Energy Saving Techniques for Phase Change Memory (PCM)

Sparsh Mittal
Department of Electrical and Computer Engineering
Iowa State University, Ames, Iowa 50011, USA
Email: sparsh0mittal@gmail.com

***Abstract*—In recent years, the energy consumption of computing systems has increased and a large fraction of this energy is consumed in main memory. Towards this, researchers have proposed use of non-volatile memory, such as phase change memory (PCM), which has low read latency and power; and nearly zero leakage power. However, the write latency and power of PCM are very high and this, along with limited write endurance of PCM present significant challenges in enabling wide-spread adoption of PCM. To address this, several architecture-level techniques have been proposed. In this report, we review several techniques to manage power consumption of PCM. We also classify these techniques based on their characteristics to provide insights into them. The aim of this work is encourage researchers to propose even better techniques for improving energy efficiency of PCM based main memory.**

***Index Terms*—Phase change memory (PCM), write power, power management, energy saving, architectural techniques.**

I. Introduction

Modern computing systems provide several 24×7 services and execute data-centric computing applications [1–3], which require large amount of computing resources. Since disk-access is slow, the amount of main memory used in computing systems has increased. However, due to this, the contribution of main memory in system power consumption has also increased, which could be as large as 40% [4]. Also, future scaling of DRAM is in doubt [5] and it low density obstructs design of large sized cache using DRAM.

To address this, researchers have explored alternative memory design technologies such as phase change memory (PCM) [6] etc., which provide high storage density. PCM is a non-volatile device which has high retention property (over 10 years) and very good operation characteristics and scalability [7]. PCM prototypes with feature size as small as 3nm have been fabricated [7]. It also has higher write endurance than the flash memory [8, 9], although its write endurance is several orders of magnitude worse than that of DRAM. The read power and delay of PCM are in the same range as that of DRAM, however, its write power is significantly higher than that of DRAM [10, 11]. Thus, power management of PCM is extremely important to ensure its wide-spread adoption and also avoid the problems due to heating of the system.

In this report, we review several architectural techniques for managing power consumption of PCM. We classify these techniques based on their properties to highlight their similarities and differences. The aim of this work is to provide a synthetic overview of the research field of PCM power management. We believe that this overview will encourage researchers to propose even better techniques for improving energy efficiency of PCM based main memory.

The rest of the report is structured as follows. Section II provides a background on power management in PCM devices. Section III discusses some energy saving techniques in detail. We only discuss the essential idea of the techniques and not the quantitative results, since different techniques have been evaluated using different platforms. Finally, Section IV discusses the future work and provides the conclusion.

II. Background and Related Work

A. Need of Power Management in Main Memory

In recent years, the total power consumption of embedded systems, data centers and supercomputers has significantly increased [12, 13], which has also increased the carbon footprint of IT. To address this, computer architects have proposed several techniques for reducing the energy consumption of computing systems [14, 15]. Since a large fraction (upto 40% [9, 12, 16]) of energy spent in server-class systems is consumed by the main memory, the techniques for saving energy in main memory systems are vital for improving energy efficiency of computing systems.

Since PCM has much smaller leakage energy than DRAM, recently, there has been a significant interest in use of PCM. PCM has been evaluated in context of GPUs (graphics processing unit) [17, 18], embedded systems [19–23], real-time systems [24], video applications [25, 26] and so on. In fact, use of PCM has also been explored for designing caches [27–31]. As the use of PCM increases, managing its power consumption becomes much more important.

B. A Brief Background on Phase Change Memory

We briefly review the design of PCM. A PCM cell comprises an NMOS access transistor and a storage resistor which is made of a chalcogenide alloy [32, 33]. To store a binary value on PCM, heat is applied to it which transitions the physical state of the alloy with particular resistances. When the alloy is heated to a very high temperature (greater than 600 degree Celsius) and quickly cooled down, it transitions into an amorphous substance with high electrical resistance which represents binary "0". On the other hand, when the alloy is heated to a temperature between the crystallization (300 degree Celsius) and melting (600 degree Celsius) points and cools down slowly, it crystallizes to a physical state with lower resistance, which represents binary "1". The difference in resistance values between the two states of PCM is typically 3 orders of magnitude. PCM memories achieve high density by exploiting this high resistance range to store multiple bits in a single cell, this structure is known as multi-level cell or MLC. PCM is byte-addressable and is immune to radiation-induced soft errors.

C. An Overview of Energy Saving Techniques for PCM

Computer architects have proposed several techniques for saving energy in PCM systems. Some researchers have proposed hybrid PCM-DRAM design [17, 18, 33–56]. These techniques aim to achieve the best of both DRAM and PCM, viz. the short latency and high write endurance of DRAM and low leakage power and high density of PCM.

Some techniques convert PCM write operation to read-before-write (or data comparison write) operation to reduce write energy [8, 10, 57–60]. These techniques, referred to as "differential write" based techniques, read out the old value in the PCM array before writing the new one and compare them to write only those bits that need to change. Some researchers propose task-scheduling based techniques to address the challenges in hybrid DRAM-PCM based main memory [41, 56].

Several other techniques are based on reducing write-traffic to PCM memory (e.g. [61]). Some researchers propose last level cache management techniques for improving energy efficiency of PCM main memory [62, 63]. Other researchers have proposed compression based techniques to reduce write-traffic to PCM [43, 64]. Several other techniques aim to address the write latency issue, and its harmful impact on read latency arising due to bank conflicts and try to utilize write locality to coalesce all possible changes to the data by using buffers before they are finally written to PCM [35, 65]. Several wear-leveling techniques which work by reducing the number of writes to PCM also generally save write energy.

