Using Non-Volatile Memory to Save Energy in Servers

David Roberts
University of Michigan
Department of CSE
Advanced Computer Architecture Lab
daverobe@umich.edu

Taeho Kgil
Intel Corporation
taeho.kgil@intel.com

Trevor Mudge
University of Michigan
Department of CSE
Advanced Computer Architecture Lab
tnm@eecs.umich.edu

Abstract—**Recent breakthroughs in circuit and process technology have enabled new usage models for non-volatile memory technologies such as Flash and phase change RAM (PCRAM) in the general purpose computing environment. These technologies display high density and low power consumption as well as persistency that are appealing properties in a memory device. This paper summarizes our earlier work on improving NAND Flash based disk caches and extends it to consider PCRAM. We first present the primary challenges in reliably managing non-volatile memories such as NAND Flash, reviewing our past work on architectural support for Flash manageability. We then provide a preliminary analysis of how our current Flash manageability architecture may be simplified when we replace Flash with PCRAM. Our evaluations on PCRAM shows a potential for more than a 65% throughput improvement for a disk-intensive database workload. Although more detailed studies are needed, we conclude that PCRAM is a strong contender to replace Flash if it becomes cost-effective.**

I. Introduction

Data centers are an integral part of today's computing platforms. As cloud computing initiatives provide IT capabilities that incorporates software as a service, it requires internet service providers such as Google and Yahoo to build large scale data centers hosting millions of servers. Energy efficiency becomes a critical aspect to address the increasing cost of operating a data center. Data centers based on off-the-shelf general purpose processors are unnecessarily power hungry, require expensive cooling systems and occupy a large space. In fact, the cost of power and cooling these data centers is starting to dominate the operating cost.

System memory power (DRAM power) and disk power contribute as much as 40% to the overall power consumption in a data center. Further, current trends suggest that this will continue to increase at a rapid rate as more DRAM and disk drives are integrated to improve throughput.

Fortunately, there are emerging memory devices in the technology pipeline that may address this concern. These devices typically display high density and consume low idle power. Flash, Phase Change RAM (PCRAM) and Magnetic RAM (MRAM) fall into this class.

In particular, Flash is an attractive technology that is already deployed heavily in various computing platforms. Today, NAND Flash can be found in hand-held devices such as smart phones, digital cameras and MP3 players. This has been made possible because of its high density, low power properties and non-volatility. Its popularity has meant that it is the focus of aggressive process scaling and innovation.

The rapid rate of improvement in density has become the primary driver to consider Flash for other usage models. There are several Flash usage models in the data center that are currently being examined by industry and academia to address rising power and cooling costs, among other things.

Recently, PCRAM has received much attention because of the challenges Flash faces when we scale below the 22nm process technology node. Studies [1], [2] have shown that PCRAM is expected to outperform Flash post 22nm and emerge as an important and widely used memory device.

This paper reviews the benefits of integrating high density non-volatile (NV) memory into a server. We specifically look at the benefits of Flash and PCRAM. This paper discusses the following:

1) We describe the challenges of integrating NAND Flash into a server and briefly describe how these challenges are addressed in the Flash management architecture published in earlier work [3], [4].
2) We quantitatively review the benefits of integrating NAND Flash for single level cell (SLC) and multiple level cell (MLC) NAND Flash.
3) We provide a preliminary analysis of how PCRAM could simplify the management architecture for Flash assuming PCRAM outperforms NAND Flash in scalability. We evaluate the performance of a scaled future PCRAM based disk cache.

The paper is organized as follows. The next section provides background on Flash and PCRAM. Section III reviews our Flash management architecture published in [4]. Section IV describes how our management architecture can be simplified when we replace Flash with PCRAM. Section V presents concluding remarks.

