

rPRAM: Exploring Redundancy Techniques to Improve Lifetime of PCM-based Main memory

Jie Chen, Zachary Winter, Guru Venkataramani, H. Howie Huang
 Department of Electrical and Computer Engineering,
 The George Washington University, Washington DC, USA

Abstract—Future main memory systems will confront the scaling challenges posed by DRAM technology and should adapt themselves to use the emerging memory technologies like Phase Change Memory (PCM, or PRAM). PCM offers advantages such as storage density, non-volatility, and lower energy consumption. However, they are constrained by limited write endurance and reduced performance. In this paper, we propose a novel PCM-based main memory system, rPRAM, which leverages a group of faulty pages in a managed way to significantly extend the PCM life while minimizing the performance impact. Our preliminary experiments show that rPRAM has the potential to extend the lifetime of PCM based memory comparable to existing schemes at only a negligible fraction of hardware cost.

I. INTRODUCTION

With multi-core processors rapidly becoming mainstream, offering sufficient memory bandwidth and storage density to satisfy the applications' demands has become a big challenge for DRAM systems. DRAM technology is also facing technology limitations in scaling beyond 30 nm [4]. Therefore, it is imperative to explore emerging resistive-memory types such as Phase Change Memory (PCM) as alternatives to DRAM.

PCM, in particular, has been shown to exhibit enormous potential as a DRAM replacement because they scale to smaller features sizes like 9nm, can offer up to 4X more density at only small orders of magnitude (up to 4X) slowdown in performance [7]. Consequently, PCM-based hybrid memory have been proposed for future generation main memory [6]. A major challenge when using PCM arises from its limited write endurance. PCM-based devices are expected to sustain an average of 10^8 writes per cell, when the cell's programming element breaks and the write operations can no longer change the values. Current solutions focus on wear-leveling [5] and reducing the number of writes to PCM [3]. However, once a PCM page begins to exhibit faults, it will be discarded as unusable by the memory controller.

To the best of our knowledge, Dynamically Replicated Memory (DRM) [1] was the first technique to rejuvenate pages that had faults and put them back to use for data storage. DRM picks pairs of faulty pages that do not have faults in the same bit position and stores replicated data in both pages. This redundant storage helps to recover the original data through reading the non-faulty byte from at least one of the paired pages. DRM leverages on the high probability of finding two compatible faulty pages and hence, one could eventually reclaim what would be decommissioned memory space. A caveat with this approach is that *simply replicating the data*

in both pages can rapidly degrade the effective capacity of the memory system. Another recent proposal, Error Correcting Pointers [8], handles errors by encoding the locations of failed cells in a table and by assigning new cells to replace them. The disadvantage with this approach is the high cost and complexity in redesigning the PCM chip specially to accommodate the dedicated ECP pointers. ECP incurs a static area overhead of about 12% just to store these pointers.

Our premise is that, we need to invent a low cost and efficient way to reuse faulty pages in a managed way to significantly extend PCM life. To this end, we propose *rPRAM* (redundancy PRAM), that explores extending PRAM device lifetime based on advanced redundancy techniques inspired by RAID (Redundant Arrays of Inexpensive Disks) technology. While a wide range of levels from RAID 0 to 6 are available, our work specifically adopts a robust approach by forcing the faulty pages to use RAID 4 (block level striping with dedicated parity, where there is a dedicated parity block for every group). As faulty pages begin to incur higher numbers of faults per page we reduce the number of faulty blocks per group due to increased complexity in finding compatible faulty pages (i.e., too many iterations are needed to pick compatible faulty pages). In rPRAM, we leverage two important observations that DRM scheme makes with regards to managing faulty pages: (1) We find two pages compatible only when the corresponding pages do not have faulty bits in the same bit position. (2) The PCM pages are deemed unusable once they have at least 160 bit failures because finding compatible pairs of pages becomes exponentially harder beyond this limit [1].

Our motivation behind exploring parity-based technique to improving the lifetime of PCM was driven by wanting to:

- *Increase the space efficiency in the usage of faulty pages:* For DRM, mirroring replicates data across two faulty pages—the storage density is 50%. In rPRAM, a group of G faulty pages have a dedicated block, P that stores the parity values for all of the G pages. Therefore, the storage density for $G+1$ (including the parity) pages is $G/(G+1)$. For example, at a group size of 3, the storage efficiency is 75%, that is 50% more efficient than DRM scheme. At higher values of G , we get better efficiency in terms of storage density although finding compatible pages for larger G values becomes increasingly difficult.

- *Utilize off-the-shelf memory components without extensive hardware redesign:* Prior techniques such as ECP [8] have to custom design the PCM chip to integrate their pointer-based lifetime-enhancing techniques; whereas, our goal is to

maximize the use of off-the-shelf components and incorporate techniques to enhance lifetime with minimum changes to hardware design. This helps minimize the performance impact on applications, as well as, reduce the cost associated with including our proposed techniques into the existing hardware.

- *Explore design choices that will offer flexibility to the user:* The end user can make an informed choice that is most suitable to her needs under a given cost budget.

