Research Article

Performance Evaluation of Page Migration Scheme for NVRAM-Based Wireless Sensor Nodes

Yeonseung Ryu

Department of Computer Engineering, Myongji University, Nam-dong, Yongin, Gyeonggi-do 449-728, Republic of Korea

Correspondence should be addressed to Yeonseung Ryu; ysryu@mju.ac.kr

Received 2 October 2013; Accepted 4 November 2013

Academic Editor: James J. Park

Copyright © 2013 Yeonseung Ryu. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A wireless sensor network consists of low-powered and multifunctional sensor nodes. Since each sensor node is operated by a battery, the energy management has become one of the critical design challenges in wireless sensor networks. Some recent studies have shown that DRAM-based main memory spends a significant portion of the total system power. In this paper, we studied a buffer management scheme for hybrid main memory that combines low-power nonvolatile RAM (NVRAM) and DRAM in order to reduce the energy consumption in a sensor node. Though NVRAMs consume less power than volatile memories, they have common problems in write performance. The proposed scheme employs the page migration technique in order to reduce the write operations on NVRAM part of hybrid main memory. We have performed simulation studies and showed that the proposed page migration scheme outperforms the legacy buffer management schemes in terms of the number of write operations on NVRAM.

1. Introduction

Wireless sensor networks (WSNs) are composed of a number of sensor nodes and are able to perform complex tasks such as monitoring a region to obtain data about the environment and sending data to a central repository station. Today's sensor nodes are full-fledged computer systems, with a processor, main memory, storage, operating system, and a suite of sensors [1–4]. Sensor nodes collect not only sensed data from the environment, but also a stream of mass media data like videos and images. In order to process massive data volumes, sensor nodes are expected to require much more memory than legacy sensor nodes. Each sensor node is operated by a battery, and usually it is not feasible to replace or recharge this battery after deployment. The lifetime of a sensor network is considered over as soon as the battery power of the nodes is completely depleted. Therefore, the energy management of sensor nodes has become one of the key challenges in designing WSNs.

There have been a lot of studies to minimize the energy consumption for WSNs [5, 6]. While most previous studies have dealt with low power communication [7–10], this work focuses on reducing the energy consumption in the main memory in sensor nodes. Sensor nodes use DRAM for main memory as legacy computer systems do. However, recent studies have shown that DRAM-based main memory spends a significant portion of the total system power [11]. Since the capacitors used in DRAM lose their charge over time, DRAM must refresh all the cells approximately 20 times a second, reading each one and rewriting its contents. Such endless refresh operations consume nontrivial amount of power and become a contributing factor for power depletion of sensor node.

The recent advance of memory technology has ushered in new nonvolatile RAM (NVRAM) designs such as PRAM (phase change RAM), STT-MRAM (spin-torque transfer magnetic RAM), and FeRAM (ferroelectric RAM) that overcome the drawbacks of existing volatile memories such as SRAM or DRAM [12–15]. Among the NVRAMs, PRAM and STT-MRAM are becoming promising candidates for main memory because of their high density, comparable read access speed, and low power consumption. Unfortunately, cost per byte of new NVRAMs is extremely higher than that of DRAM until now. As a result, hybrid main memory using DRAM and NVRAM seems to be practicable instead of pure NVRAM-based main memory in the near future. Some recent studies have introduced NVRAM-based main memory organization as follows: PRAM-based main memory organization [16, 17], DRAM/PRAM hybrid main memory organization [18–22], and STT-MRAM-based...
memory organization [23, 24]. Also, there have been some buffer management schemes for PRAM-based main memory and DRAM/PRAM hybrid main memory [20, 25, 26]. From this research trend, it is highly expected that NVRAM-based hybrid main memory will be used in the next generation sensor nodes soon.

In this paper, we study an NVRAM-aware buffer management scheme for wireless sensor nodes which use DRAM/NVRAM hybrid main memory. Figure 1 illustrates the system configuration considered in this paper. Though NVRAM has attractive features, the write performance (access latency and energy consumption) of NVRAM is not comparable to that of DRAM. The goal of proposed buffer management scheme is to reduce the number of write operations on NVRAM. To do so, the proposed buffer management scheme performs page migration which moves the data from NVRAM to DRAM when the data needs to be written on NVRAM. Furthermore, proposed scheme deallocates clean DRAM buffers eagerly (greedy deallocation) in order to secure free DRAM buffers and thus minimize the number of write operations on NVRAM. We show, through trace-driven simulation, that the proposed scheme outperforms other legacy buffer management schemes in terms of the buffer hit ratio and the number of writes on NVRAM.

