Microprocessors and Microsystems 37 (2013) 24–32

Contents lists available at SciVerse ScienceDirect

Microprocessors and Microsystems

journal homepage: www.elsevier.com/locate/micpro

PTL: PRAM translation layer ☆

Gyu Sang Choi [a], Byung-Won On [b,*], Kwonhue Choi [a], Sungwon Yi [c]

[a] Department of Information and Communication Engineering, Yeungnam University, 214-1, Dae-dong, Gyeongsan, Gyeongsangbuk 712-749, Republic of Korea
[b] Advanced Institutes of Convergence Technology (AICT), Seoul National University, 864-1 lui-dong, Yeongtong-gu, Suwon-si, Gyeonggi-do 443-270, Republic of Korea
[c] Electronics and Telecommunications Research Institute, 138 Gajeongno, Yuseong-gu, Daejeon 305-700, Republic of Korea

ARTICLE INFO

Article history:
Available online 21 August 2012

Keywords:
PRAM
PRAM translation layer
Storage systems
NAND Flash memory
Flash translation layer

ABSTRACT

In this paper, we attempt to replace NAND Flash memory with PRAM, while PRAM initially targets replacing NOR Flash memory. To achieve it, we need to handle wear-leveling issue of PRAM since the maximum number of writes in PRAM is only 10^6. Thus, we have proposed PRAM Translation Layer (PTL) to resolve endurance problem for a PRAM-based storage system. We modified FlashSim to support both PRAM and NAND Flash memory and measured the performance by using real workloads from PC and server.

In our experiment, PRAM shows up to 300% performance improvement compared to NAND Flash memory. Moreover, our results revealed that the PRAM's endurance is improved up to 25% compared to NAND Flash memory due to no erase operation. All these results suggest that PRAM is a viable candidate to replace NAND Flash memory.

1. Introduction

Non-Volatile (NV) memory is one of promising memory types since it can provide read/write latency as short as DRAM and its contents are not deleted after power-off, while DRAM cannot keep its contents after power-off. Due to these advantages, research on NVRAM has been rapidly increasing.

NAND Flash memory [1] is one of these NV memories to achieve increasing density and lower price, and starts replacing HDD due to its high performance and strong reliability, while its price is still almost 5 or 10 times more expensive than that of HDD. NAND Flash memory, however, has very unique characteristics compared to DRAM and HDD.

Among next-generation NV memory, Phase-Cahgne RAM (PRAM) has been developed with several advantages compared to NAND Flash memory. First, PRAM has no erase operation, which means that data can be read/written a certain area without the erase operation. Second, there is no block unit and PRAM is byte-addressable. In NAND Flash memory, the minimum read/write is a page which might be 2 or 4 kB. It means that we have to write a whole page, even if we need to update only few bytes within the page. In contrast, we can update only the area for new writes in PRAM. In addition, when it needs to read a few words, we can read only these words in PRAM, while we should read a whole page which includes these words in NAND Flash memory. Third, PRAM also has the limited number of writes on a certain area up to 10^6, while its number increases 10 times compared to NAND Flash memory. Due to these advantages, Lee et al. [2] proposed PRAM as a DRAM alternative, but they cannot perfectly address PRAM endurance problem, since they only guarantee up to 5 years to use PRAM with their schemes. In addition, there are only few studies [3,4] to use PRAM with NAND Flash memory to alleviate the disadvantages of NAND Flash memory.

In this paper, we have proposed a PRAM-based storage system as NAND Flash memory alternative. In order to use PRAM as storage, we first need to address PRAM endurance limitation. Thus, we propose PRAM Translation Layer (PTL) which dynamically translates logical address to physical address or vice versa, while this address translation scheme cannot use PRAM as DRAM alternative. This is because main memory needs really short read/write latency and the read/write latency in PRAM should be significantly slowed down. Thus, we believe that PRAM is a prominent candidate to replace NAND Flash memory, instead of main memory. In this paper, we propose the new mapping scheme and estimated its space requirement using an analytical model.

We used simulations to measure the performance of the proposed scheme in this paper. We modified FlashSim [5] which was originally developed to simulate NAND Flash memory to support PRAM instead of NAND Flash memory. We model three different 32 GB PRAM configurations which are carefully chosen from [6–11]. In our experiments, we choose two kinds of workloads which represent PC and server systems. The server workloads [12] which consist of WebSearch and OLTP traces are collected from popular search engines and large financial institutions by Storage Performance Council, while PC workloads are collected

☆ This work was supported by the 2011 Yeungnam University Research Grant.
* Corresponding author.
E-mail addresses: castchoi@ynu.ac.kr (G.S. Choi), bwon@snu.ac.kr (B.-W. On), gonew@ynu.ac.kr (K. Choi), sungyi@etri.re.kr (S. Yi).

0141-9331/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.micpro.2012.07.002

by Dirik and Jacob [13] from their laptops and desktops. With these workloads, we have conducted performance evaluation of both PRAM and NAND Flash memory, using the modified FlashSim.