PCM also offers the ability to store multiple bits per cell and several researchers propose techniques to achieve this in an energy efficient manner [57, 60, 66–68]. Finally, some researchers have proposed architecture or device-level simulators for studying non-volatile memories [69–73], which facilitate study of PCM.

III. Energy Saving Techniques

The read power and delay of PCM are in the same range as that of DRAM, however, its write power is significantly higher, which can be several times that of DRAM [10, 11]. In contrast, for DRAM, both read and write times are equivalent. Writing to a PCM cell requires high current density over a large period of time. Hence, to ensure correct operation, hard limits on the number of simultaneous writes must be enforced which limits write throughput and overall performance. Thus, failure to save write energy may nullify the energy saving advantage gained due to low leakage power of PCM.

Further, large power consumption of PCM can have deleterious effect on its operation. It may lead to violating power limits, which may in turn lead to voltage drops in the power supply or excessive currents flowing through the processor. It may increase the temperature which may further increase the leakage energy consumption of other components of the system. It may also create logical errors, incomplete PCM phase transitions, PCM read errors, etc. which may lead to chip failures or chip-aging. Thus, power management of PCM is extremely important. In this section, we review several techniques for managing power consumption of PCM.

Hay et al. [5] propose a technique to reduce write power consumption of PCM banks. Their technique is based on the observation that typically only a small portion of the bits (for example, less than 25% on average) are written to which consume power. Their technique monitors the number of bits that will change on a write and hence, need to be written. This gives an estimate of the number of bits and hence, amount of power consumed in a write. Then, to not exceed the power budget, the memory controller issues writes only when there is enough power to support them.

Cho et al. [8] propose a technique named, Flip-N-Write to improve PCM write bandwidth, write energy, and write endurance under an instantaneous write power constraint. Their technique works on the observation that many bit-writes to PCM are redundant. Their technique replaces a write operation with read-modify write operation to skip writing a bit if the bit being written is same as the originally stored bit. Further, to restrict the maximum number of bits which are written, they use a "flip" bit. If storing the flipped value of data requires less number of bit-write operations, their technique stores the data in flipped form and changes the flip bit to ON. Using their technique, the write bandwidth is doubled, which also improves the write endurance and reduces the write energy.

Lee et al. [65] propose using multiple row-buffers inside a PCM chip, which reduces the read latency and also the write energy through write coalescing. Multiple writes to the same location are absorbed in the buffers, thus resulting in much smaller number of write-backs to the PCM array. They also propose a technique which uses multiple dirty bits in the cache blocks to enable partial writes. Using this, the number of bit updates are reduced by not writing untouched, clean data portion in a dirty cache block to the main memory when the cache block is replaced. Their techniques also increase the lifetime of PCM.

Zhang et al. [35] study PCM in the context of 3D die-stacking. Using analytical and circuit-level modeling for PCM characterization, they show that the programming power of PCM cells can be reduced as the chip temperature is elevated. This high-temperature friendly operation of PCM can be advantageously used to design 3D die-stacking memory systems. They propose a hybrid memory design where a large portion of PCM is used as a primary memory space and a small portion of DRAM is used as a write-buffer to reduce the number of writes to PCM. They also propose an OS-level paging scheme that takes into account the memory reference characteristics of applications and migrates the hot-modified pages from PCM to DRAM so that the life time degradation of PCM is alleviated. Their technique also improves the energy efficiency of the memory system.

Ferreira et al. [63] propose a cache replacement policy for saving PCM main memory energy. Their approach aims to reduce the write-back traffic to main memory. The policy is called N-Chance where N can be varied. This policy evicts the least recently accessed clean page from cache, unless all of the N least recently accessed pages are dirty, if so, it evicts the least recently accessed page. For the case when $N = 1$, this policy becomes the conventional LRU (least-recently used) policy. They have shown that for a proper choice of N, their policy can be significantly better than the LRU policy.

Hu et al. [20] propose a technique for reducing the number of writes to PCM main memory. Their technique is based on data migration and re-computation. In an embedded CMP (chip multiprocessor) having scratch-pad memory, their technique migrates data to the scratch-pad memory of a different core to avoid write-backs of shared data. Thus, by temporarily storing the data on scratch-pad, their technique reduces the number of write-backs. Their technique uses program analysis to determine when and where the data should be migrated. They also propose data re-computation to reduce the number of write activities by discarding the data which should have been written back to the main memory and recomputing these data when they are needed. They model the problem of data migration as a shortest path problem. Also, they propose an approach to find the optimal data migration path with minimal cost for both dirty data and clean data.

Qureshi et al. [36] propose a hybrid memory design where PCM memory is augmented with a small DRAM that acts as a "page cache" for the PCM memory. The page cache buffers frequently accessed pages and thus helps performance and improves PCM endurance by reducing the number of writes to PCM with write combining and coalescing. Further, at cache line level, only the lines modified in a page are written to the main memory. Finally, at block-level, swapping is used for achieving wear-leveling. Their technique also reduces the page faults which improves the performance of the system. However, when the applications have poor locality, the advantage of using page cache reduces.

Liu et al. [37] study the variable partitioning problem on a hybrid main memory designed with PCM and DRAM for DSP systems [37]. They propose ILP (integer linear programming) formulations and heuristic algorithms, such that the energy efficiency of PCM can be leveraged while also minimizing the performance and lifetime degradation caused by PCM writes.