The rest of this paper is organized as follows. In Section 2, we describe the characteristics of nonvolatile memories such as PRAM, STT-MRAM, and NAND flash memory. Also, we introduce major buffer management schemes which are based on nonvolatile memory. In Section 3, we propose a buffer management scheme called NVRAM-aware buffer (NAB) scheme. It is followed by the description of page migration and greedy deallocation techniques in detail. Section 4 presents the performance evaluation results. Finally, Section 5 concludes the paper.

2. Related Works

2.1. Nonvolatile Memories. Among the NVRAMs, PRAM and STT-MRAM are becoming promising candidates for main memory because of their high density, comparable read access speed, and low power consumption. Table I shows the comparison of PRAM, STT-MRAM, and DRAM.

A PRAM cell uses a special material, called phase change material, to represent a bit [12, 27]. The phase change material can exist in two different but stable structural states: amorphous and crystalline, each of which has drastically different resistivity which can be used to represent logic 0 or 1. PRAM density is expected to be much greater than that of DRAM (about four times). Further, PRAM has negligible leakage energy regardless of the size of the memory. While its read performance (latency and energy) is comparable to that of DRAM, its write performance is worse than that of DRAM. Also, PRAM has a worn-out problem caused by limited write endurance (i.e., $10^8$). Since the write operations on PRAM significantly affect the performance of system, it should be carefully handled.

STT-MRAM is a next generation memory technology that takes advantage of magnetoresistance for storing data [13–15]. It uses a magnetic tunnel junction (MTJ), the fundamental building block, as a binary storage. An MTJ comprises a three-layered stack: two ferromagnetic layers and an MgO tunnel barrier in the middle (see Figure 2). Among them, the fixed layer located at the bottom has a static magnetic spin, the spin of the electrons in the free layer at the top is influenced by applying adequate current through the fixed layer to polarize the current, and the current is passed to the free layer. Depending on the current, the spin polarity of the free layer changes either parallel or antiparallel to that of the fixed layer. The parallel indicates a zero state, and the antiparallel a one state.

One of the biggest weaknesses of STT-MRAM is long write latency compared to DRAM. Since the fast access time of memories on a chip must be guaranteed and cannot be negotiable, the slow write operations of STT-MRAM limit its popularity, even though it shows competitive read performance. Another serious drawback of STT-MRAM is high power consumption in write operations.

Flash memory is a sort of nonvolatile memory which has been widely used in storage devices [28, 29]. Unlike PRAM and STT-MRAM, flash memory is a kind of electrically erasable programmable ROM (EEPROM). A flash memory consists of multiple blocks, and each block is composed of multiple pages. A block is the smallest unit of an erase operation, whereas the smallest unit for the read and write operation is a page. Erase operations are
Table 1: Comparison of memories.

|                | PRAM | STT-MRAM | DRAM |
|----------------|------|----------|------|
| Volatility     | No   | No       | Yes  |
| Cost/TB        | High | High     | Low  |
| Read latency   | 50~100 ns | 30 ns   | 15~50 ns |
| Write latency  | 150 ns | 30~100 ns | 15~50 ns |
| Endurance      | $10^4$ for write | — | — |
| Idle power     | $\sim$0.05 W | $\sim$0.05 W | $\sim$1.3 W/GB |
| Read energy    | $\sim$0.1 nJ/b | $\sim$0.1 nJ/b | $\sim$0.1 nJ/b |
| Write energy   | $\sim$0.5 nJ/b | $\sim$0.5 nJ/b | $\sim$0.1 nJ/b |

significantly slower than the read/write operations. Further, write operations are slower than read operations. Existing data in flash memory cannot be written over; the memory has to be erased in advance in order to write new data. Erase operation degrades the system performance and consumes a considerable amount of power.