II. EXTENDING PCM LIFETIME

In rPRAM, we assume that the PCM-based main memory starts without any faults and does wear-leveling to uniformly distribute the writes. For cases where Error Detection and Correction (ECC) exist on chip, the first few bit faults can be tolerated using pre-built schemes. When the first bit fault (beyond ECC tolerance limit) occurs, we temporarily decommission the PCM page k and place it in a separate pool of faulty PCM pages that are waiting to be matched with other compatible pages. We then find a compatible group of g faulty PCM pages, and store the corresponding parity in a separate high-performance buffer to avoid performance bottleneck. In this work, we use a small DRAM to store parity for faster access, which we believe is more cost-effective than simply having additional PCM pages for parity. This is mainly motivated by two facts: (1) Parity information is much smaller than data itself- typically G data blocks have one corresponding parity block. (2) Parity needs to be accurate and cannot have errors in order to recover the faulty data block.

As number of faults per page increase, it becomes harder to form a group of compatible faulty pages that don't have fault at the same byte positions. Experimentally, we observe that the average number of random trials needed for three-way matching of faulty PCM pages (with at least 80 bit faults) is more than twice the number of trials required for two-way matching. Therefore, in order to bound the complexity associated with matching, we reduce the RAID 4 group size from three to two, once the faulty page incurs more than 80 bit faults.

In our experiments, we measure the lifetime (as the total number of writes to the PCM) in rPRAM and compare it against prior schemes like Fail_Stop (that does not have any Error Correction capabilities and discards a PCM block after the first fault occurs), DRM [1], ECP [8] schemes. We assume a baseline of 4GB PCM Memory with 4KB page size and perform writes at a granularity of 64 Byte blocks. We assume a 50% probability that any single write operation would flip a particular bit. We model the PCM to have lifetimes that follow normal distribution with a mean of 10^8 respectively and variation coefficient of 0.2.

Figure 1 shows our preliminary results. We observe that in a 4 KB page, there is a high probability of at least one byte having a lifetime in the tail-end of the normal distribution. This makes the Fail_Stop scheme to quickly decommission all the pages rapidly and the PCM's effective capacity drops to zero. DRM offers an extended lifetime of approximately 1.86X over Fail_Stop scheme, while ECP achieves a lifetime improvement of 2.68X. rPRAM achieves lifetime improvement of 2.72X,

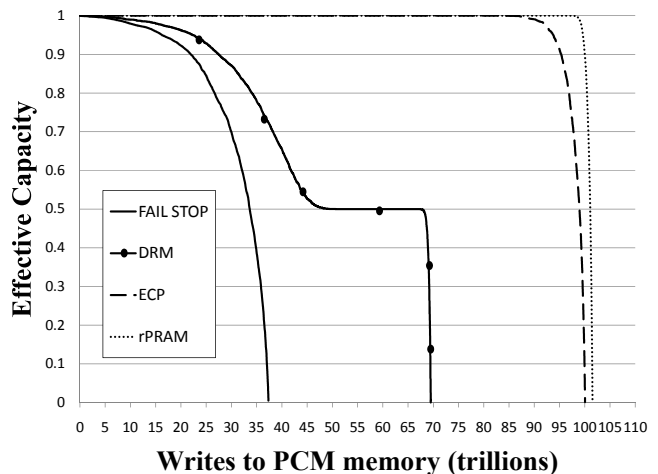


Fig. 1. Effective capacity of PCM Main Memory versus the total number of writes issued to the main memory.

that is comparable to ECP at a fraction of ECP's cost. Using CACTI [2], we estimate that rPRAM incurs <1% area overheads (offchip and onchip) and uses off-the-shelf components without having to custom design the PCM chip.

III. ACKNOWLEDGMENTS

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

REFERENCES

- [1] Engin Ipek, Jeremy Condit, Edmund B. Nightingale, Doug Burger, and Thomas Moscibroda. Dynamically replicated memory: building reliable systems from nanoscale resistive memories. In *Proceedings of the fifteenth edition of ASPLOS on Architectural support for programming languages and operating systems*, ASPLOS '10, pages 3–14, New York, NY, USA, 2010. ACM.
- [2] HP Labs. Cacti 5.3. <http://quid.hpl.hp.com:9081/cacti/>, 2010.
- [3] Benjamin C. Lee, Engin Ipek, Onur Mutlu, and Doug Burger. Architecting phase change memory as a scalable dram alternative. In *Proceedings of the 36th annual international symposium on Computer architecture*, ISCA '09, pages 2–13, New York, NY, USA, 2009. ACM.
- [4] Devices Process Integration and Structures. International technology roadmap for semiconductors. <http://www.itrs.net>, 2007.
- [5] Moinuddin K. Qureshi, John Karidis, Michele Franceschini, Vijayalakshmi Srinivasan, Luis Lastras, and Bulent Abali. Enhancing lifetime and security of pcm-based main memory with start-gap wear leveling. In *Proceedings of the 42nd Annual IEEE/ACM International Symposium on Microarchitecture*, MICRO 42, pages 14–23, New York, NY, USA, 2009. ACM.
- [6] Moinuddin K. Qureshi, Vijayalakshmi Srinivasan, and Jude 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 '09, pages 24–33, New York, NY, USA, 2009. ACM.
- [7] S. Raoux, G. W. Burr, M. J. Breitwisch, C. T. Rettner, Y.-C. Chen, R. M. Shelby, M. Salinga, D. Krebs, S.-H. Chen, H.-L. Lung, and C. H. Lam. Phase-change random access memory: a scalable technology. *IBM J. Res. Dev.*, 52:465–479, July 2008.
- [8] Stuart Schechter, Gabriel H. Loh, Karin Straus, and Doug Burger. Use ecp, not ecc, for hard failures in resistive memories. In *Proceedings of the 37th annual international symposium on Computer architecture*, ISCA '10, pages 141–152, New York, NY, USA, 2010. ACM.