Energy-Aware Writes to Non-Volatile Main Memory

Jie Chen Ron C. Chiang H. Howie Huang Guru Venkataramani
Department of Electrical and Computer Engineering
George Washington University, Washington DC

ABSTRACT

Scalability challenges of DRAM technology call for advances in emerging memory technologies, among which Phase Change Memory (PCM) has received considerable attention due to its non-volatility, storage density and capacity advantages. The drawbacks of PCM include limited write endurance and high power consumption for write operations (upto 10× in comparison to read operations). In this paper, we investigate new techniques that would perform writes to PCM with energy awareness. Our results show that we can minimize the write energy consumption by up to 8.1× by simply converting PCM native writes to read-before-write, and upto an additional 22.9% via intelligent out-of-position updates.

1. INTRODUCTION

Current memory technology has begun to face challenges in storage density, capacity and energy efficiency for incorporation into high performance multi-core processors. Conventional technology, such as Dynamic Random Access Memory (DRAM), is unable to deliver sustained scaling to feature sizes smaller than 40 nm due to technology limitations [13]. This creates a necessity to look for DRAM alternatives in future multi-core machines. Among the many emerging memory technologies, Phase Change Memory (PCM) has received considerable attention as a replacement for current DRAM-based memory due to non-volatility, storage density and capacity advantages [11, 14, 15]. PCM is projected to scale up to 9 nm [13, 16]. This translates to having at least four times the storage density of current DRAM. Further, PCM has memory access latencies that are only 2× to 4× slower than DRAM [14].

Unfortunately, PCM also has a number of drawbacks such as limited write endurance and high power consumption for write operations (upto 10× in comparison to read operations [9]). PCM is made up of chalcogenide glass material that can be in one of two states- crystalline (low resistance or SET) and amorphous (high resistance or RESET). The PCM material is inserted between two electrodes and a heating element from the bottom electrode establishes contact with the PCM material. Electric current is injected into the junction of the PCM material and the heating element to introduce phase change. In particular, the RESET operation (conversion from crystalline to amorphous state) requires very high voltage, which needs delivery of high power. Above certain threshold voltage, the conductivity of the PCM material rapidly increases leading to large current flows, and consequently heats the PCM to very high temperatures. This creates power and thermal problems, especially when there are a lot of write operations to be performed.

The main goal of this work is to investigate new techniques that would perform writes to PCM with *energy awareness*. Write operations account for 10-40% of memory operations [8]. As the popularity of PCM continues to grow, the power consumption problem imposed by the write operations could severely limit its adoption in the future (Table 1). Especially, a higher ratio of RESET to SET within write operations could lead to higher energy. In this work, we study how to perform intelligent PCM writes such that the total energy consumed for memory accesses can be minimized. Our experimental results show that we can reduce write energy by up to 8.1× through simply converting

HotPower'11, October 23, 2011, Cascais, Portugal.

PCM native writes to read-before-write and get an additional 22.9% savings through intelligent out-of-position updates in industry-standard applications from SPEC 2006 benchmark suite [18].

The contributions of our work are:

1. We propose new energy-aware, out-of-position update techniques to do PCM memory writes such that the energy consumption per write can be minimized.
2. We explore low-cost, practically implementable solutions for out-of-position PCM writes so that our techniques could be incorporated into real hardware.
3. We perform experiments to show energy savings of our proposed schemes using SESC simulator [17] and SPEC 2006 benchmarks [18].

2. BACKGROUND AND MOTIVATION

Before data centers and high performance computing facilities begin to deploy this PCM technology, it is critical to understand the constraints posed by PCM. PCM endures only a limited number of writes before failure. Current projections place the number at $\sim 10^8$ (note that this is much higher than Flash, e.g., $\sim 10^6$ writes in NOR flash) [15]. To solve this, wear leveling algorithms evenly distribute the writes across PCM based memory [14]. Even with wear leveling, power consumption will be a major hurdle leading to reduced bandwidth on account of the high current needed for programming PCM cells. In this work, we will show that a good PCM design shall also take into account the power consumption of PCM writes by minimizing the number of bit flips (i.e., number of SET and RESET operations per write). By reducing the number of write operations to PCM cells for power savings, we note that indirect benefits of extended PCM lifetime can also be achieved. Thus, we envision that integrating our proposed techniques into existing wear leveling algorithms will be useful to have both lifetime and power/energy benefits.