2.2. Buffer Management Schemes. There have been a lot of studies on buffer management schemes considering non-volatile memories [25, 26, 28–33]. In particular, a number of flash memory aware buffer management schemes have been studied over the past decade [28–33]. The goal of these schemes is to minimize the number of erase operations on flash memory. A page-level scheme called clean-first least recently used (CFLRU) was proposed by [32]. CFLRU maintains a page list by LRU order and divides the page list into two regions, namely, the working region and clean-first region. In order to reduce the write cost on flash memory, CFLRU first evicts clean pages in the clean-first region by the LRU order, and if there are no clean pages in the clean-first region, it evicts dirty pages by their LRU order. CFLRU can reduce the number of write and erase operations by delaying the flush of dirty pages in the page cache.

Also, a block-level buffer cache scheme called block padding LRU (BPLRU) was proposed, which considers the erase operations on flash memory [33]. BPLRU maintains an LRU list based on the flash memory block. Whenever a page in the buffer cache is referenced, all pages in the same block are moved to the MRU position. When buffer cache is full, BPLRU scheme evicts all the pages of a victim block but it simply selects the victim block at the LRU position. In addition, it writes a whole block into a log block by the in-place scheme using the page padding technique. In page padding procedure, BPLRU reads some pages that are not in the victim block and writes all pages in the block sequentially. The page padding may perform unnecessary reads and writes, but it is effective because it can change an expensive full merge to an efficient switch merge. In BPLRU, all log blocks can be merged by the switch merge technique which results in decreasing the number of erase operations.

For the DRAM/PRAM hybrid main memory, a multiple queue scheme (we call it 4Q scheme) was proposed [25]. 4Q maintains a page list by LRU order and evicts a page from the LRU position. In order to reduce the writes on PRAM, 4Q predicts the page access pattern and migrates pages to DRAM or PRAM according to the access pattern. 4Q dynamically moves the write-bound pages from PRAM to DRAM and moves the read-bound pages from DRAM to PRAM. For prediction of the access pattern, 4Q calculates the weighting values of each page at every request and maintains 4 types of monitoring queues (see Figure 3). 4Q shows good performance when the access pattern is highly skewed like financial workload. Though 4Q tries to reduce the number of write operations on PRAM, it does incur high run-time overhead.

3. NVRAM-Aware Buffer Management

We propose a buffer management scheme called NVRAM-aware buffer (NAB) for wireless sensor nodes which use DRAM/NVRAM hybrid main memory. Figure 1 illustrates the system configuration considered in this paper. The goal of the proposed scheme is to reduce the number of write operations on NVRAM.

3.1. Buffer Page Management. We assume that the main memory is divided into DRAM and NVRAM by a memory address. A portion of main memory is reserved for use as buffer. The buffer space is divided into a set of pages, each of which is a unit of buffer allocation/deallocation. The size of a page is fixed (i.e., 1 Kbytes).

The proposed NAB scheme maintains allocated pages as a page list by least recently used (LRU) order shown in Figure 4. The NAB defines a search region as a set of pages from the LRU position of the page list. When a new page needs to be allocated for storing data, the NAB allocates it from the free buffer pool and stores the data in it. And then, the NAB places it at the most recently used (MRU) position of the page list.
Whenever a page in the page list is accessed, it is moved to
the MRU position. When a page is deallocated, it is removed
from the page list and returned to the free buffer pool.

In Figure 4, there are 6 pages in the page list. The gray
pages are allocated from NVRAM and the white pages are
allocated from DRAM. The search region size is 3 pages.

3.2. Page Migration. In order to reduce the number of write
operations on NVRAM, when a clean (i.e., not modified)
page in the NVRAM is referenced by a write operation, the
NAB performs the page migration procedure as shown in
Algorithm 1.

First, the NAB tries to allocate a free DRAM page and
writes requested data to the allocated DRAM page. Then
it deallocates the original NVRAM page. In Figure 5, for
example, data D0 in a NVRAM page is accessed by write
request. Then the NAB allocates a free DRAM page and
migrates the new data D0 to the newly allocated DRAM page.
And then the NAB places it at MRU position of the page list.

