Journal of Theoretical and Applied Information Technology
15th September 2017. Vol.95. No.17
© 2005 - Ongoing JATIT & LLS

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

EFFECTIVE PAGE REPLACEMENT ALGORITHM OF HYBRIDE MEMORY

[1]BOSUNG JUNG, [2]SEUNGMIN LEE, [3]JUNGHOON LEE
[1,3] *Dept. of Control & Instrumentation, Gyeongsang National University, Korea*)
[2] *Dept. of Multimedia, Namseoul University, Korea*
E-mail: [1]blueking80@gnu.ac.kr, [2]mini0920@nsu.ac.kr, [3]leejh@gnu.ac.kr

ABSTRACT

The main objective of this paper is to implement high-performance next-generation main memory by proposing an effective page replacement algorithm for the hybrid structure of DRAM and PCM. To replace conventional DRAM, a DRAM&PCM hybrid memory is one of the effective structures because it can utilize an advantage of DRAM and PCM. However, in order to use the characteristics of DRAM and PCM, pages should be replaced frequently. This is still a major problem in PCM that have write-limits. Therefore, it needs an effective page management method for exploiting each memory characteristics dynamically and adaptively. To reduce frequent page replacement, page replacement in the proposed hybrid memory is handled by two localities in PCM and recent write requests in DRAM. According to our simulation, the proposed algorithm for the DRAM&PCM hybrid can reduce the PCM write count by around 22% and the average access time by 31% given the same PCM size, compared with Clock-DWF algorithm

Keywords: *Hybrid Memory, Page Management, Temporal/Spatial Locality, Memory Characteristics*

1. INTRODUCTION

With the advent of recent large-capacity applications and the use of multi-core processors, the memory working set continues to increase. In addition, with the rapid growth of the portable mobile device market and the emergence of social networks in recent years, the demand for personal and network data storage is continuously increasing. Up to now, DRAM, with its high-speed access time and low price, has been used as the main memory of a computing system. However, it is no longer the ideal memory [1], due to the power consumption required to store the data and the difficulty of high integration.

Highly integrated non-volatile memories such as PCM (Phase Change Memory), RRam (Resistive RAM), and STT-RAM (Spin-Transfer Torque RAM) are attracting attention as next-generation memories to replace DRAM [2]~[6]. In particular, PCM is attracting the most attention. Unlike DRAM, PCM does not require a refresh operation, and utilizes low standby power consumption. Also, as with DRAM, it is possible to access byte address, and it can be produced almost in the same way as the DRAM process. Therefore, it is advantageous from the viewpoint of capital investment and development costs compared to other non-volatile memories [7].

However, despite the high integration and low power consumption, PCM has a major disadvantage when it comes to replacing DRAM. First, PCM has a limited number of write operations, like NAND flash. It is known that the number of PCM write operations is 10^7-10^8. Second, PCM has a slower memory access time than DRAM. Compared to DRAM, the read operation is 1 to 2 times slower, and the write operation is about 7 to 10 times slower [8]~[10].

Therefore, in order to overcome the drawbacks of PCM various studies are being conducted. A hybrid memory that combines DRAM and PCM is one effective solution. In hybrid memory, if the DRAM is able to effectively store pages that can be referred to in the near future, the PCM can effectively provide long write latency. Otherwise, frequent page replacement of PCM and DRAM can occur, which can result in system performance degradation. Therefore, in a hybrid memory structure, an effective page replacement policy is important.

In this paper, we proposed an effective page replacement policy algorithm for a hybrid memory of DRAM and PCM for next generation main memory. In the proposed hybrid structure, DRAM and PCM are located in the same layer. Then, in order to reduce frequent page replacement, page replacement in the proposed hybrid memory is handled by two localities in PCM and recent write requests in DRAM.

2. BACKGROUND

2.1 The hybrid memory structure

As shown in Fig. 1, based on the operation of DARM, the structure of the hybrid memory can be divided into two structures. Fig. 1(a) shows that DRAM is utilized as a buffer cache of PCM. In this case, since access to the memory is sequentially generated from the DRAM to the PCM, it is possible to reduce access to the PCM using a page that has frequently requested blocks. However, this hybrid memory structure has the disadvantage that it cannot manage DRAM and PCM using the conventional page table.

