Power-Aware Memory Management for Hybrid Main Memory

Youngwoo Park, Dong-Jae Shin, Sung Kyu Park, and Kyu Ho Park
Computer Engineering Research Laboratory
Korea Advanced Institute of Science and Technology
{ywpark, djshin, skpark}@core.kaist.ac.kr, and kpark@ee.kaist.ac.kr

***Abstract*—In this paper, we propose the main memory management mechanism of the Operating System (OS) for the hybrid main memory which combines both advantages of DRAM and PRAM (Phase Change RAM). We separate memory pages into hot and cold pages, and dynamically located them to the proper memory region. It was implemented into the Linux OS. We decreased the energy consumption of main memory up to 35% while performance degradation was less than 5%. Moreover, when we decreased the write count for PRAM, life-time of PRAM increased. In addition to these advantages, the system does not need additional hardware.**

I. Introduction

For several decades, DRAM has been computer's main memory system. Main memory size grew to support many cores and current applications, such as cloud computing environment, in modern computer systems. For large scale main memory design, DRAM idle power has non-negligible effect on the total power consumption. The power absorption of DRAM based main memory has significantly increased in proportion to its size. Currently, the portion of power consumption which is consumed by DRAM in a server system is about 20% [1], thus it is the main factor for reducing the energy utilization. Furthermore, DRAM has the scalability wall for sub-45nm technology. Therefore it has almost reached the limit of increasing the memory capacity.

As the emerging new memory technologies, various researches on these new memory devices are on-going to replace DRAM. They are called resistive memory. One of the resistive memories is PRAM (Phase Change Memory). PRAM is one of the promising candidates for future main memory since PRAM is a byte-addressable and non-volatile memory which has no refresh power in idle state. It is also believed that PRAM is more scalable than DRAM. However, PRAM has limit of 10^8 times to writing operation, high write latency (5–10 times higher than DRAM), and high write energy. Because of these problems, it is difficult to use only PRAM as the main memory.

In order to overcome the limitations of DRAM and PRAM, hybrid main memory architecture is presented for exploiting both advantages. DRAM is used to make up for the slow latency of PRAM and PRAM is used to reduce the consumed power. It demands new memory management mechanisms to achieve low power consumption and high performance in hybrid main memory architecture.

In this work, we address OS level power-aware memory management problems for next generation's main memory architecture. Proposed ideas above target the hybrid main memory of DRAM and PRAM. In addition, the emerging memory technology provides the partial DRAM memory to turn-on and off [2][3]. We propose three techniques for hybrid main memory management with following contributions.

- In order to eliminate the hardware support overhead, we present a memory access monitoring technique by periodically scanning a page table.
- With the information from the memory access monitoring, it is possible to identify which pages are hot or cold. Through this identification, we set up the page migration policy for reducing the power consumption.
- We can exploit turn-on and turn-off ability of DRAM by applying the page culling and gathering technique. This technique helps to minimize the DRAM's idle power.

The remainder of this paper is organized as follows. This paper starts from previous works for cutting down the power consumption of main memory. In section 3, we describe a hybrid main memory architecture. We then describe our proposed main memory management scheme in section 4. The performance evaluation results are placed in section 5 and section 6 concludes this paper and presents further work.

II. Related Work

Many researchers have tried to reduce power consumption of main memory. In case of only using DRAM as the main memory, previous approaches control the power state of memory device through access pattern monitoring [4][5][6]. Nevertheless, DRAM main memory consumes power in idle state because of volatile characteristics. For large scale main memory design, DRAM idle power has non-negligible effect on total energy consumption. Therefore, exploiting the emerging memory technologies which provide the fine-grained DRAM memory turn on,off ability are also used for reducing power consumption in idle state [2][3].

As emerging non-volatile memory technologies, several researches have tried to replace DRAM with non-volatile memory like PRAM as a candidate of the main memory [7][8][9][10]. These works reduce power dissipation compared to DRAM main memory architecture because non-volatile memory has low idle power. But non-volatile memory

has many challenging issues such as slow latency and limited endurance to be used as the main memory.

In order to overcome these limitations, hybrid main memory architecture which combines DRAM with non-volatile memory is proposed for achieving both high performance of DRAM and low energy consumption of non-volatile memory [11][12]. There are two design options for composing hybrid main memory architecture. First, DRAM is used as a buffer, located in front of non-volatile memory which is used for main memory [11]. It can improve performance by buffering the data which is updated frequently in DRAM. And it also overcomes the endurance problem by reducing the write count to non-volatile memory. However, it cannot directly access to data in non-volatile memory region and the memory controller should be changed to support this architecture.

