. The authors assume all responsibility for the contents of the paper.

References

[1] M. Canim, G. A. Mihaila, B. Bhattacharjee, K. A. Ross, and C. A. Lang. SSD Bufferpool Extensions for Database Systems. *Proceedings of the VLDB Endowment*, 3(2), 2010.

[2] J. Condit, E. B. Nightingale, C. Frost, E. Ipek, B. Lee, D. Burger, and D. Coetzee. Better I/O Through Byte-Addressable, Persistent Memory. In *Proceedings of the ACM Symposium on Operating Systems Principles (SOSP)*, 2009.

[3] R. F. Freitas. Storage Class Memory: Technology, Systems and Applications (invited talk). In *Proceedings of ACM SIGMOD*, 2009.

[4] Intel Corp. Understanding the Flash Translation Layer (FTL) Specification. Application Note AP-684, Dec. 1998.

[5] Intel Corp. OLTP performance comparison: Solid-state drives vs. hard disk drives. Test report, Jan. 2009.

[6] International Technology Roadmap for Semiconductors. Process Integration, Devices, and Structures. 2009 Edition.

[7] B. C. Lee, E. Ipek, O. Mutlu, and D. Burger. Architecting Phase Change Memory as a Scalable DRAM Alternative. In *Proceedings of the 36th Annual International Symposium on Computer Architecture*, pages 2–13, June 2009.

[8] K.-J. Lee et al. A 90 nm 1.8 V 512 Mb Diode-Switch PRAM With 266 MB/s Read Throughput. *Solid-State Circuits, IEEE Journal of*, 43(1):150–162, Jan. 2008.

[9] S.-W. Lee and B. Moon. Design of Flash-Based DBMS: An In-Page Logging Approach. In *Proceedings of ACM SIGMOD*, June 2007.

[10] S.-W. Lee, B. Moon, and C. Park. Advances in Flash Memory SSD Technology for Enterprise Database Applications. In *Proceedings of the 35th SIGMOD international conference on Management of data*, pages 863–870, 2009.

[11] S.-W. Lee, B. Moon, C. Park, J.-M. Kim, and S.-W. Kim. A Case for Flash Memory SSD in Enterprise Database Applications. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1075–1086, 2008.

[12] Micron Technology, Inc. NAND Flash 101: An Introduction to NAND Flash and How to Design It into Your Next Product (Rev. B). Technical Note TN-29-19, Apr. 2010.

[13] Numonyx. Omneo P8P 128-Mbit Parallel Phase Change Memory. Data Sheet 316144-06, Apr. 2010.

[14] M. K. Qureshi, V. Srinivasan, and J. A. Rivers. Scalable High Performance Main Memory System Using Phase-Change Memory Technology. In *Proceedings of the 36th annual international symposium on Computer architecture (ISCA)*, 2009.

[15] Samsung Electronics. 2G x 8 Bit NAND Flash Memory (K9F8G08U0M). Data sheet, 2007.

[16] Samsung Electronics. 4Gb A-die DDR3L SDRAM (K4B4G0446A). Data sheet(rev. 1.01), Nov. 2010.