II. Background

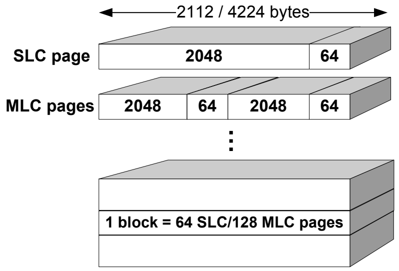
The ITRS 2007 roadmap (see Table I) gives the projected cell density and endurance characteristics of Flash and PCRAM. Flash remains the densest device, aided greatly by the use of MLC technology. For example, in 2009, the density of SLC Flash is predicted to be over 3x that of PCRAM

978-3-9810801-5-5/DATE09 © 2009 EDAA

	2009	2011	2013	2015	2017
NAND Flash-SLC*($\mu m^2/bit$)	0.0081	0.0052	0.0031	0.0021	0.0013
NAND Flash-MLC*($\mu m^2/bit$)	0.0041	0.0013	0.0008	0.0005	0.0003
PCRAM(nMOSFET)-SLC*($\mu m^2/bit$)	0.0254	0.0123	0.0069	0.0036	0.0024
PCRAM(nMOSFET)-MLC*($\mu m^2/bit$)	0.0127	0.0061	0.0017	0.0009	0.0006
DRAM Cell density($\mu m^2/bit$)	0.0153	0.0096	0.0061	0.0038	0.0024
Flash write/erase cycles	1E+05	1E+05	1E+05	1E+05	1E+04
PCRAM write/erase cycles	1E+10	1E+10	1E+12	1E+15	1E+15
Flash SLC/MLC data retention (years)	10-20	10-20	10-20	10-20	10-20
PCRAM SLC/MLC data retention (years)	>10	>10	>10	>10	>10

* SLC - Single level Cell, MLC - Multi Level Cell

TABLE I
ITRS 2007 ROADMAP FOR MEMORY TECHNOLOGY.



(a) Flash block diagram

Fig. 1. Example dual mode SLC/MLC Flash bank organization

and almost 2x that of DRAM. The endurance of PCRAM however is consistently a factor of 10^5 or more better than Flash. Its data retention time is also good. In terms of latency, however, PCRAM is far superior to Flash. Read latency is over 200x lower than SLC NAND Flash, and write latency is over 5x lower [2]. This suggests that future systems may employ more Flash than PCRAM, but use PCRAM as a higher level in the memory hierarchy (such as a cache) for Flash. That way, average latency can be reduced but at low cost. Evaluating a multi-level NV memory hierarchy of this type is beyond the scope of this paper and will be addressed in future work. As a preliminary study, however, we evaluate the effects of completely replacing NAND Flash with PCRAM, and anticipate the performance of a multi-level hierarchy will lie between these best and worst-case data points.

A. Properties of a NAND Flash device

Flash memory is a non-volatile memory device that can be electrically read, written and erased. Flash memory cells in NAND Flash are connected in series to maximize cell density. Further, to improve Flash density, each Flash memory cell can use multiple threshold voltage levels to store more than one bit per cell, called multi-level cells (MLC). NAND Flash using a single threshold voltage level (technically two levels) is called SLC. Cutting edge MLC NAND Flash supports 4 bits per cell. MLC Flash is cheaper and denser relative to SLC, but MLC is slower to read and write and has shorter lifetime by a factor of 10 or more. Typical latencies for read, write and erase are 25 μs, 250 μs and 0.5 ms for SLC and 50 μs, 900 μs and 3.5 ms for 2-bit MLC.

NAND Flash is organized in units of *pages* and *blocks*. A typical Flash *page* is 2KB in size and a Flash *block* is made up of 64 Flash *pages* (128KB). Random Flash reads and writes are performed on a *page* basis and Flash erasures are performed per *block*. A Flash must perform an erase on a *block* before it can write to a *page* belonging to that *block*. Each additional write must be preceded by an erase. Therefore *out-of-place* writes are commonly used to mitigate wear out. These writes append new data to the end of the log while old data pages are invalidated.