If there is no free DRAM page in the free buffer pool, the
NAB tries to find a clean DRAM page from the search region
and uses it for storing the requested write data. If there is
no clean DRAM page in the search region, the NAB writes
requested data to the original NVRAM page.

3.3. Page Deallocation. The NAB proposes a greedy dealloca-
tion technique which frees clean DRAM pages even though
free buffers are still available in the system. Because there
could be a lot of used buffers that will not be accessed soon,
we can free them early with little influence on the cache
performance. To do so, the NAB searches clean DRAM pages
from the search region periodically or whenever the number
of free DRAM pages falls down below a threshold. Then,
it makes them free. This scheme results in decreasing the
number of writes on NVRAM because the NAB can secure
free DRAM pages for new page allocations.

If all free pages are used up, the NAB selects a victim page
from the search region in order to make free pages. In order
to reduce the number of write operations on flash memory,
the NAB tries to find a clean page. If there is no clean page
in the search region, the NAB just selects a page at the LRU
position of the page list. If necessary, the data in the victim
page is stored in the storage (flash memory).

4. Performance Evaluation

4.1. Experiment Setup. In order to evaluate the proposed
scheme, we have developed a trace-driven simulator. For
the workload, we obtained the virtual memory traces of an
application which is similar to database applications.

For example, TinyDB is a distributed query processor that
runs on each of the nodes in a sensor network [4]. TinyDB
runs on the Berkeley mote platform, on top of the TinyOS
operating system [3]. Using sensor data management func-
tionalities, users connect to the sensor network using a work-
station or base station directly connected to a sensor desig-
nated as the sink. Aggregate queries over the sensor data are
formulated using a simple SQL-like language and then dis-
tributed across the network. Aggregate results are sent back to
the workstation over a spanning tree, with each sensor com-
bining its own data with results received from its children.

We set the DRAM-to-NVRAM ratio as 1 : 1 and the search
region size as 1/3 of the page list. We evaluate the buffer hit
ratio and the write counts on NVRAM with varying the buffer
size until the buffer hit ratio becomes 100%.

4.2. Buffer Hit Ratio. Figure 6 shows the comparison of buffer
hit ratio with varying the buffer size. All schemes except
Q4 are similar. When the buffer size is small (i.e., 10), the
buffer hit ratio of Q4 is nearly 5% smaller than those of other
schemes.

However, as the buffer size increases, the buffer hit ratios
of all schemes increase rapidly and reach almost 100%.

4.3. Write Counts on NVRAM. Figure 7 shows the com-
parsion of write counts on NVRAM. As the buffer size
increases, the write counts on NVRAM decreases. Also, we
can see that NVRAM-aware schemes such as Q4 and the
NAB outperform other schemes. But, the write counts of NAB
decreases faster than that of Q4 as the buffer size increases.
Therefore, proposed NAB scheme outperforms all other
legacy schemes in terms of the write counts on NVRAM.

4.4. Effect of DRAM-to-NVRAM Ratio. Figure 8 shows the
write counts on NVRAM when the ratio of the size of DRAM
to the size of NVRAM varies. We set the total buffer size as
500 pages. It is normal that the write counts on NVRAM
increase as the ratio of NVRAM size increases. However, the
write counts of NAB are smaller than those of other schemes for all cases. Hence, the proposed NAB scheme outperforms legacy buffer schemes regardless of the DRAM-to-NVRAM ratio. The evaluation results of buffer hit ratios are not shown because they are similar to Figure 6.

5. Conclusion

The power of wireless sensor networks lies in the ability to deploy large number of sensor nodes that assemble and configure themselves. A sensor node is a battery-powered computer. If power is used naively, individual nodes will deplete their energy supplies in only a few days. In contrast, if sensor nodes are very spartan about power consumption, months or years of lifetime are possible. Hence, the energy management of sensor nodes has become one of the key design challenges in WSNs. While there are many studies about energy management for low power communication, there have been few researches on low power main memory using NVRAMs in sensor nodes.