As another option, DRAM and non-volatile memory are located in the same linear region [12]. This architecture can directly access data in both DRAM and non-volatile memory while exploiting a conventional memory controller. Despite it needs additional OS support to allocate or migrate data to DRAM or non-volatile memory, and also has an additional hardware overhead for evaluating write count in non-volatile memory, these researches proposed novel main memory architectures and showed better performance compared to the conventional hybrid architecture. Moreover, it is necessary to consider more efficient techniques for power utilization in this architecture.

In this paper, we propose page migration and page culling and gathering schemes for cutting down the power consumption in hybrid main memory architecture. We also present an memory access monitoring techniques without H/W overhead.

III. Hybrid Main Memory Architecture

Fig. 1 shows the hybrid main memory architecture we targeted. Both DRAM and PRAM are used as the main memory. This architecture reduces the power budget because large portion of main memory is replaced by PRAM. Using small size of DRAM, it can minimize the performance degradation. Similar to the traditional main memory architecture, there is memory page swap between hybrid main memory and the second level storage. On the other hand, the number of page swapping can be reduced because the main memory capacity is increased by PRAM.

In the hybrid main memory architecture, DRAM and PRAM are assigned to a single physical address space. All memory pages of PRAM and DRAM can be directly managed by the Linux kernel. Thus it is necessary for kernel to distinct the DRAM and PRAM region. We assume that DRAM always has lower physical address than PRAM and the size of each memory is provided by the kernel option. Linux kernel has information of the exact physical address range of DRAM and PRAM as shown in Fig. 1. Physical pages of DRAM and PRAM can be distinguished by the physical address.

IV. Proposed Idea

This section explains how to manage the hybrid main memory architecture for reducing the energy consumption.

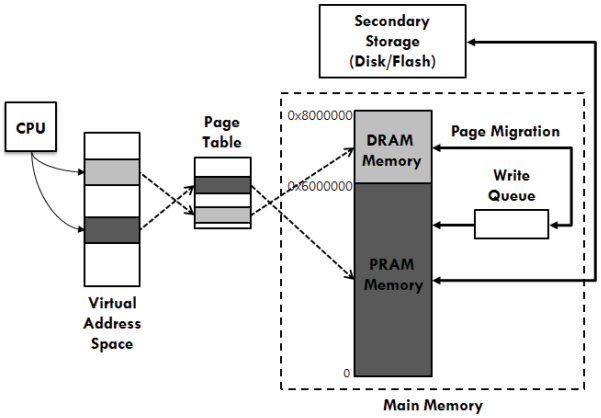


Fig. 1. Hybrid Main Memory Architecture

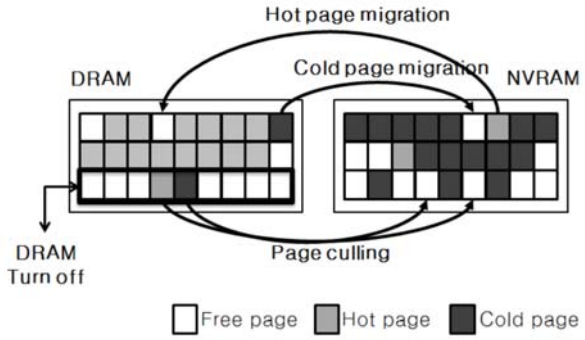


Fig. 2. Power-aware Page Migration and Culling

We propose three techniques: memory access monitoring without hardware support, page migration with hot/cold page separation, and page culling. In the following sub-sections, we describe these techniques in detail.

A. Memory Access Monitoring without Hardware Support

It is important to monitor the memory access pattern for efficient hybrid memory management. According to data allocation between DRAM and PRAM, it shows extremely different power absorption. The reason why it absolutely influences the amount of energy consumption will be explained in detail in the following section. We present a memory access monitoring technique by exploiting a page table instead of using an additional hardware. Using kernel thread, we periodically monitor the memory access. Two reference bit in the page table are used to separate hot/cold pages. Two consecutive memory references mean a hot page. On the contrary two consecutive non-references mean a cold page. A 2 times of continuative accesses means that the page is referenced frequently and vice versa. With this information, we can implement other techniques such as page migration and page culling.

B. Page Migration with Hot/Cold Page Separation

There are two reasons that hot pages should be moved to the DRAM. First, access delay of PRAM is more than

DRAM's one especially for write operations. Second, the write operation to the PRAM absorbs extremely high energy. The page migration tries to move hot pages in DRAM and cold pages in PRAM as shown in Fig. 2. If hot page is located in PRAM, it consumes much dynamic power. On the other hand, cold page in DRAM only consume idle power without access. In hybrid main memory, the page migration is necessary to reduce latency overhead as well as power consumption. Because PRAM access latency is slower than DRAM, hot page in PRAM increases total access latency of main memory. The page migration decreases the number of access for PRAM.