In the case of Fig. 1(b), PCM and DRAM are used as main memory in the same hierarchy. When using as memory of the same hierarchy, it is possible to use a page table like a conventional memory page policy. Generally, in order to use effective memory, pages with a high re-reference possibility are stored in the DRAM, and pages that are discarded from the DRAM will be stored in the PCM. However, page replacement can occur frequently between DRAM and PCM in order to save pages that are likely to be re-referenced to DRAM. Therefore, the hybrid memory requires a page replacement policy suitable for the characteristics of DRAM and PCM when a program performs an operation

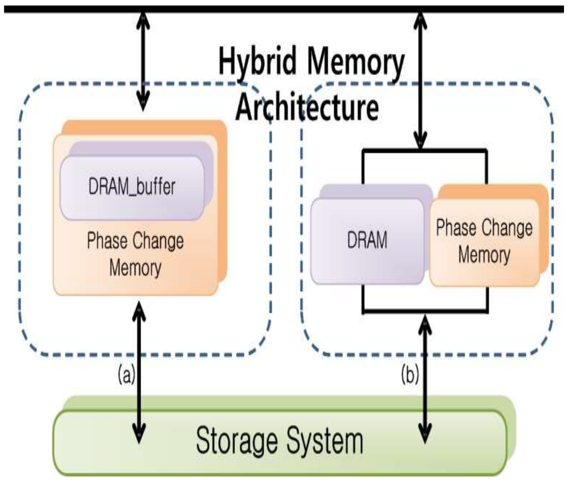

Figure 1: DRAM & PCM Hybrid memory architecture

2.2 Related studies

As typical method for to improve PCM performance, wear leveling and hybrid structure is representative.

The wear leveling method limits the number of writes to specific pages or blocks that frequently require PCM write operations, so it can extend the life of the PCM[1][11][12]. However, this method cannot improve the performance of read/write operation by merely having the effect of limited PCM write operation.

Hybrid structure is method utilizing DRAM that guarantees high-speed memory access time to PCM. So it can effectively reduce the disadvantage of PCM such as high latency and limited write number.[13]~[17].

Qureshi[13] and Lee[14] proposed a PCM structure with a DRAM buffer. Since Qureshi utilizes a DRAM buffer with the same page size as PCM, it reduced the write operations of PCM, and improved the complete memory system performance. On the other hand, Lee [14](we called the Selective Data Buffering) adapts DRAM buffers with blocks of various sizes to store data in which write requests occur frequently, thereby reducing the write count of PCM and improving performance. However, as mentioned above, these buffer structures require a buffer controller.

Dhimna[15](PDRAM), Lee[16] (CLOCK -DWF) and Lee [17] (M-CLOCK) proposed a hybrid memory architecture in which DRAM and PCM are in the same hierarchy. PDRAM proposed an algorithm where the page with a counter of the write operations that reached the critical point in the PCM is stored in the DRAM. However, this did not consider the characteristics of the program execution, where a write operation occurs intensively in a specific block or word.

CLOCK-DWF and M-CLOCK reduced the write operation of PCM using an algorithm that stores pages referenced recently in DRAM for a long time. However, it is difficult to operate efficient memory because the page is saved in DRAM or PCM using write and read operations. Furthermore, when a write hit to PCM occurs, pages are exchanged in the DRAM, so page replacement can occur frequently. Unlike CLOCK-DWF, M-CLOCK is stored in the priority DRAM of the page request and uses the lazy algorithms that replace pages from PCM to DRAM. However, the proposed lazy algorithm is applied only when there is no page that can be evicted from DRAM to PCM.

The existing hybrid memory structure does not consider the characteristics of the program for frequent block access. Therefore, in this paper, we proposed a page replacement algorithm suitable for program characteristics when PCM and DRAM pages are exchanged

3. THE PROPOSED HYBRID MEMORY

The main objective of this paper is to implement high-performance next-generation main memory by proposing an effective page replacement algorithm for the hybrid structure of DRAM and PCM.

3.1 Motivation

A hybrid memory structure operating in the same hierarchy of DRAM and PCM can effectively manage the pages of DRAM and PCM by using a conventional page table. Furthermore, by employing DRAM without a write limit, PCM can reduce write operations, which is the major disadvantage of using it as a main memory. Therefore, it is effective for DRAM to store the page where the write operation has been frequently requested. PCM is effective in storing the page that requested the read operation because it has a similar read latency to DRAM. Even though DRAM can reduce PCM write operations, this frequent page replacement between DRAM and PCM causes degradation of the overall memory system performance.