NAND Flash can also be dynamically configured to support multiple Flash memory cell types for each page or block. In fact, such devices are now commercially available, e.g., Samsung's Flex-OneNAND [5]. Figure 1(a) illustrates the organization of an SLC/MLC dual mode device. Pages in SLC mode consist of 2048 bytes of data area and 64 bytes of 'spare' data for error correction code (ECC) bits. When in MLC mode, a single SLC page can be split into two 2048 byte MLC pages. Pages are erased together in blocks of 64 SLC pages or 128 MLC pages.

Multi-level cell Flash ages quicker than single level cell Flash. An MLC Flash can support fewer reliable write/erase cycles due to the smaller threshold voltage margin between bit values. New Flash architectures [5] can circumvent this problem by switching from high-density MLC to lower density or even single-level mode to counter wear-out. No policy currently exists to perform the mode selection, so we have proposed a mechanism for changing mode, tailored to a disk caching application.

Because Flash blocks have a limited number of erases before they develop faulty bits, *wear-leveling* algorithms are employed to equalize the number of erases performed on each block [6], [7]. This has to be achieved without performing more erases than necessary. The simplest method of wear-leveling is to treat the device as a circular log. New data is written to the next available page and the old page is invalidated. However, wear-leveling causes fragmentation problems. Fragmentation is addressed with garbage collection. The process of garbage collection reads valid pages from erase blocks containing some invalid pages, then writes them to a previously erased block [8]. Garbage collections frees up pages that are ready to write new data. This process takes time and increases the amount of wear in the Flash blocks.

B. Properties of a Phase Change Memory (PCRAM) device

Phase Change memory has recently emerged as a potential candidate for non-volatile storage. Several companies (e.g. Samsung and Numonyx) have prototype devices of up to 512 Mbit capacity. Rather than storing electrons to represent data, phase change memory records values as the physical state of the storage material (typically Chalcogenide glass). In crystalline and amorphous states it has low and high resistivity, respectively. High current causes the material to freeze to an amorphous state in less than 100ns [9]. A medium current for a longer time re-crystallizes the material into a crystalline state. To read from a bit cell, a much lower current is used to determine the resistance and thus the bit value.

The characteristics of PCRAM are better than Flash in terms of latency and durability (around 10^{10} erase cycles instead of 10^5 for Flash). PCRAM does not have to be erased before writing, unlike Flash. This reduces the need for our proposals for Flash lifetime improvement schemes such as read/write partitioned caches. However, our scheme for dynamic MLC/SLC mode switching and ECC strength control can still help to minimize latency. This paper uses access latencies from an MLC-only variant of PCRAM [2]. The read latency is conservative, but the write latency assumes that there is enough bandwidth to store a 2 KB page of data during the write interval for this prototype PCRAM. It is expected that as the cell technology and memory array architecture matures, the available write bandwidth will be able to accommodate this demand.

III. Flash Management Architecture

The right side of Figure 2 shows the Flash based disk cache architecture proposed in [3], [4]. Compared to a conventional DRAM-only architecture shown on the left side of Figure 2, our architecture uses a two level disk cache, composed of a relatively small DRAM in front of a dense Flash. The much lower access time of DRAM allows it to act as a cache for the Flash without significantly increasing power consumption. A Flash memory controller is also required, for reliability management.

Our design uses a NAND Flash that stores 2 bits per cell (MLC) and is capable of switching from MLC to SLC mode using techniques proposed in [10], [5]. Finally, our design uses variable-strength error correction code (ECC) to improve reliability while adding the smallest possible delay.

Our architecture requires additional data structures to manage the Flash blocks and pages. These tables are read from the hard disk drive and stored in DRAM at run-time to reduce access latency and mitigate wear out. Together, they describe whether pages exist in DRAM or Flash, and specify the various Flash memory configuration options for reliability.