C. Page Culling

With minimum set of active pages in DRAM, we can further reduce main memory's power consumption by turning off unused DRAM region. If there is single active page in DRAM region, we cannot turn off that region since DRAM is volatile memory. In order to maximize the turn-off efficiency of DRAM, we propose page culling and gathering. Page culling finds DRAM region which contains few used pages and compulsorily migrate those pages to PRAM as shown in Fig. 2. Therefore, we can turn off one more DRAM region. If the migrated pages are moved to DRAM again, we use page gathering mechanism. It always search for free pages in online DRAM region and helps minimize DRAM idle power. For page culling implementation, we assume 4MB DRAM region as minimum unit of turn off memory.

V. Evaluation

From now on, we explain the experimental setup such as hardware specification and performance metrics. We then show the performance results which are energy consumption and memory access delay compared with the conventional main memory architecture.

A. Experimental Setup

We extend the Linux operating system to implement power-aware memory management in the real hardware platform whose specification is shown in TABLE I. For the evaluation, we simulated hybrid main memory using two virtual NUMA nodes. Using page fault exception, we traced every memory access in Linux kernel. Based on access trace, we calculated the total access latency and energy consumption. For this calculation, we used the memory specifications in TABLE II. We selected and tested memory intensive workload of SPEC CPU2000 benchmark suite.

B. Experimental Result

We give experimental results in this section by showing the energy consumption and memory access delay. By comparing DRAM main memory architecture, we show that our proposed scheme can achieve energy saving with a little overhead.

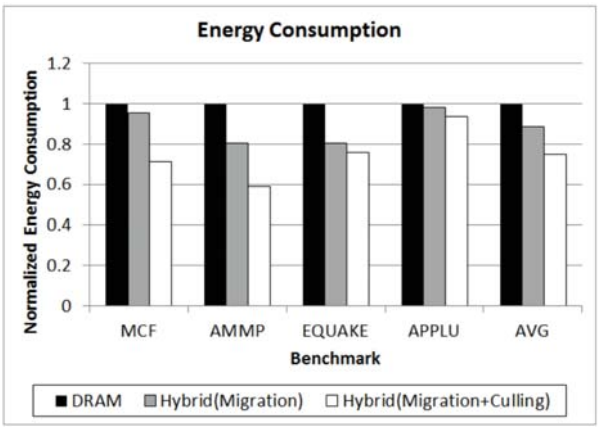
As the first experiment, we compared the energy consumption. As shown in Fig 3(a), it is possible to reduce the energy consumption by average 25% compared to DRAM based main memory. When applying both techniques which are page migration and page culling, it shows the best result. It signifies that it is important to allocate pages to appropriate locations between DRAM and PRAM as well as turning off

TABLE I
Hardware Specification

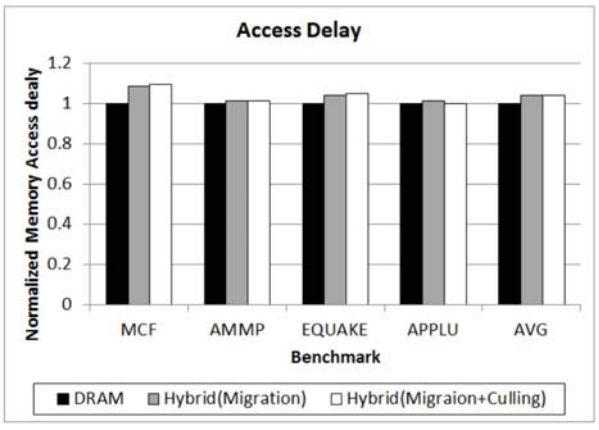
Experimental Hardware	
Processor	Intel Core2 CPU E6600 Clock: 2.5GHz Last Level Cache Size: 4MB
Memory	DDR3 1066MHz 1GB * 2 = 2GB
Operating System	Linux 64bit Modified 2.6.31 kernel

TABLE II
Memory Characteristics [12]

Parameter	DRAM	PRAM
Row Read Power	210mW	78mW
Row Write Power	195mW	773mW
Active Power	75mW	25mW
Standby Power	90mW	45mW
Refresh Power	4mW	0mW



(a) Energy Saving



(b) Performance Overhead

Fig. 3. Energy Saving and Performance Overhead Results

when DRAM region is unused.

In order to show the performance overhead, we evaluate the memory access delay. Because DRAM access latency is faster than PRAM access latency, hybrid main memory is inevitably slower than DRAM based main memory. However, we can increase the memory access delay by up to 5% with efficient page migration and page culling by considering hot/cold pages.

From our proposed techniques which are memory access monitoring, page migration, and page culling, we can highly reduce the energy consumption with negligible performance overhead.