Bock et al. [74] propose a technique to save PCM energy and increase its endurance by avoiding useless write-backs. They define a write-back to a lower level cache to be useless when the data that are written back are not used again by the program. As an example, a useless write-back results when a dirty cache line (block) that belongs to a dead memory region is evicted from the cache. Their technique assumes that suitable schemes can be employed to detect dead memory regions in different parts of memory, such as heap, stack and global memory. Assuming that such information is available, their technique estimates the maximum energy savings that could be achieved by avoiding useless write-backs. Further, since writes are not on critical path of execution, avoiding useless write-backs does not have a significant influence on performance.

Mirhoseini et al. [59] propose a coding-level technique for saving PCM energy, which is based on the observation that PCM set and reset energy costs are not equal. Their technique aims at minimizing the energy cost of rewriting to PCM by designing low overhead data encoding methods. Their encoding scheme utilizes PCM bitwise manipulation ability during the word overwrites such that only the bits which are changing for the new word compared to the original word would require overwriting. They propose an ILP (integer linear programming) method and employ dynamic programming to produce codes for uniformly distributed data.

Yoon et al. [75] use a design space exploration approach for finding the optimal configuration of cache hierarchy in a system with non-volatile memory. Use of non-volatile devices enables designing caches with higher capacity and hence, the cache hierarchy in modern systems is becoming deeper with L4 and L5 caches (e.g. off-chip DRAM caches). They consider both performance and power while performing the experiments. They consider cache hierarchy of different depth, where the cache at any level can be designed with either SRAM, DRAM or PCM. Also the main memory could be designed with DRAM or PCM. They observe that a large last level cache (LLC) designed with PCM can improve energy efficiency by reducing the costly off-chip accesses. Also, deep cache hierarchies are less energy efficient than flat hierarchies (2 or 3 levels).

Seok et al. [42] propose a migration based page caching technique for PCM-DRAM hybrid main memory system. Their technique aims to overcome the problem of the long latency and low endurance of PCM. For this, read-bound access pages are kept in PCM and write-bound access pages are kept in DRAM. Their technique uses separate read and write queues for both PCM and DRAM and uses page monitoring to make migration decisions. Write-bound pages are migrated from PCM to DRAM and read-bound pages are migrated from DRAM to PCM. The decision to migrate is taken as follows: when a write access is hit and the accessed page is in PCM write queue, it is migrated. Similarly, if a read access is hit and the accessed page is in the DRAM read queue, it is migrated.

Dhiman et al. [9] propose a hybrid main memory system composed of DRAM and PCM. Their memory system exposes DRAM and PCM addressability to the software (OS). In their technique, data placement is performed based on the write frequency to data. If the number of writes to a PCM page exceeds a threshold, the contents of the page are copied to another page (either in DRAM or PCM) for achieving PCM wear-leveling.

Fang et al. [25] propose a technique called "SoftPCM" which utilizes the error tolerance characteristic of video applications to relax the accuracy of write operations. It is well-known that several multimedia applications have the inherent ability of error tolerance [76–81]. Thus, a slight error in multimedia data may not be perceived by human end-users. Their technique leverages this fact and provisions that if the stored old data in PCM are very close to the new data to be written, the write operation is cancelled and the old data are taken as the new data. This leads to significant reduction in write-traffic which also reduces the energy consumption of PCM.

Qureshi et al. [82] propose a technique to alleviate the problem of slow writes. Their technique works on the observation that PCM writes are slow only in one direction (SET operation) and are almost as fast as reads in the other direction (RESET operation). Thus, a write operation to a line in which all memory cells have been SET before the write will consume much less time. Based on this, their technique pro-actively SETs all the bits in a given memory line much before the anticipated write to that memory line. As soon as a line becomes dirty in the cache, their technique initiates a SET request for that line, which allows a large window of time for the SET operation to complete.

As the demand for data increases [83–85], the size of main memory required will also increase and this would require effective management of main memory. Zhou et al. [11] propose a technique which works by removing redundant bit to reduce the unnecessary bit writes to PCM. Their technique performs a read before write, and writes only those bits to PCM which have changed.

Ramos et al. [33] propose a PCM-DRAM hybrid design for improving energy efficiency of main memory. Their technique uses a hardware-driven page placement policy. Their policy leverages the memory controller to monitor program access patterns and uses this information to migrate pages between DRAM and PCM, and translate the memory addresses coming from the cores. Further, the operating system periodically updates its page mappings based on the translation information used by the memory controller. Since most frequently accessed pages reside in DRAM, the high write latency of accessing PCM is avoided.

Qureshi et al. [86] propose a write cancellation and write pausing technique to indirectly improve the PCM read performance. Although a higher value of write latency can be tolerated using buffers and large write bandwidth, once a write request is scheduled for service to a PCM bank, a subsequent read access to the same bank needs to wait until the write access has completed. Thus, the slow write can increase the effective latency of read accesses and since read accesses are latency-critical, this may severely affect the program performance. Write cancellation policy aborts an on-going write if a read request arrives to the same bank and the write operation is not close to completion. It avoids aborting an ongoing write that has completed more than a threshold percentage of its service time and this threshold can be adapted during runtime.

A limitation of write-cancellation technique is that it requires some writes to be re-executed which incurs power and bandwidth overhead. To avoid this overhead, Qureshi et al. [86] propose write pausing technique. This technique utilizes fundamental characteristic of PCM that most multi-bit PCM devices use iterative write algorithms. In each iteration data are written and the current state of the device is compared with the desired state. Write pausing allows iterative write algorithms to potentially pause a write request at the end of each write iteration, complete a pending read request, and then resume the paused write request.