Table 1: DRAM and PCM energy

	DRAM	**PCM**
Read Energy (pJ/bit)	4.4	2.47
Write Energy (pJ/bit)	5.5	14.03 (set) 19.73 (reset)

Table 1 shows the comparison between PCM and DRAM in terms of read and write energy needs. We obtained the energy numbers for DRAM via CACTI 5.3 [10]; for PCM we quote the specifications used by Lee et al [11]. As shown, write energy per bit is significantly higher than read for PCM, while DRAM consumes approximately the same energy for read and write operations.

Prior works have studied incorporating PCM as a DRAM alternative [11], as well as, using a hybrid PCM-DRAM main memory [15]. Architecture modifications have been studied to reduce PCM energy consumption by minimizing write traffic and through efficient bit flipping strategies [3], removing redundant writes [20], reorganizing row buffer [11], and using DRAM buffer to filter out PCM accesses [15]. Condit et al. [4] show how to build a new file system with significantly better performance based on byte-addressable persistent memory. PCM based systems have also been designed for specific domains such as database design [2]. While most prior studies focus on extending lifetime and reducing latency of PCM, our work proposes new systems-level techniques to minimize energy consumption.

3. EXPERIMENTAL SETUP

We evaluate the performance and energy impact resulting from energy-aware PCM using SESC [17], a cycle-accurate, execution driven simulator. Our baseline models a Intel Nehalem like four core processor [5] running at 3 GHz, 4 way, out-of-order core, each with a private 32KB, 8 way set-associative L1 and a shared 4MB, 16 way set associative L2. L1 caches are kept coherent using the MESI protocol. The block size is 64 bytes in all caches. We model 4 GB PCM main memory with 4KB pages with read access latency of 50 ns and write access latency of 1 μs [12]. Our read and write energy numbers are shown in Table 1. In this paper, we show the results on six memory intensive benchmarks from SPEC2006 benchmark suite [18], namely astar, gcc, mcf, omnetpp, perlbench, and soplex. We use the train input sets and run the benchmarks from start to end.

4. ENERGY-AWARE WRITES TO PCM

PCM writes can consume upto 10× more energy compared to reads [9]. To address this, PCM write operations are changed into a new read-before-write sequence, where we first determine whether data values have changed before performing the energy-expensive PCM write operation. Yang et al. [19] have shown that the additional read operation has only 1% performance and 0.2% energy overheads, while being able to reduce the write power to a half. Our experiments (Figure 1) show energy savings between 7.6× - 8.1× by using read-before-write compared to native write operations. PCM energy

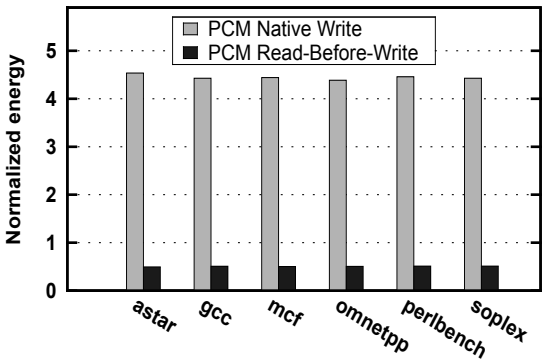


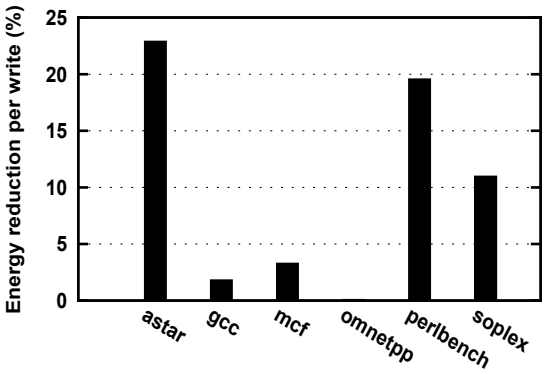
Figure 1: Write Energy Consumption of PCM-based Memory normalized to DRAM