Table 1 shows the performance of DRAM and PCM. As shown in Table 1, the read latency from 64-byte access of DRAM and PCM is similar, but the write latency of PCM is approximately 10 times slower than DRAM. When considering the page size of the main memory, page replacement between the DRAM and PCM of the hybrid memory requires latency of approximately 32*us*. However, for such a reduction in replacement latency, delaying page replacement from PCM to DRAM can cause shortening of the lifetime for a specific page of PCM.

Table 1: DRAM & PCM Characteristics

	DRAM	PCM
Cell size	6-10F^2	4-10F^2
Access Granularity	64bytes	64bytes
Read latency	50ns	50ns
Write latency	50ns	500 ns
Endurance	>10^{15}	10^8-10^9
Standby Power	Refresh	X

When a program is implemented, the data has both spatial and temporal locality. Performing a write operation to the PCM with pages having a high spatial locality can reduce the page replacement between the DRAM and PCM, reducing the overall latency.

Therefore, in this paper, we proposed an effective page replacement policy utilizing two localities for hybrid memory consisting of DRAM and PCM.

The page replacement policy proposed is as follows:

1) A page fault of hybrid memory or a page with a temporal locality in PCM is stored in DRAM.

2) If a page hit to the DRAM occurs, the proposed DRAM status bits are updated to match the read/write operation. If there is no invalid page in the DRAM and the new page is stored in the DRAM, the DRAM of the proposed algorithm will select the victim page where all the status bits are '0' and store it to the PCM.

3) If the read operation hit is in PCM, the reference bit of PCM is simply updated. On the other hand, when a write operation hit occurs, the proposed algorithm for PCM will determine the access type of the temporal or spatial localities. If the access type of the hit is a temporal locality, the PCM page will be replaced by the victim page of the DRAM. On the other hand, if the access type of the hit is a spatial locality, the state bits of the proposed PCM are updated. If there is no invalid page in the PCM to store the new page, a page in which the value ('0') of the state bits is evicted.

3.2 The proposed algorithm

In the proposed hybrid memory, the DRAM has three state bits: Reference bit (DRAM_R, DR), Write_hit bit (W), and Frequency bit (F). The DR bit indicates whether a page is referenced in the DRAM or not. If a page of the DRAM has a hit due to a read or write operation request, the DR bit is updated to a value of '1', and the DR bit will be stored in the PCM for the proposed algorithm when the page is evicted from the DRAM.

The W bit consists of two bits (W[0], W[1]). The W[0] bit indicates a page where the write operation has been requested recently in the DRAM, and the W[1] bit indicates a page where the write operation occurred previously. When a write hit occurs on a page of DRAM, only the W[0] bits that indicate the recent write operation are updated to the value of '1'.

Finally, the F bit composed of 1 bit indicates the page in which the write request is temporally generated in the DRAM. In the proposed algorithm, the frequent write operation of a page is defined as the state where both W[0] and W[1] are

'1' when a write hit occurs in the page. Accordingly, when a write hit to the page of the DRAM occurs, if the value of W[0] and W[1] is '1', the F bit is updated to '1' .

If a miss occurs in the hybrid memory and there is no invalid page in the DRAM, the DRAM will check the status bits of each page. If the status bits are all '1', the page will be selected as the victim page and will be moved to the PCM. Then the state bits of a newly stored page are all updated to '0'. On the other hand, if the state bits are not all '0', the value of each state bit will be moved to a bit in another state. That is, the W[1] bit is updated by the W[0]bit, and the W[1] bit is updated by the F bit. The value of the F bit is also updated to '0'. These operations are performed cyclically from the DRAM until the victim page is selected.

The state bits of the PCM consist of the Reference bit (PCM_R, PR), which indicates the access hits by read or write operations, the Dirty_Count bit (DC), which indicates the number of write hits with spatial locality, and the Block_Dirty bit (BD), which indicates the small block of a page where the write operation occurred. Therefore, if the small block size of the proposed write operation is 64 bytes, the page has a BD of 64 bits and the DC is composed of 8 bits because the basic page size is 4 Kbytes.

The PR is updated to '1' when the page of the PCM experiences a hit. If there is a write hit to the page of the PCM and the BD bit of the referenced block is '0', the BD bit is updated to '1', at which time the value of the DC bit will increase. On the other hand, if the BD bit of the write request block is already '1', it is determined by the proposed algorithm that the page has a temporal locality, and it is replaced by the victim page of the DRAM. At that time, the PR bit is updated by the DR bit of the victim page in DRAM, and all other remaining status bits are updated to '0'. If there is no invalid page to store a new page in the PCM, the PCM will check the PR and DC bits. If The PR bit is '1', the PR bit will be updated to '0' and the DC bit will decrease. The PCM will then check the status bits of the next page. On the other hand, when the PR bit is '0', only the DC bit is checked. If the DC bit is not '0', the value of the DC bit is decreased and the next page is checked. These operations occur cyclically in the PCM until the PR and DC bits of some page have a value of '0'. Therefore, if the PR and DC bits of the PCM are all '0', that page will be selected as the victim page. Then, the victim page is stored in the lowest hierarchical memory.