Tian et al. [41] present a task-scheduling based technique for addressing the challenges of hybrid DRAM-PCM main memory. They study the problem of task-scheduling, assuming that a task should be entirely placed in either PCM bank or DRAM bank. Their approach works for different optimization objectives such as 1. minimizing the energy consumption of hybrid memory for a given PCM and DRAM size and given PCM endurance 2. minimizing the number of writes to PCM for a given PCM and DRAM size and given threshold on energy consumption and 3. minimizing PCM size for a given DRAM size, given threshold on energy consumption and PCM endurance.

In context of PCM-DRAM hybrid main memory, Meza et al. [48] propose a technique for efficiently managing the metadata (such as tag, LRU, valid, and dirty bits) for data in a DRAM cache at a fine granularity. Their technique uses the observation that storing metadata off-chip in the same row as their data can exploit DRAM row buffer locality; also it reduces the access latency from two row buffer conflicts (one for the metadata and another for the datum itself). Based on this, their technique only caches the metadata for recently accessed rows on-chip using a small buffer. Since metadata needed for data with temporal or spatial locality is cached on-chip, it can be accessed with the same latency as an SRAM tag store. This provides better energy efficiency than using a large SRAM tag store.

Yoon et al. [50] propose row-buffer locality aware caching policies for hybrid PCM-DRAM main memories. Their technique works on the observation that both DRAM and PCM have row buffers, with (nearly) same latency and bandwidth. However, the cost of row buffer misses in terms of latency, bandwidth, and energy is much higher in PCM than in DRAM. Based on this, their technique avoids allocating in PCM data that frequently causes row buffer misses. Such data are allocated (cached) in DRAM, whereas the data that frequently hits in the row-buffer are stored in PCM. Further, since PCM has much higher write latency/power than read latency/power, their technique uses a caching policy such that the pages that are written frequently are more likely to stay in DRAM.

Huang et al. [87] propose a register-allocation based technique with re-computation to reduce the number of store instructions to non-volatile memory. Register allocation refers to multiplexing a large number of target program variables onto a small number of physical registers. The less the number of physical registers a processor contains, the more number of spills will be generated. Each spill is mapped to one store instruction and one (or few) load instructions during the compilation process. Traditional register allocation process does not distinguish read and write activities and does not try to minimize writes. Huang et al. use graph-coloring approach to extend traditional register allocation technique with re-computation to reduce non-volatile memory write activities by reducing store instructions. Their technique discards a set of carefully-selected actual spills and re-computes them when they are needed

Although PCM MLC devices offer more density than SLC (single level cell) devices, they also present significant challenges. For MLC devices to work properly, precise reading of reistance values is required. As the number of levels increase, the resistance region assigned to each data value decreases significantly. Thus, the read latency of MLC devices may increase linearly or exponentially with the number of bits. Qureshi et al. [88] present a memory architecture which aims to achieve the latency, lifetime and energy of SLC devices in the common case, while still achieving the high memory capacity of MLC device. Their technique divides the main memory in two regions, one with high-density, high-latency which uses MLC mode, and another with low-latency, low-density that uses half the number of bits per cell than high-density region. By tracking the memory requirements of the workloads, their technique adapts the fraction of both regions. When the workload requires high memory capacity, the system uses capacity benefits of MLC device. When the workload requirements can be satisfied with SLC (or fewer bits per level cell), the system increases the size of SLC region to avoid increased energy and latency. This is achieved by restricting the number of levels used in a MLC device to emulate a fewer bits per cell device. To avoid high latency for frequently accessed pages, the system transfers a page from high-density region to low-density (faster) region when the page is accessed.

Lee et al. [40] propose a memory management technique for hybrid PCM-DRAM memory to hide the slow write performance of PCM. Their technique uses methods such as dirty bit clearing and frequency accumulation to accurately estimate future write references. They observe that using write history alone performs better than using both read and write history in estimating future write references. Also, by using temporal locality and frequency characteristics, more accurate estimates of write references can be obtained. Based on these observations, they propose a page replacement algorithm called CLOCK-DWF (CLOCK with dirty bits and write frequency) that reduces the number of PCM write operations by using DRAM to absorb most of the write references.

Wang et al. [57] propose a technique for mitigating the energy overhead of MLC PCM devices. Their technique works on the observation that there are significant value-dependent energy variations in programming MLC PCM. Thus, by using data encoding, the write energy can be reduced. In a 2-bit PCM, there are four states, viz. 00, 01, 10 and 11. They show that programming states 00 and 11 require significantly less energy than programming the other states. Thus, data encoding is used to increase the 00 and 11 states in writing the data. They also use data comparison write (DCW) approach to enhance the effectiveness of the data encoding scheme.

Joshi et al. [60] present a circuit and microarchitecture-level technique to address the high write latency of MLC-PCM. The write latency and energy of PCM vary significantly with target resistance level and the initial state of PCM cell. Their technique adapts the programming scheme of MLC PCM by taking into consideration the initial state of the cell, the target resistance to be programmed and the effect of process variation on the programming current profile of the MLC. For states mapped at lower resistance values, they use single reset pulse programming and for states mapped at higher resistance values, they use staircase programming. Also, data comparison writes (DCW) is used to enhance the effect of their technique. Also, when the cell is already present in the stable completely set state, their technique skips initialization sequence for programming, which further improves the write latency and energy saving.

Xu et al. [61] propose data manipulation techniques to reduce the write energy of PCM main memory. Their technique works on the observation that PCM read incurs much less energy than PCM writes. Also, write of different value to a PCM cell incurs significantly different energy. Their technique uses selective-XOR operations to bias the data value distribution. For a given word to be written and originally stored word, their technique finds an optimal bit-pattern such that writing a XOR-masked value of word to be written with bit pattern leads to minimum write energy.