consumption is normalized to a baseline DRAM system. Also, PCM read-before-write only consumes about half of the energy compared to DRAM.

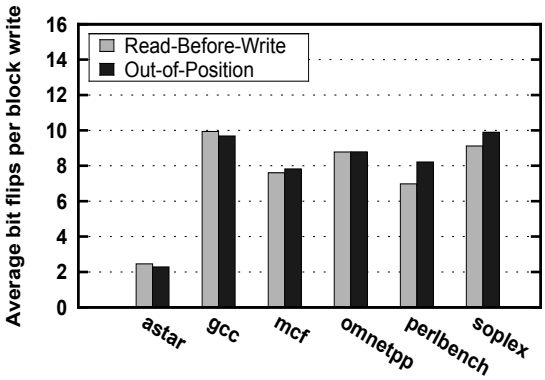
Beyond PCM read-before-write, we explore further opportunities for energy savings. In particular, we note that RESET consumes about 1.5× more energy than SET operations. Thus, it is potentially beneficial to reduce the number of RESET operations, and explore avenues to further reduce the overall energy consumption. Toward this goal, we choose to perform energy-aware, out-of-position updates to data blocks. We pick a free (invalid) PCM page that consumes the lowest write energy and remap the data block to that page. Note that out-of-position update is an optimization on top of in-position PCM read-before-write. Hence, we assume that out-of-position PCM update also has PCM read-before-write feature. In Figure 2(a), we observe upto 22% energy reduction over in-position PCM read-before-write in applications like astar. We also measure the average number of SET (Figure 2(b)) and RESET (Figure 2(c)) operations needed for every 64 byte block update. Our results show that the average number of RESET operations decreases in out-of-position updates in all applications. In some applications like soplex, we notice a slight increase in the average number of SET operations during block update. Despite this phenomenon, the overall write energy consumption decreases predominantly because of the reduction in the average number of RESETs per write.

5. PRACTICAL IMPLEMENTATION

An important consideration for energy-aware, out-of-position updates is choosing the ideal out-of-position PCM block that will consume the least write energy. In light of this requirement, a straightforward solution is to perform exhaustive search of all invalid PCM blocks. However, such an approach would be time consuming, and we would need to spend energy to read out all possible candidate PCM blocks and compute the amount of write energy needed for bit flips to identify the target out-of-position candidate block. As evident, too many read operations on free (invalid) PCM blocks would diminish any potential energy savings obtainable through out-of-position PCM block update. Therefore, we need an intelligent, low cost way to quickly



(a) Energy savings of out-of-position PCM read-before-write over in-position PCM read-before-write



(b) Average number of bit SET per Block Write

(c) Average number of bit RESET per Block Write

Figure 2: Comparison between in-position and out-of-position PCM read-before-write

determine a good out-of-position PCM block that would offer the best energy savings.

To this end, we explore a combination of approximation and sampling techniques and devise a LSH (Locality-Sensitive Hashing) based algorithm that is often used for similarity detection and nearest neighbor search [1, 6]. LSH utilizes various hash functions to group different blocks so as to the blocks in the same group will have high probability that share most common bits. In this work, we use a simple prime number based bit sampling technique to measure the distance between two blocks (number of bit flips), and effectively estimate the write energy needed to update the target block. We find that this technique is very efficient and effective in locating a replacement block and yields comparable energy savings to our ideal energy-aware, out-of-position writes. Figure 3 shows the results of write energy savings by using LSH-based approximation for selecting out-of-position blocks. The baseline assumes that writes are performed as in-position PCM read-before-write operations. Clearly, the energy savings in LSH is comparable to the ideal scheme (Figure 2(a)), while the cost of using LSH scheme is very small.