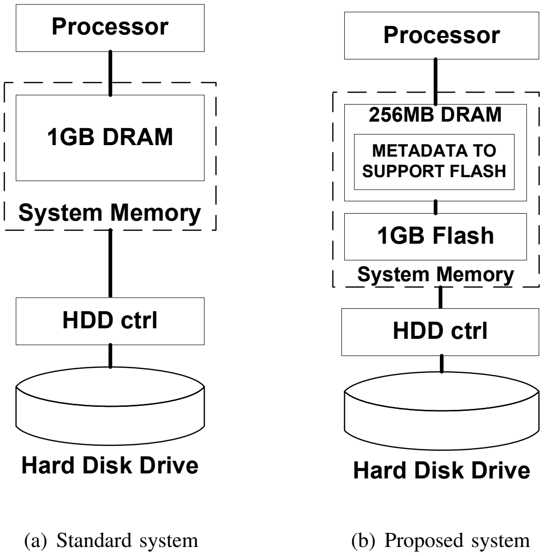


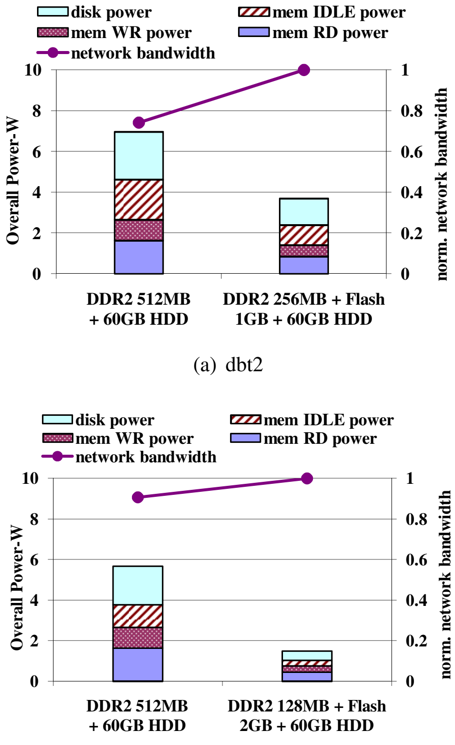
Fig. 2. 1GB DRAM is replaced with a smaller 256MB DRAM and 1GB NAND-based Flash. Additional components are added to control Flash.

We divide the Flash into a read disk cache and a write disk cache. Read caches are less susceptible to *out-of-place* writes, which reduce the read cache capacity and increase the risk of garbage collection. An *out-of-place* write happens when existing data is modified, because Flash has to be erased before it can be written to a second time. It is simple to invalidate the old data page then write new data into a previously erased page. However, the invalid pages accumulate as wasted space and must be garbage collected later. By splitting Flash into read and write regions, we were able cut down on time consuming garbage collections. PCRAM does not suffer from this limitation because erases are not required for new data to be written. Therefore read/write splitting only applies to a Flash memory based disk cache.

Flash needs architectural support to improve reliability and lifetime when used as a cache. We address this need with a programmable Flash memory controller providing error correction and cell density (MLC/SLC) selection features.

Page cache misses from the operating system provide the address being accessed and any data to be written. In addition, the OS specifies the strength of error correction code (ECC) and whether the page is in MLC or SLC mode. The controller returns any data that was read along with information concerning the number of errors currently being corrected by the ECC logic.

The architecture uses a BCH (Bose, Chaudhury, Hocquenghem) encoder and decoder to perform error correction and a CRC checker to perform error detection. The BCH code guarantees that several faulty bits can be corrected. However,



(a) dbt2

(b) SPECWeb99

Fig. 3. Breakdown in system memory and disk power and network bandwidth for architecture with/without a Flash based disk cache.

as the number of faulty bits increases it takes longer to perform the correction. Doubling the number of correctable bits approximately doubles the time needed to decode the data and extract the correct value. Our system adapts the ECC strength to the appropriate number of faulty bits in each page to achieve graceful Flash wear out.