IV. Conclusion

Driven by the quest for exascale performance, modern processors, data-centers and supercomputers use large sized memory and limitations of conventional devices forces the designers to explore new avenues of design cache and memory hierarchy. The use of phase change memory (PCM) as a universal choice for main memory opens us several new challenges. It is clear that while PCM provides a promising alternative to conventional DRAM, it cannot completely replace DRAM due to its limitations. Thus, the computer architects and researchers need to design effective techniques at system, architecture and device level to address the shortcomings of PCM and utilize its strengths.

In this report, we have presented a review of techniques proposed for power management of phase change memory. We believe that our work will be useful for both beginners and experts in the field of PCM. Also, it will help them in gaining insights into the working of architectural techniques of PCM power management and encourage them to improve these techniques even further.

References

[1] A. Pande, E. Baik, and P. Mohapatra, "Efficient health data compression on mobile devices," in *3rd ACM MobiHoc workshop on Pervasive wireless healthcare*, 2013, pp. 25–30.

[2] K. Lee *et al.*, "Twitter trending topic classification," in *11th International Conference on Data Mining Workshops (ICDMW)*. IEEE, 2011, pp. 251–258.

[3] A. Pande, V. Ramamurthi, and P. Mohapatra, "Quality-oriented video delivery over lte," *Journal of Computing Science and Engineering*, 2011.

[4] C. Lefurgy, K. Rajamani, F. Rawson, W. Felter, M. Kistler, and T. W. Keller, "Energy management for commercial servers," *Computer*, vol. 36, no. 12, pp. 39–48, 2003.

[5] A. Hay, K. Strauss, T. Sherwood, G. H. Loh, and D. Burger, "Preventing PCM banks from seizing too much power," in *IEEE/ACM International Symposium on Microarchitecture*, 2011, pp. 186–195.

[6] M. K. Qureshi, S. Gurumurthi, and B. Rajendran, "Phase change memory: From devices to systems," *Synthesis Lectures on Computer Architecture*, vol. 6, no. 4, pp. 1–134, 2011.

[7] S. Raoux *et al.*, "Phase-change random access memory: A scalable technology," *IBM Journal of Research and Development*, vol. 52, no. 4.5, pp. 465–479, 2008.

[8] S. Cho and H. Lee, "Flip-n-write: a simple deterministic technique to improve pram write performance, energy and endurance," in *IEEE/ACM International Symposium on Microarchitecture*, 2009, pp. 347–357.

[9] G. Dhiman, R. Ayoub, and T. Rosing, "PDRAM: a hybrid PRAM and DRAM main memory system," in *46th ACM/IEEE Design Automation Conference (DAC)*, 2009, pp. 664–669.

[10] B.-D. Yang, J.-E. Lee, J.-S. Kim, J. Cho, S.-Y. Lee, and B.-G. Yu, "A low power phase-change random access memory using a data-comparison write scheme," in *IEEE International Symposium on Circuits and Systems (ISCAS)*, 2007, pp. 3014–3017.

[11] P. Zhou, B. Zhao, J. Yang, and Y. Zhang, "A durable and energy efficient main memory using phase change memory technology," in *ACM SIGARCH Computer Architecture News*, vol. 37, no. 3, 2009, pp. 14–23.

[12] S. Mittal, "A Survey of Architectural Techniques For DRAM Power Management," *International Journal of High Performance Systems Architecture*, vol. 4, no. 2, pp. 110–119, 2012.

[13] A. Pande and J. Zambreno, *Embedded Systems for Smart Appliances and Energy Management*. Springer, 2012.

[14] S. Mittal, "Dynamic cache reconfiguration based techniques for improving cache energy efficiency," Ph.D. dissertation, Iowa State University, 2013.

[15] S. Mittal and Z. Zhang, "EnCache: Improving cache energy efficiency using a software-controlled profiling cache," in *IEEE International Conference On Electro/Information Technology*, USA, May 2012.

[16] L. Barroso and U. Hölzle, "The datacenter as a computer: An introduction to the design of warehouse-scale machines," *Synthesis Lectures on Computer Architecture*, vol. 4, no. 1, pp. 1–108, 2009.

[17] B. Wang, B. Wu, D. Li, X. Shen, W. Yu, Y. Jiao, and J. S. Vetter, "Can PCM Benefit GPU? Reconciling Hybrid Memory Design with GPU Massive Parallelism for Energy Efficiency," College of William and Mary, Tech. Rep., 2013.

[18] D. Kim, S. Lee, J. Chung, D. H. Kim, D. H. Woo, S. Yoo, and S. Lee, "Hybrid DRAM/PRAM-based main memory for single-chip CPU/GPU," in *49th Annual Design Automation Conference*, 2012, pp. 888–896.

[19] Z. Shao, Y. Liu, Y. Chen, and T. Li, "Utilizing PCM for Energy Optimization in Embedded Systems," in *IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, 2012, pp. 398–403.

[20] J. Hu, C. J. Xue, Q. Zhuge, W.-C. Tseng, and E. H.-M. Sha, "Write activity reduction on non-volatile main memories for embedded chip multiprocessors," *ACM Trans. Embed. Comput. Syst.*, vol. 12, no. 3, pp. 77:1–

77:27, Apr. 2013.

[21] J. Hu, Q. Zhuge, C. J. Xue, W.-C. Tseng, and E. H.-M. Sha, "Software enabled wear-leveling for hybrid PCM main memory on embedded systems," in *Design, Automation Test in Europe Conference Exhibition (DATE)*, 2013, pp. 599–602.