Fig. 2 shows a concrete operation of the proposed hybrid memory. Fig. 2(a) shows the state of the DRAM, and Fig. 2(b) shows the state of the PCM. In this example, it is assumed that the DRAM and PCM have four pages, and the page of the PCM consists of four blocks:

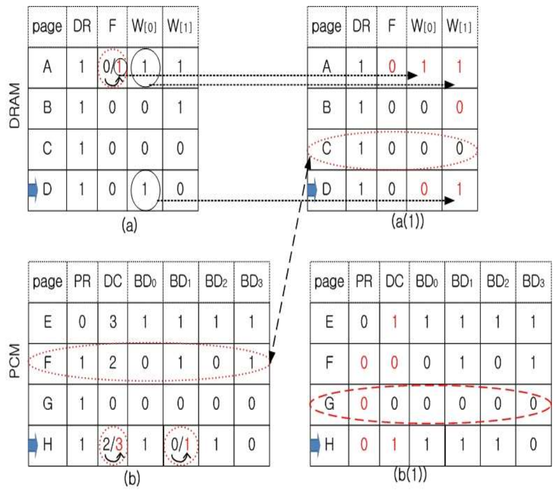

Figure 2 : The proposed algorithm operations

1) DRAM Hit: In Fig. 2(a), when page 'A' experiences a write operation hit, the F bit of page 'A' is updated to '1' using our algorithm because the W[0] bit is '1' and W[1] bit is '1'. If the write operation hit occurs on a page other than page 'A', only the W[0] bit is updated to '1'.

2) PCM Hit: If a read operation hit occurs in the PCM, only the PR bit of the PCM is updated to '1'. In Fig 2(b), if the BD_1 of page 'H' experiences a write operation hit, the BD_1 bit is updated to '1' , and the value of DC is updated from '2' to '3' because BD_1 is a block in which the write operation has not yet occurred. On the other hand, when the BD_3 of page 'F' experiences a write operation hit, page 'H' is exchanged for the victim page of the DRAM because the BD_3 bit is '1,' that is, defined as a temporal locality by the proposed algorithm.

2-1) Victim page selection of the DRAM : In Fig. 2(a), checking for victim page selection starts with page 'D'. Page 'D' is excluded from the candidate victim page because the W[0] bit is '1'. At this time, the W[0] bit of page 'D' is stored in W[1], the F bit is stored in W[1], and the F bit is updated to '0'. Then, the state bits of the next page 'A' will be checked. Because none of the state bits (F, W[0],. W[1]) of page 'A' are '0', page 'A' is excluded from the candidate victim page. At this time, the values of the state bits of page 'A' are also updated. Then, the state bits of the next page 'B' will be checked.

Page 'B' is also excluded from the candidate victim page because of the W[1] bit, and the values of the state bits are updated. Then, the state bits of the next page 'C' will be checked. Page 'C' is selected as the victim page because the state bits (F, W[0], W[1]) are all '0', and it is exchanged with page 'F' of the PCM. When page 'C' is stored in the PCM, the PR bit is updated by the DR bit of page 'C', and all other state bits in the PCM are updated to '0'. When page 'F' is stored in the DRAM, all state bits of page 'F' of the DRAM are updated to '0'. Then, the starting page to select the next victim page is page 'D' because page 'D' is the next page of the newly stored page 'F' in the DRAM.

Fig. 2(a(1)) shows the state bits of the page after selecting the victim page of the DRAM.

3) Hybrid memory miss : If a miss occurs in the hybrid memory, a new page requested from the lowest hierarchical memory is stored in the DRAM. In Fig. 2(a(1)), the victim page 'C' is stored in the PCM. The proposed algorithm will select the victim page in Fig. 2(b) and will check the state bits from page 'H'. Page 'H' is excluded from the candidate victim pages because the PR bit and DC bit of page 'H' are not all '0'. The PR bit is updated to '0', and the DC value is decreased from '3' to '2'. Then, the state bits of the next page 'E' will be checked. Even if the PR bit of page 'E' is '0', page 'E' is excluded from the candidate victim page because the value of DC is '3', and the value of DC is decreased from '3' to '2'. The next page 'F' is also excluded from the victim page due to the PR bit and DC value, and the PR bit and the value of DC are updated (the PR bit is '0', and the value of DC is '1'). The state bits of the next page 'G' will be checked. Page 'G' is also excluded from the candidate victim page because the value of DC is '0' but the PR bit is '1', and the PR bit is updated to '0'.