Our performance evaluation concludes several important indications. First, PRAM shows up to 300% performance improvement compared to NAND Flash memory. This is because PRAM has shorter read/write latency than NAND Flash memory. Second, PRAM's endurance is improved up to 25% compared to NAND Flash memory. This is because PRAM has no erase operation which can reduce the number of writes in PRAM. Finally, there is little performance difference between PRAM and NAND Flash memory in write-intensive workloads, while PRAM shows better performance in read-intensive workloads, compared to NAND Flash memory. This is because PRAM show shorter read latency but slower write latency compared to NAND Flash memory. Thus, all these results suggest that PRAM will be a prominent candidate to replace NAND Flash memory alternative in storage systems.

The remainder of this paper is organized as follows. Section 2 explains background and related works, and the proposed PTL is covered in Section 3. In Section 4, we present performance evaluation in PRAM and the conclusions of this study are described in Section 5.

2. Background and related works

In this section, we summarize prior studies on Phase-change RAM (PRAM or PCM) and NAND Flash memory.

2.1. PRAM

Recently, Samsung shipped first 512 Mbits Multi-chip Package (MCP) based on PRAM for handsets in 2010, while Samsung [14,15] introduced a prototype of PRAM in 2006 and Intel [16] sampled PRAM in 2007. PRAM initially targets replacing NOR Flash memory because PRAM provides a relatively slow write latency but a short read latency. Thus, there are only few studies [2–4] to use PRAM as main memory instead of DRAM, or storage instead of NAND Flash memory. PFFS [4] stores meta-data of NAND Flash memory into PRAM because writes into meta-data are mainly a few words and the read/write unit in PRAM is a word, while the NAND Flash memory must perform a read/write per page. Due to storing meta-data into PRAM, the PFFS shows 25% performance improvement compared to YAFFS. However, this scheme did not perfectly address a wear-level issue in PRAM, even though the authors proposed a smart wear-leveling scheme. Doh et al. [3] analyzed memory requirement of NVRAM in the combination of NAND Flash memory and NVRAM. In their research, they showed significant performance improvement with the combination of Ferroelectric RAM (FeRAM or FRAM) [17] and NAND Flash memory, compared to only NAND Flash memory. While advantage of FeRAM is no need to address wear-leveling since the maximum number of read/writes in FeRAM is around 10^{15}, its disadvantage is low scalability and high cost. Lee et al. [2] exploited the possibility to use PRAM as a DRAM alternative and concluded that PRAM shows 1.2 times slower performance and 1.0 times as much as DRAMs energy consumption. However, they did not perfectly solve a wear-leveling issue of PRAM and it provides 5.6 years of lifetime.

2.2. NAND Flash memory

NAND Flash memory is one of non-volatile memories to achieve increasing density and lowering the price, and starts replacing HDD due to its high performance and strong reliability. However, it has very unique characteristics compared to DRAM and HDD. First of all, it has the erase-before-write limitation, which means that an erase operation must be performed on a certain memory area before we write data on its area in NAND Flash memory. This erase operation consumes around 2 ms and so is a major performance bottleneck in NAND Flash memory. Second, the erase operation is performed on a block, while a read/write operation is executed on a page. A block consists of multiple pages (i.e., 32 or 64 pages) in NAND Flash memory. Third, NAND Flash memory limits the number of writes on a specific page up to 10^5. It means that if we keep writing data on the same page, we cannot write the data anymore on the page after 10^5 writes. According to these characteristics, Flash Translation Layer (FTL) which dynamically translates between logical address and physical address had been proposed to address NAND Flash memory limitations. There are three types of mapping schemes in NAND Flash memory [1]: page-level, block-level and log-based schemes.

First, a page-level mapping scheme [1,18,19] conducts the address translation per page which is typically 2 or 4 kB. Fig. 1 shows how to translate LA to PA in the page-level mapping scheme. When a read request for a logical page 1 comes to an Solid-State Disk (SSD), it first accesses the mapping table to translate the logical page 1 to the physical page 2. Then, it reads the requested data from the physical page 2 in NAND Flash memory. When a write request for a logical page 1 comes to an SSD, it first accesses the mapping table to find an empty physical page which is 5. After finding it, the requested data were written on the physical page

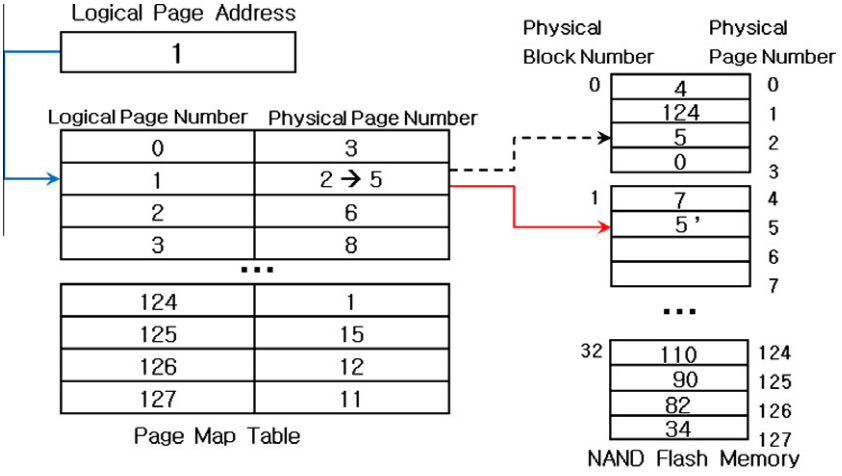

Fig. 1. A page-level mapping scheme.