[22] T. Wang, D. Liu, Z. Shao, and C. Yang, "Write-activity-aware page table management for PCM-based embedded systems," in *17th Asia and South Pacific Design Automation Conference (ASP-DAC)*, 2012, pp. 317–322.

[23] J. Hu, Q. Zhuge, C. J. Xue, W.-C. Tseng, and E. H. Sha, "Optimizing Data Allocation and Memory Configuration for Non-Volatile Memory Based Hybrid SPM on Embedded CMPs," in *IEEE 26th International Parallel and Distributed Processing Symposium Workshops & PhD Forum (IPDPSW)*, 2012, pp. 982–989.

[24] M. Zhou, S. Bock, A. P. Ferreira, B. Childers, R. Melhem, and D. Mossé, "Real-time scheduling for phase change main memory systems," in *IEEE 10th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, 2011, pp. 991–998.

[25] Y. Fang, H. Li, and X. Li, "SoftPCM: Enhancing Energy Efficiency and Lifetime of Phase Change Memory in Video Applications via Approximate Write," in *IEEE 21st Asian Test Symposium (ATS)*, 2012, pp. 131–136.

[26] S. Kwon, S. Yoo, S. Lee, and J. Park, "Optimizing video application design for phase-change RAM-based main memory," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 20, no. 11, pp. 2011–2019, 2012.

[27] P. Mangalagiri, K. Sarpatwari, A. Yanamandra, V. Narayanan, Y. Xie, M. J. Irwin, and O. A. Karim, "A low-power phase change memory based hybrid cache architecture," in *18th ACM Great Lakes symposium on VLSI*, 2008, pp. 395–398.

[28] Y. Joo, D. Niu, X. Dong, G. Sun, N. Chang, and Y. Xie, "Energy-and endurance-aware design of phase change memory caches," in *Conference on Design, Automation and Test in Europe*. European Design and Automation Association, 2010, pp. 136–141.

[29] J. Li, L. Shi, C. J. Xue, C. Yang, and Y. Xu, "Exploiting set-level write non-uniformity for energy-efficient NVM-based hybrid cache," in *9th IEEE Symposium on Embedded Systems for Real-Time Multimedia (ESTIMedia)*, 2011, pp. 19–28.

[30] J. Wang, X. Dong, Y. Xie, and N. P. Jouppi, "i2WAP: Improving non-volatile cache lifetime by reducing inter- and intra-set write variations," in *HPCA*, 2013, pp. 234–245.

[31] S. Guo, Z. Liu, D. Wang, H. Wang, and G. Li, "Wear-resistant hybrid cache architecture with phase change memory," in *IEEE 7th International Conference on Networking, Architecture and Storage (NAS)*, 2012, pp. 268–272.

[32] J. Tominaga, T. Kikukawa, M. Takahashi, and R. Phillips, "Structure of the optical phase change memory alloy, Ag–V–In–Sb–Te, determined by optical spectroscopy and electron diffraction," *Journal of applied physics*, vol. 82, no. 7, pp. 3214–3218, 1997.

[33] L. E. Ramos, E. Gorbatov, and R. Bianchini, "Page placement in hybrid memory systems," in *International conference on Supercomputing*, 2011, pp. 85–95.

[34] H. Park, S. Yoo, and S. Lee, "Power management of hybrid DRAM/PRAM-based main memory," in *48th Design Automation Conference*. ACM, 2011, pp. 59–64.

[35] W. Zhang and T. Li, "Exploring phase change memory and 3d die-stacking for power/thermal friendly, fast and durable memory architectures," in *18th International Conference on Parallel Architectures and Compilation Techniques (PACT)*. IEEE, 2009, pp. 101–112.

[36] M. K. Qureshi, V. Srinivasan, and J. A. Rivers, "Scalable high performance main memory system using phase-change memory technology," *ACM SIGARCH Computer Architecture News*, vol. 37, no. 3, pp. 24–33, 2009.

[37] T. Liu, Y. Zhao, C. J. Xue, and M. Li, "Power-aware variable partitioning for DSPs with hybrid PRAM and DRAM main memory," in *48th ACM/EDAC/IEEE Design Automation Conference (DAC)*, 2011, pp. 405–410.

[38] R. Bheda, J. Poovey, J. Beu, and T. Conte, "Energy efficient phase change memory based main memory for future high performance systems," in *International Green Computing Conference and Workshops (IGCC)*, 2011, pp. 1–8.

[39] D.-J. Shin, S. K. Park, S. M. Kim, and K. H. Park, "Adaptive page grouping for energy efficiency in hybrid PRAM-DRAM main memory," in *ACM Research in Applied Computation Symposium*, 2012, pp. 395–402.

[40] S. Lee, H. Bahn, and S. Noh, "Characterizing Memory Write References for Efficient Management of Hybrid

PCM and DRAM Memory," in *IEEE 19th International Symposium on Modeling, Analysis Simulation of Computer and Telecommunication Systems (MASCOTS)*, 2011, pp. 168–175.

[41] W. Tian, J. Li, Y. Zhao, C. J. Xue, M. Li, and E. Chen, "Optimal task allocation on non-volatile memory based hybrid main memory," in *ACM Symposium on Research in Applied Computation*, 2011, pp. 1–6.

[42] H. Seok, Y. Park, and K. H. Park, "Migration based page caching algorithm for a hybrid main memory of DRAM and PRAM," in *ACM Symposium on Applied Computing*. ACM, 2011, pp. 595–599.