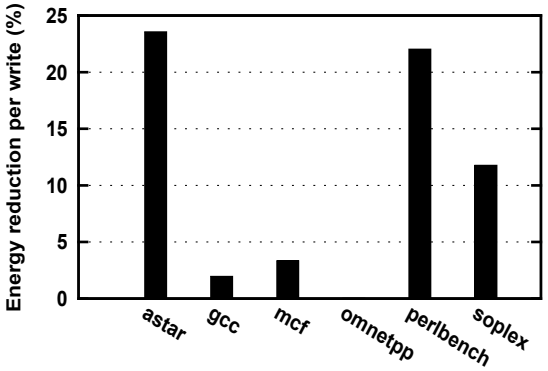


Figure 3: Energy Savings using LSH-based algorithm over PCM read-before-write

Another consideration while performing out-of-position update is that we will need to remap the chosen target PCM page out-of-position candidate to correctly point to the data page that is about to be written to it. A possible solution to solve this problem is to use the real address that denotes actual PCM page numbers and these real addresses are different from system physical addresses. To reduce the hardware complexity, the operating system (OS) can manage the physical-real address translation and at the same time, such mappings can be temporarily stored (cached) inside the memory

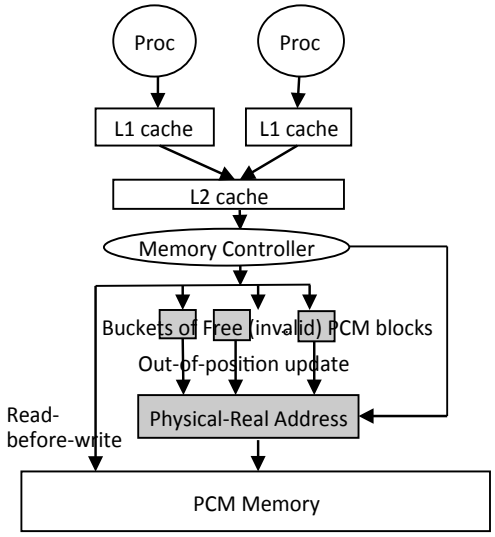


Figure 4: Architecture Modifications to incorporate energy-aware PCM write

controller to minimize the impact on program performance. This approach is similar to Dynamically Replicated Memory [7], where mappings of paired PCM pages containing replicated data are stored for fault tolerance purposes.

Figure 4 shows architecture modifications needed to implement energy-aware writes in PCM using LSH-based algorithm. We modify the memory controller hardware to include four 16 entry buffers (that serve as buckets to group free PCM blocks with similar numbers of ones and zeros to aid easy replacement). As shown in our experiments, such small buffers provide the sufficient amount of space, as well as, lower read energy needs to quickly identify a suitable out-of-position PCM block. Once the out-of-position PCM block is identified, the mapping information between the system's physical address and the PCM real address should be stored for future memory references. The OS can offer such services, however, an OS invocation for every ensuing memory read for a remapped PCM page can be very expensive. For memory reads, we provide a temporary cache of 64 entries to store frequently used physical-real address mappings to minimize performance impact. For memory writes, however, we mandate OS invocation as they could potentially involve access an out-of-position PCM block. Further optimizations such as aggressive filtering of certain memory writes and heuristic-based strategies to avoid physical-real address remapping are possible. Also, we note that out-of-position write operations may not likely affect application performance since writes are off the critical path. Additional

optimizations such as memory buffer can hide expensive latencies exposed by out-of-position writes, and help us to realize the energy benefits of our proposed approach.