The programmable Flash memory controller also dynamically controls the density of a Flash page. Density control benefits Flash performance and endurance, because we are able to reduce access latency for frequently accessed pages and possibly improve endurance for aging Flash pages by changing MLC pages into SLC pages as needed.

A. Energy Efficiency of Flash management architecture

We evaluated the Flash memory controller and Flash device using a full system simulator called M5 [11]. The M5 simulation infrastructure is used to generate access profiles for estimating system memory and disk drive power consumption along with published access energy data. Given the limitations in our simulation infrastructure, a server workload that uses a large working set of 100's∼1000's of gigabytes cannot easily be evaluated. We scaled our benchmarks, system memory size, Flash size and disk drive size accordingly to run on our simulation infrastructure.

Figure 3 shows a breakdown of power consumption in the system memory and disk drive (left y-axis). Figure 3 also shows the measured network bandwidth (right y-axis). Throughput measured as network bandwidth is a good indicator of overall system performance as it represents the amount of data that the server can handle in each configuration. We calculated power for a DRAM-only system memory and a heterogenous (DRAM+Flash) system memory that uses a Flash as a secondary disk cache with hard disk drive support. We assume equal die area for a DRAM-only system memory and a DRAM+Flash system memory. Figure 3 shows the reduction in disk drive power and system memory power that results from adopting Flash. The primary power savings for system memory come from using Flash instead of DRAM for a large amount of the disk cache. The power savings for disk come from reducing the accesses to disk due to a bigger overall disk cache made possible by using Flash. We also see improved throughput with Flash because of lower access latency than disk.

IV. PCRAM Management Architecture

This section describes a preliminary proposal for software and hardware changes supporting PCRAM as a secondary disk cache instead of NAND Flash.

The software support for PCRAM is slightly modified relative to the Flash disk cache. First, there is no need to do out-of-place writes because old data can simply be overwritten without erasing other pages. This also eliminates the need for garbage collection because there will be no invalid pages to erase. Furthermore, we do not implement the read/write splitting feature because it is no longer necessary. Wear-leveling is still necessary to maximize chip lifetime, evenly spreading write operations across all pages to prevent premature page wear-out. It will take many more erase cycles than Flash for wear-out to exhibit errors. A further possible improvement is to use the PCRAM to store some of the system meta-data, although the analysis is beyond the scope of this paper. Similar to the usage model in prior work [12], the PCRAM's fast random access and in-place updates make it possible to track metadata in non-volatile memory. This was not practical in Flash due to its high latency and frequent out-of-place writes to metadata, causing fragmentation and increasing garbage collection overheads. The benefit of this is to reduce the footprint of in-DRAM metadata, saving energy. This may be important for the larger data structures like the Flash Cache Hash Table (FCHT) [3] used to translate disk addresses to Flash addresses. The Page Status table, containing ECC strength, density mode and number of writes to each page can also be stored in PCRAM. Figure 4 shows the metadata stored in PCRAM along with page data. There may be a need for hardware to perform a fast look-up of the cached page's address from the hash table. When a read or write is requested, the driver is notified if the page is invalid (not stored in the cache). Additional commands are needed to manipulate the metadata. These include changing the ECC strength or density mode of a page and marking pages as invalid (removing them from the cache).

The PCRAM hardware controller architecture is similar to the Flash controller and a possible design is shown in Figure