[43] S. Baek, H. G. Lee, C. Nicopoulos, and J. Kim, "A dual-phase compression mechanism for hybrid dram/pcm main memory architectures," in *Proceedings of the Great lakes symposium on VLSI*. ACM, 2012, pp. 345–350.

[44] S. Kwon, D. Kim, Y. Kim, S. Yoo, and S. Lee, "A case study on the application of real phase-change RAM to main memory subsystem," in *Design, Automation and Test in Europe*, ser. DATE '12, 2012, pp. 264–267.

[45] T. J. Ham, B. K. Chelepalli, N. Xue, and B. C. Lee, "Disintegrated control for energy-efficient and heterogeneous memory systems," in *IEEE 19th International Symposium on High Performance Computer Architecture (HPCA)*, 2013, pp. 424–435.

[46] H. B. Sohail, B. Vamanan, and T. Vijaykumar, "MigrantStore: Leveraging Virtual Memory in DRAM-PCM Memory Architecture," Purdue University, Tech. Rep., 2012.

[47] X. Zhang, Q. Hu, D. Wang, C. Li, and H. Wang, "A read-write aware replacement policy for phase change memory," in *Advanced Parallel Processing Technologies*, ser. Lecture Notes in Computer Science, O. Temam, P.-C. Yew, and B. Zang, Eds., 2011, vol. 6965, pp. 31–45.

[48] J. Meza, J. Chang, H. Yoon, O. Mutlu, and P. Ranganathan, "Enabling efficient and scalable hybrid memories using fine-granularity dram cache management," *Computer Architecture Letters*, 2012.

[49] Y. Park, D.-J. Shin, S. K. Park, and K. H. Park, "Power-aware memory management for hybrid main memory," in *2nd International Conference on Next Generation Information Technology (ICNIT)*. IEEE, 2011, pp. 82–85.

[50] H. Yoon, J. Meza, R. Ausavarungnirun, R. A. Harding, and O. Mutlu, "Row buffer locality aware caching policies for hybrid memories," in *IEEE 30th International Conference on Computer Design (ICCD)*, 2012, pp. 337–344.

[51] H. G. Lee, S. Baek, C. Nicopoulos, and J. Kim, "An energy-and performance-aware DRAM cache architecture for hybrid DRAM/PCM main memory systems," in *IEEE 29th International Conference on Computer Design (ICCD)*, 2011, pp. 381–387.

[52] G. Wu, J. Gao, H. Zhang, and Y. Dong, "Improving pcm endurance with randomized address remapping in hybrid memory system," in *IEEE International Conference on Cluster Computing (CLUSTER)*, 2011, pp. 503–507.

[53] J.-W. Hsieh and Y.-H. Kuan, "Double Circular Caching Scheme for DRAM/PRAM Hybrid Cache," in *IEEE 18th International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA)*, 2012, pp. 469–472.

[54] M. V. Vamsikrishna, Z. Su, and K.-L. Tan, "A Write Efficient PCM-Aware Sort," in *Database and Expert Systems Applications*. Springer, 2012, pp. 86–100.

[55] L. Ramos and R. Bianchini, "Exploiting phase-change memory in cooperative caches," in *IEEE 24th International Symposium on Computer Architecture and High Performance Computing (SBAC-PAD)*, 2012, pp. 227–234.

[56] W. Hwang and K. H. Park, "HMMSched: Hybrid Main Memory-Aware Task Scheduling on Multicore Systems," in *FUTURE COMPUTING 2013, The Fifth International Conference on Future Computational Technologies and Applications*, 2013, pp. 39–48.

[57] J. Wang, X. Dong, G. Sun, D. Niu, and Y. Xie, "Energy-efficient multi-level cell phase-change memory system with data encoding," in *29th International Conference on Computer Design (ICCD)*. IEEE, 2011, pp. 175–182.

[58] J. Chen, R. C. Chiang, H. H. Huang, and G. Venkataramani, "Energy-aware writes to non-volatile main memory," *ACM SIGOPS Operating Systems Review*, vol. 45, no. 3, pp. 48–52, 2012.

[59] A. Mirhoseini, M. Potkonjak, and F. Koushanfar, "Coding-based energy minimization for phase change memory," in *49th Annual Design Automation Conference*, 2012, pp. 68–76.

[60] M. Joshi, W. Zhang, and T. Li, "Mercury: A fast and energy-efficient multi-level cell based phase change memory system," in *IEEE 17th International Symposium*

on High Performance Computer Architecture (HPCA), 2011, pp. 345–356.

[61] W. Xu, J. Liu, and T. Zhang, "Data manipulation techniques to reduce phase change memory write energy," in *14th ACM/IEEE international symposium on Low power electronics and design*, 2009, pp. 237–242.

[62] N. Barcelo, M. Zhou, D. Cole, M. Nugent, and K. Pruhs, "Energy efficient caching for phase-change memory," in *Design and Analysis of Algorithms*. Springer, 2012, pp. 67–81.

[63] A. P. Ferreira, M. Zhou, S. Bock, B. Childers, R. Melhem, and D. Mossé, "Increasing PCMmain memory lifetime," in *Design, Automation and Test in Europe*, 2010, pp. 914–919.

[64] Y. Du, M. Zhou, B. Childers, R. Melhem, and D. Mossé, "Delta-compressed caching for overcoming the write bandwidth limitation of hybrid main memory," *ACM Trans. Archit. Code Optim.*, vol. 9, no. 4, pp. 55:1–55:20, Jan. 2013.