VI. Conclusion

We propose the main memory management mechanism for the hybrid main memory which consists of DRAM and PRAM. We present three techniques which are memory access monitoring, page migration, and page culling. The memory access monitoring can eliminate additional hardware overhead for monitoring memory access patterns. Through page migration, cold pages are moved to PRAM and hot pages are allocated to DRAM, allowing us to reduce the energy consumption of main memory architecture. With the page culling, which allows DRAM to turn off when DRAM is unused, it is possible to reduce the energy consumption. The experimental results show that the proposed techniques can reduce the energy consumption by up to 35% with 5% delay overhead of memory access.

VII. Further Work

Although current evaluation only shows the possibility of OS-level power management using hybrid main memory, we expect that proposed mechanisms further decrease main memory energy consumption in real system. In real OS environment where many applications are concurrently executed, the use of main memory is more dynamic. There are many page allocations and it is distributed throughout all memory regions. Memory access patterns become complicated and diversified. The classical page allocation and reclamation algorithms does not satisfy performance and power requirement of future main memory architecture. The page monitoring and classification is also difficult to measure. It is necessary to develop methods to reduce page level memory management overhead. By discussion of OS-level page management mechanism, we have more chance to improve performance and energy consumption of current DRAM based main memory.

Acknowledgment

The work presented in this paper was supported by MKE (Ministry of Knowledge Economy, Republic of Korea), Project No. 10035231-2010-01.

References

[1] S. Electronics, "Memory specification," in *http://www.samsung.com/global/business/semiconductor/Greenmemory/Applications/ServerStorage/ServerStorage_DDR3.html*.

[2] A. N. Udipi, N. Muralimanohar, N. Chatterjee, R. Balasubramonian, A. Davis, and N. P. Jouppi, "Rethinking dram design and organization for energy-constrained multi-cores," in *Proceedings of the 37th annual international symposium on Computer architecture*, ser. ISCA '10. New York, NY, USA: ACM, 2010, pp. 175–186. [Online]. Available: http://doi.acm.org/10.1145/1815961.1815983

[3] J. Haas and P. Vogt, "Fully-buffered dimm technology moves enterprise platforms to the next level," *Technology@Intel Magazine*, March 2005.

[4] H. Huang, P. Pillai, and K. G. Shin, "Design and implementation of power-aware virtual memory," in *Proceedings of the annual conference on USENIX Annual Technical Conference*. Berkeley, CA, USA: USENIX Association, 2003, pp. 5–5. [Online]. Available: http://portal.acm.org/citation.cfm?id=1247340.1247345

[5] V. Delaluz, M. Kandemir, N. Vijaykrishnan, A. Sivasubramaniam, and M. J. Irwin, "Hardware and software techniques for controlling dram power modes," *IEEE TRANSACTIONS ON COMPUTERS*, vol. 50, pp. 1154–1173, 2001.

[6] X. Li, Z. Li, Y. Zhou, and S. Adve, "Performance directed energy management for main memory and disks," *Trans. Storage*, vol. 1, pp. 346–380, August 2005. [Online]. Available: http://doi.acm.org/10.1145/1084779.1084782

[7] D. Roberts, T. Kgil, and T. Mudge, "Using non-volatile memory to save energy in servers," in *Proceedings of the Conference on Design, Automation and Test in Europe*, ser. DATE '09. 3001 Leuven, Belgium, Belgium: European Design and Automation Association, 2009, pp. 743–748. [Online]. Available: http://portal.acm.org/citation.cfm?id=1874620.1874804

[8] 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*, ser. ISCA '09. New York, NY, USA: ACM, 2009, pp. 14–23. [Online]. Available: http://doi.acm.org/10.1145/1555754.1555759

[9] B. C. Lee, E. Ipek, O. Mutlu, and D. Burger, "Architecting phase change memory as a scalable dram alternative," *SIGARCH Comput. Archit. News*, vol. 37, pp. 2–13, June 2009. [Online]. Available: http://doi.acm.org/10.1145/1555815.1555758

[10] A. P. Ferreira, M. Zhou, S. Bock, B. Childers, R. Melhem, and D. Mossé, "Increasing pcm main memory lifetime," in *Proceedings of the Conference on Design, Automation and Test in Europe*, ser. DATE '10. 3001 Leuven, Belgium, Belgium: European Design and Automation Association, 2010, pp. 914–919. [Online]. Available: http://portal.acm.org/citation.cfm?id=1870926.1871147

[11] 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*, ser. ISCA '09. New York, NY, USA: ACM, 2009, pp. 24–33. [Online]. Available: http://doi.acm.org/10.1145/1555754.1555760

[12] G. Dhiman, R. Ayoub, and T. Rosing, "Pdram: a hybrid pram and dram main memory system," in *Proceedings of the 46th Annual Design Automation Conference*, ser. DAC '09. New York, NY, USA: ACM, 2009, pp. 664–469. [Online]. Available: http://doi.acm.org/10.1145/1629911.1630086