In this paper, we study a buffer management scheme for sensor nodes which use NVRAM-based hybrid main memory. Though NVRAM is attractive in terms of power consumption and read performance, the write performance (access latency and energy consumption) of NVRAM is worse than that of DRAM. The proposed buffer management scheme employs a simple page migration technique which migrates the data from NVRAM to DRAM when the data needs to be written on NVRAM. Further, in order to secure free DRAM buffer, the proposed scheme employs a greedy deallocation technique that deallocates clean DRAM buffers even though free buffers are still available in the system. The proposed page migration scheme exhibits better performance than legacy buffer management schemes in terms of the buffer hit ratio and the number of writes on NVRAM. The
proposed scheme can be used when DRAM/NVRAM hybrid main memory is adopted in sensor nodes in the near future.

Acknowledgments

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2010-0021897). The author would like to thank Soohyun Yang who helped in the simulation experiment.

References

[1] N. Lin, Y. Dong, and D. Lu, “Providing virtual memory support for sensor networks with mass data processing,” International Journal of Distributed Sensor Networks, vol. 3, Article ID 324641, 20 pages, 2013.
[2] A. Lachenmann, P. J. Marrón, M. Gauger, D. Minder, O. Saukh, and K. Rothermel, “Removing the memory limitations of networks with flash-based virtual memory,” in Proceedings of the Eurosys Conference, pp. 131–144, March 2007.
[3] M. O. Farooq and T. Kunz, “Operating systems for wireless sensor networks: a survey,” Sensors, vol. 11, no. 6, pp. 5900–5930, 2011.
[4] S. R. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong, “TinyDB: an acquisitional query processing system for sensor networks,” ACM Transactions on Database Systems, vol. 30, no. 1, pp. 122–173, 2005.
[5] Y. W. Law, M. Palaniswami, L. V. Hoels, J. Doumen, P. Hartel, and P. Havinga, “Energy-efficient link-layer jamming attacks against wireless sensor network MAC protocols,” ACM Transactions on Sensor Networks, vol. 5, no. 1, article 6, 2009.
[6] Y. Gu and T. He, “Bounding communication delay in energy harvesting sensor networks,” in Proceedings of the 30th IEEE International Conference on Distributed Computing Systems (ICDCS ’10), pp. 837–847, Genoa, Italy, June 2010.
[7] M. Yoon, Y. Kim, and J. Chang, “An energy-efficient routing protocol using message success rate in wireless sensor networks,” Journal of Convergence, vol. 4, no. 2, pp. 15–22, 2013.
[8] G. Carvalho, I. Wounang, A. Anpalagan, and S. Dhurandher, “Energy-efficient radio resource management scheme for heterogeneous wireless networks: a queueing theory perspective,” Journal of Convergence, vol. 3, no. 4, pp. 15–22, 2012.
[9] B. Singh and D. Lobiyal, “A novel energy-aware cluster head selection based on particle swarm optimization for wireless sensor networks,” Human-Centric Computing and Information Science, vol. 2, no. 13, 2012.
[10] R. Sumathi and M. Srinivas, “A survey of QoS based routing protocols for wireless sensor networks,” Journal of Information Processing Systems, vol. 8, no. 4, pp. 589–602, 2012.
[11] L. A. Barroso and U. Hölzle, “The case for energy-proportional computing,” Computer, vol. 40, no. 12, pp. 33–37, 2007.
[12] Y. Xie, “Modeling, architecture, and applications for emerging memory technologies,” IEEE Design and Test of Computers, vol. 28, no. 1, pp. 44–50, 2011.
[13] C. Augustine, N. N. Mojumder, X. Fong, S. H. Choiday, S. P. Park, and K. Roy, “Spin-transfer torque MRAMs for low power memories: perspective and prospective,” IEEE Sensors Journal, vol. 12, no. 4, pp. 756–766, 2012.
[14] D. Lee and S. Gupta K. Roy, “High-performance low-energy STT-MRAM based on balanced write scheme,” in Proceedings of the ACM/IEEE International Symposium on Low Power Electronics and Design (ISLPED ’12), pp. 9–14, 2012.
[15] X. Guo, E. Ipék, and T. Soyata, “Resistive computation: avoiding the power wall with low-leakage, STT-MRAM based computing,” in Proceedings of the 37th International Symposium on Computer Architecture (ISCA ’10), pp. 371–382, June 2010.
[16] M. K. Qureshi, V. Srinivasan, and J. 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), pp. 24–33, June 2009.
[17] P. Zhou, B. Zhao, J. Yang, and Y. Zhang, “A durable and energy efficient main memory using phase change memory technology,” in Proceedings of the 36th Annual International Symposium on Computer Architecture (ISCA ’09), pp. 14–23, June 2009.
[18] H. Park, S. Yoo, and S. Lee, “Power management of hybrid DRAM/PRAM-based main memory,” in Proceedings of the 48th ACM/EDAC/IEEE Design Automation Conference (DAC ’11), pp. 59–64, June 2011.
[19] J. Hu, Q. Zhu, C. Xue, W. Tseng, and E. Sha, “Software enabled wear-leveling for hybrid PCM main memory on embedded systems,” in Proceedings of the Design, Automation & Test in Europe Conference & Exhibition (DATE’13), pp. 599–602, March 2013.
[20] D. Lee and H. Oh, “A lifetime aware buffer assignment method for streaming applications on DRAM/PRAM hybrid memory,” ACM Transactions on Embedded Computing Systems, vol. 12, no. 1, 2013.
[21] D. Kim, S. Lee, J. Chung et al., “Hybrid DRAM/PRAM-based main memory for single-chip CPU/GPU,” in Proceedings of the 49th Annual Design Automation Conference (DAC ’12), pp. 888–896, 2012.
[22] J. Choi, S. Kim, C. Kim, and K. Park, “OPAMP: evaluation framework for optimal page allocation of hybrid main memory architecture,” in Proceedings of the International Conference on Parallel and Distributed Systems (ICPADS ’12), pp. 620–627, 2012.
[23] H. Jang, B. An, N. Kulkarni, K. Yum, and E. Kim, “A hybrid buffer design with STT-MRAM for on-chip interconnects,” in *Proceedings of the ACM/IEEE International Symposium on Networks-on-Chip (NOCS '12)*, pp. 193–200, 2012.