6. CONCLUSIONS

In this paper, we propose novel techniques to save energy consumed by PCM write operations. We explore energy-aware, out-of-position PCM block updates where the energy consumed per write can be minimized. Our experimental results show that upto 8.1× energy savings can be achieved by simply converting PCM writes into read-before-write, and further energy savings of upto 22.9% can be achieved from intelligently performing out-of-position PCM block writes. We also study how our proposed mechanisms can be implemented in real hardware.

As future work, we plan to investigate how to incorporate low-cost, performance-friendly techniques such that out-of-position PCM writes can be performed without excessive performance degradation. We will also investigate energy saving measures on other promising resistive memory technologies like memristors that have the potential to replace DRAM.

7. ACKNOWLEDGMENTS

This material is based upon work supported in part by the National Science Foundation under Grant No. CCF-1117243 and OCI-0937875.

8. REFERENCES

[1] A. Andoni and P. Indyk. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. In *MFCS*, 2006.

[2] S. Chen, P. B. Gibbons, and S. Nath. Rethinking database algorithms for phase change memory. In *CIDR*, 2011.

[3] S. Cho and H. Lee. Flip-n-write: a simple deterministic technique to improve pram write performance, energy and endurance. In *MICRO*, 2009.

[4] E. B. Condit, J.and Nightingale, C. Frost, E. Ipek, B. Lee, D. Burger, and D. Coetzee. Better i/o through byte-addressable, persistent memory. In *SOSP*, 2009.

[5] Intel Corporation. Intel core i7-920 processor. *http://ark.intel.com/Product.aspx?id=37147*, 2010.

[6] A. Gionis, P. Indyk, and R. Motwani. Similarity search in high dimensions via hashing. In *VLDB*, 1999.

[7] E. Ipek, J. Condit, E. B. Nightingale, D. Burger, and T. Moscibroda. Dynamically replicated memory: building reliable systems from nanoscale resistive memories. In *ASPLOS*, 2010.

[8] A Jaleel. Memory characterization of workloads using instrumentation driven simulation. *http://http://www.glue.umd.edu/ ajaleel/workload/*, 2010.

[9] Y. Joo, D. Niu, X. Dong, G. Sun, N. Chang, and Y. Xie. Energy- and endurance-aware design of phase change memory caches. In *DATE*, 2010.

[10] HP Labs. Cacti 5.3. *http://quid.hpl.hp.com:9081/cacti/*, 2010.

[11] B. C. Lee, E. Ipek, O. Mutlu, and D. Burger. Architecting phase change memory as a scalable dram alternative. In *ISCA*, 2009.

[12] Numonyx. Phase change memory: A new memory to enable new memory usage models. *White Paper http://www.numonyx.com/*, 2009.

[13] Devices Process Integration and Structures. International technology roadmap for semiconductors. *http://www.itrs.net*, 2007.

[14] M. K. Qureshi, J. Karidis, M. Franceschini, V. Srinivasan, L. Lastras, and B. Abali. Enhancing lifetime and security of pcm-based main memory with start-gap wear leveling. In *MICRO*, 2009.

[15] M. K. Qureshi, V. Srinivasan, and J. A. Rivers. Scalable high performance main memory system using phase-change memory technology. In *ISCA*, 2009.

[16] S. Raoux, G. W. Burr, M. J. Breitwisch, and et al. Phase-change random access memory: a scalable technology. *IBM J. Res. Dev.*, 52, July 2008.

[17] Jose Renau et al. SESC. *http://sesc.sourceforge.net*, 2006.

[18] Standard Performance Evaluation Corporation. SPEC Benchmarks. *http://www.spec.org*, 2006.

[19] B. Yang, J. Lee, J. Kim, J. Cho, S. Lee, and B. Yu. A low power phase change random access memory using a data comparison write scheme. In *ISCAS*, 2007.

[20] P. Zhou, B. Zhao, J. Yang, and Y. Zhang. A durable and energy efficient main memory using phase change memory technology. In *ISCA*, 2009.