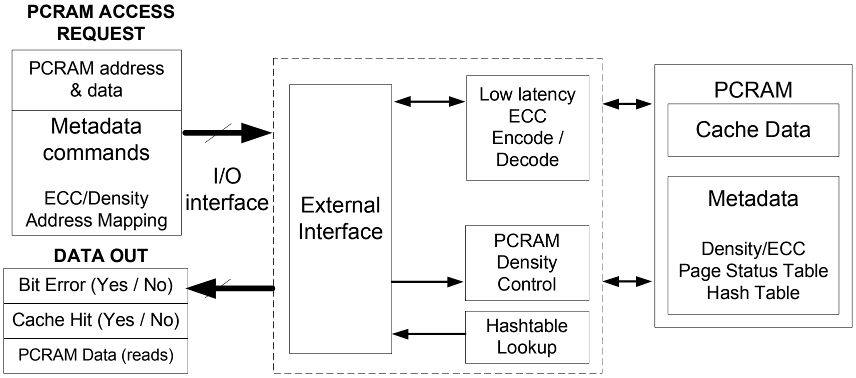


Fig. 4. PCRAM memory controller architecture. The disk cache device driver sends read and write requests to the hardware interface. The controller includes logic to perform a page look-up from metadata stored in PCRAM. The driver is informed of a hit or miss. The driver is also able to invalidate (evict) pages from the cache. In turn, the controller accesses the Flash chip after performing low latency ECC encoding for a write, or decoding for a read. The device driver software receives any requested data along with an indication of the number of failing bits.

4. It supports error correcting codes as well as MLC/SLC mode switching. Because of the higher endurance of PCRAM, MLC/SLC mode switches will initially be triggered only by workload changes to maximize cache hit rate, rather than failing bits. Later on when cells being to wear out, heuristics will decide if increasing ECC strength or switching to MLC mode will provide the best performance. Reference [4] provides details of how density modes can be adaptively selected based on the workload. To take full advantage of the lower latency afforded by PCRAM, the controller's ECC algorithms must operate with extremely low latency and support higher bandwidth. The number of correctable bits need not be as high as for Flash because the higher endurance of PCRAM means a lower error probability for the same number of write cycles (see Section II-B). In earlier work [4] we determined that decoding a 2 KB BCH coded Flash block correcting 2 errors takes around 30 μs using an ASIC. Decreasing this latency implies a larger, more power-hungry ECC component. To approach the raw read latency of the PCRAM device, simpler ECC schemes may have to be used such as parity and redundancy to avoid the overhead of a complex error-correction circuit.

A. Improvements due to PCRAM

We also evaluated our PCRAM architecture using M5 [11]. Our intention was to see the effects of PCRAM's lower latency and removing the need to erase blocks of pages. We applied similar setups to those used for our Flash studies. It should be noted that these preliminary simulations were again scaled down relative to the capacity of real servers because of resource limitations on the host system. However, the simulations are sufficient to highlight the trade-offs between the different memory technologies.

Figure 5 compares the relative network bandwidth achieved in servers using Flash or PCRAM as a secondary disk cache. We examined dbt2, the most disk intensive benchmark. This is not an in-memory database so there are significant numbers of file accesses at run-time to stress the storage subsystem. Doubling the DRAM capacity (which includes the primary page cache) increases performance by around 35% when combined with an SLC Flash secondary cache. At any particular size of main memory, MLC Flash performs slightly worse than SLC Flash as expected, and PCRAM performs up to 65% better than SLC Flash. The increased performance due to PCRAM also translates to a very significant total energy saving. Assuming that the server is in a low power state during idle periods, completing the work faster means that less energy is consumed by the power supply, processors, memory and disk drives in the system.

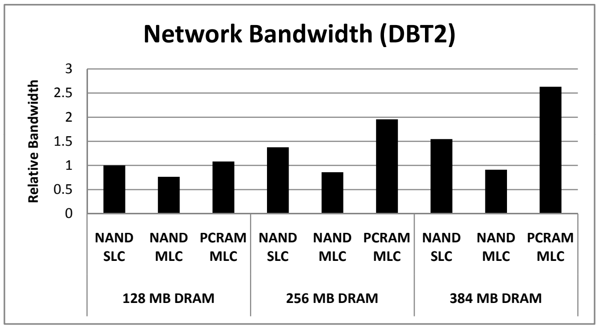


Fig. 5. Network bandwidth as a function of DRAM size (including primary disk cache) and secondary disk cache technology. The system was provisioned with 1GB of secondary disk cache.