[65] B. C. Lee, E. Ipek, O. Mutlu, and D. Burger, "Architecting phase change memory as a scalable DRAM alternative," *ACM SIGARCH Computer Architecture News*, vol. 37, no. 3, pp. 2–13, 2009.

[66] Q. Li, L. Jiang, Y. Zhang, Y. He, and C. J. Xue, "Compiler directed write-mode selection for high performance low power volatile PCM," in *14th ACM SIGPLAN/SIGBED conference on Languages, compilers and tools for embedded systems*, ser. LCTES '13, 2013, pp. 101–110.

[67] L. Jiang, Y. Zhang, B. R. Childers, and J. Yang, "FPB: Fine-grained power budgeting to improve write throughput of multi-level cell phase change memory," in *IEEE/ACM International Symposium on Microarchitecture*, 2012, pp. 1–12.

[68] X. Dong and Y. Xie, "AdaMS: Adaptive MLC/SLC phase-change memory design for file storage," in *16th Asia and South Pacific Design Automation Conference (ASP-DAC)*. IEEE, 2011, pp. 31–36.

[69] M. Poremba and Y. Xie, "NVMain: An Architectural-Level Main Memory Simulator for Emerging Non-volatile Memories," in *IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, 2012, pp. 392–397.

[70] X. Dong, C. Xu, Y. Xie, and N. P. Jouppi, "Nvsim: A circuit-level performance, energy, and area model for emerging nonvolatile memory," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 31, no. 7, pp. 994–1007, 2012.

[71] G. Zhu, K. Lu, and X. Li, "SCM-BSIM: A Non-Volatile Memory Simulator Based on BOCHS," in *Emerging Technologies for Information Systems, Computing, and Management*. Springer, 2013, pp. 977–984.

[72] X. Li, K. Lu, and X. Zhou, "SIM-PCM: A PCM Simulator Based on Simics," in *Fourth International Conference on Computational and Information Sciences (ICCIS)*. IEEE, 2012, pp. 1236–1239.

[73] X. Dong, N. P. Jouppi, and Y. Xie, "PCRAMsim: System-level performance, energy, and area modeling for phase-change RAM," in *International Conference on Computer-Aided Design*. ACM, 2009, pp. 269–275.

[74] S. Bock, B. Childers, R. Melhem, D. Mossé, and Y. Zhang, "Analyzing the impact of useless write-backs on the endurance and energy consumption of pcm main memory," in *IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*. IEEE, 2011, pp. 56–65.

[75] D. H. Yoon, T. Gonzalez, P. Ranganathan, and R. S. Schreiber, "Exploring latency-power tradeoffs in deep nonvolatile memory hierarchies," in *9th conference on Computing Frontiers*. ACM, 2012, pp. 95–102.

[76] A. Pande and J. Zambreno, *Embedded Multimedia Security Systems: Algorithms and Architectures*. Springer, 2013.

[77] A. Sood *et al.*, "A novel rate-scalable multimedia service for e-learning videos using content based wavelet compression," in *Annual IEEE India Conference*, 2006, pp. 1–6.

[78] A. Mittal *et al.*, "Content-based network resource allocation for real time remote laboratory applications," *Signal, Image and Video Processing*, vol. 4, no. 2, pp. 263–272, 2010.

[79] A. Pande, A. Verma, A. Mittal, and A. Agrawal, "Network resource allocation of e-learning videos for scalable video delivery using content-based compression," *International Journal of Signal and Imaging Systems Engineering*, vol. 2, no. 3, pp. 117–125, 2009.

[80] A. Pande, A. Verma, A. Mittal, and A. Agrawal, "Dynamic multimedia content allocation for scarce resource networks," *International Journal of Advanced Media and Communication*, vol. 3, no. 3, pp. 247–259, 2009.

[81] A. J. Chan *et al.*, "Temporal quality assessment for

mobile videos," in *18th annual international conference on Mobile computing and networking*. ACM, 2012, pp. 221–232.

[82] M. K. Qureshi, M. M. Franceschini, A. Jagmohan, and L. A. Lastras, "PreSET: improving performance of phase change memories by exploiting asymmetry in write times," in *39th Annual International Symposium on Computer Architecture (ISCA)*. IEEE, 2012, pp. 380–391.

[83] P. McDonagh *et al.*, "Toward Deployable Methods for Assessment of Quality for Scalable IPTV Services," *IEEE Transactions on Broadcasting*, 2013.

[84] S. Jana *et al.*, "Mobile video chat: issues and challenges," *IEEE Communications Magazine*, vol. 51, no. 6, 2013.

[85] A. Pande, V. Ahuja, R. Sivaraj, E. Baik, and P. Mohapatra, "Video delivery challenges and opportunities in 4g networks," *MultiMedia, IEEE*, vol. 20, no. 3, pp. 88–94, 2013.

[86] M. K. Qureshi, M. M. Franceschini, and L. A. Lastras-Montaño, "Improving read performance of phase change memories via write cancellation and write pausing," in *IEEE 16th International Symposium onHigh Performance Computer Architecture (HPCA)*, 2010, pp. 1–11.

[87] Y. Huang, T. Liu, and C. J. Xue, "Register allocation for write activity minimization on non-volatile main memory," in *16th Asia and South Pacific Design Automation Conference*. IEEE Press, 2011, pp. 129–134.

[88] M. K. Qureshi, M. M. Franceschini, L. A. Lastras-Montaño, and J. P. Karidis, "Morphable memory system: a robust architecture for exploiting multi-level phase change memories," in *ACM SIGARCH Computer Architecture News*, vol. 38, no. 3, 2010, pp. 153–162.