[24] E. Kultursay, M. Kandemir, A. Sivasubramaniam, and O. Mutlu, “Evaluating STT-RAM as an energy-efficient main memory alternative,” in *Proceedings of the IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS '13)*, 2013.

[25] H. Seok, Y. Park, and K. Park, “Efficient page caching algorithm with prediction and migration for a hybrid main memory,” *Applied Computing Review*, vol. 11, no. 4, 2012.

[26] L. E. Ramos, E. Gorbatov, and R. Bianchini, “Page placement in hybrid memory systems,” in *Proceedings of the 25th ACM International Conference on Supercomputing (ICS '11)*, pp. 85–95, June 2011.

[27] O. Zilberberg, S. Weiss, and S. Toledo, “Phase-change memory: an architectural perspective,” *ACM Computing Surveys*, vol. 45, no. 3, 2013.

[28] E. Gal and S. Toledo, “Algorithms and data structures for flash memories,” *ACM Computing Surveys*, vol. 37, no. 2, pp. 138–163, 2005.

[29] Y. Ryu, “SAT: switchable address translation for flash memory storages,” in *Proceedings of the 34th Annual IEEE International Computer Software and Applications Conference (COMPSAC '10)*, pp. 453–461, Seoul, South Korea, July 2010.

[30] Y. Yoo, H. Lee, Y. Ryu, and H. Bahn, “Page replacement algorithms for NAND flash memory storages,” in *Proceedings of the International Conference on Computational Science and its Applications*, pp. 201–212, 2007.

[31] X. Tang and X. Meng, “ACR: an adaptive cost-aware buffer replacement algorithm for flash storage devices,” in *Proceedings of the 11th IEEE International Conference on Mobile Data Management (MDM '10)*, pp. 33–42, Kansas City, Mo, USA, May 2010.

[32] S.-Y. Park, D. Jung, J.-U. Kang, J.-S. Kim, and J. Lee, “CFLRU: a replacement algorithm for flash memory,” in *Proceedings of the International Conference on Compilers, Architecture and Synthesis for Embedded Systems (CASES '06)*, pp. 234–241, October 2006.

[33] H. Kim and S. Ahn, “BPLRU: a buffer management scheme for improving random writes in flash storage,” in *Proceedings of the 6th USENIX Conference on File and Storage Technologies*, 2008.
Submit your manuscripts at
http://www.hindawi.com