It could be argued from this data that simply increasing the amount of system DRAM is a way to increase performance. This would be true if cost, density and power consumption were not constraints. In a server requiring multiple Gigabytes of main memory, the much lower cost per Gigabyte and greater density of NAND Flash (presently almost 4x) mean that it is cost effective and a lower power solution to construct main memory with more Flash memory than DRAM as our studies show. Furthermore, if PCRAM continues to scale as predicted, it will then become a candidate to replace or supplement Flash in our disk cache usage model.

V. Conclusion

This paper presents the challenges and opportunities of integrating Flash and PCRAM into a server platform. First, we reviewed an earlier proposal to manage Flash as a secondary disk cache with adaptive performance and lifetime enhancing schemes. We then proposed changes to this management architecture that would aid in replacing Flash with PCRAM. We observe significant improvement in performance, primarily due to the much lower access latencies of PCRAM. Furthermore, PCRAM has the potential to provide energy savings.

We believe these new memory technologies will force architects to rethink the current system memory and storage hierarchy in a server to help realize more efficient data centers.

References

[1] G. W. Burr, B. N. Kurdi, J. C. Scott, C. H. Lam, K. Gopalakrishnan, and R. S. Shenoy, “Overview of candidate device technologies for storage-class memory,” *IBM Journal of Research and Development*, vol. 52, no. 4, Sep 2008.

[2] F. Bedeschi, R. Fackenthal, C. Resta, E. M. Donz, M. Jagasivamani, E. C. Buda, F. Pellizzer, D. W. Chow, A. Cabrini, G. M. A. Calvi, R. Faravelli, A. Fantini, G. Torelli, D. Mills, R. Gastaldi, and G. Casagrande, “A bipolar-selected phase change memory featuring multi-level cell storage,” *JSSC*, vol. 44, 2009.

[3] T. Kgil and T. Mudge, “Flashcache: a nand flash memory file cache for low power web servers,” *International Conference on Compilers, Architecture and Synthesis for Embedded Systems*, 2006.

[4] T. Kgil, D. Roberts, and T. Mudge, “Improving NAND Flash based Disk Caches,” in *Proc. Int’l Symp. on Computer Architecture (ISCA)*, 2008.

[5] “Flex-OneNAND,” http://www.samsung.com/global/business/semiconductor/products/fusionmemory/Products_FlexOneNAND.html.

[6] “Technical Note: TrueFFS Wear-Leveling Mechanism(TN-DOC-017),” http://www.embeddedfreebsd.org/Documents/TrueFFS_Wear_Leveling_Mechanism.pdf.

[7] L.-P. Chang, “On efficient wear-leveling for large-scale flash-memory storage systems,” in *22nd ACM Symposium on Applied Computing (ACM SAC)*, 2007.

[8] L.-P. Chang and T.-W. Kuo, “Real-time garbage collection for flash-memory storage system in embedded systems,” in *ACM Transactions on Embedded Computing Systems, Vol 3, No. 4*, 2004.

[9] G. Atwood and R. Bez, “Current status of chalcogenide phase change memory,” *Device Research Conference (DRC)*, 2005.

[10] T. Cho et al., “A dual-mode NAND flash memory: 1-Gb multilevel and high-performance 512-mb single-level modes,” *IEEE Journal of Solid State Circuits*, vol. 36, no. 11, Nov 2001.

[11] N. Binkert, R. Dreslinski, L. Hsu, K. Lim, A. Saidi, and S. Reinhardt, “The M5 simulator: Modeling networked systems,” *IEEE Micro*, vol. 26, no. 4, pp. 52–60, Jul/Aug 2006.

[12] Y. Park, S.-H. Lim, C. Lee, and K. H. Park, “Pffs: A scalable flash memory file system for the hybrid architecture of phase-change ram and nand flash,” *SAC*, Mar. 2008.