If the proposed algorithm does not select a victim page after checking all the pages of the PCM, the proposed algorithm will check the state bits of all pages until a victim page is selected. Therefore, in this example, the page of the PCM will again check the state bits. In this example, page 'G' is selected as the victim page of the PCM. Fig. 2(b(1)) shows the result of the state bits after the victim page 'G' of the PCM is selected.

4. RESULTS AND DISCUSSION

In this paper, we modified Valgrind's Cachegrind [18] for performance evaluation and extracted a SpecCpu 2006 benchmark trace file [19]. After approaching from the L1 cache and L2 cache, a miss trace was used for the 100 million addresses of data accessed in memory. We also utilized the DRAM and PCM characteristics for the simulation, as shown in Table 1.

For performance evaluation, we adopted the PCM of a 4-Gbit capacitor for the proposed hybrid memory. The DRAM was utilizing 25% of the capacity of the PCM.

We evaluated the performance to select the block size of the write request for effective spatial locality. Fig. 3 shows the average of the block write count and page write count in the PCM. Here, the block_write count indicates that the write operation is executed only once with the proposed block size and the page_write count means that the write operation of the page unit is executed in the PCM. The write block size of the page was evaluated from 64 bytes to 512 bytes.

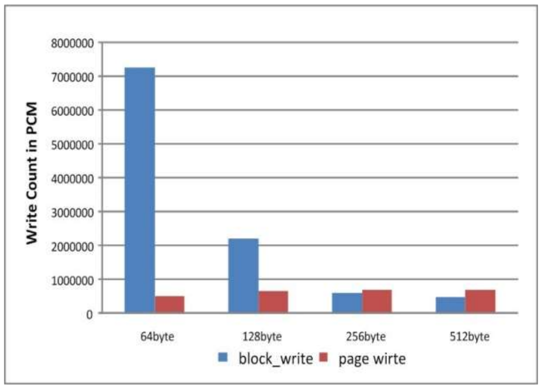

Figure 3 : Write count of the Block and Page in PCM

As shown in Fig.3, the page_write with the block size of 64 bytes is reduced by an average of 25% compared with the page_write of the block size. On the other hand, the block_write of the 64-byte block size is increased by an average of approximately 92% compared to the 512-byte and the 256-byte block sizes, and by approximately 70% compared to the 128-byte block size. This means that the proposed page replacement algorithm effectively reduces the page replacement of the PCM and DRAM by a 64-byte block size.

Fig. 4 shows the average write count by converting the proposed block size into page units. The page size is 4 Kbytes. In Fig. 4, the 64-byte block size reduces the write operation of 4 Kbytes on average by approximately 18% compared with the other block size.

And, we also compare the page write count of CLOCK-DWF, which has the same structure and purpose as the proposed hybrid memory. As

Journal of Theoretical and Applied Information Technology
15th September 2017. Vol.95. No.17
© 2005 - Ongoing JATIT & LLS

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

shown in Fig. 4, the proposed page replacement algorithm reduces the write count of the page as a whole compared to CLOCK-DWF. For a block size of 64 bytes, we reduce write operations by approximately 22% compared to CLOCK-DWF.

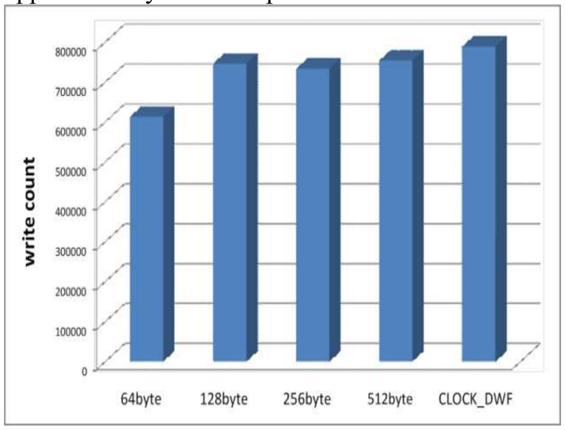

Figure 4: Average writing counts of a page in PCM

In this paper, the Average Memory Access Time (AMAT) was measured in order to evaluate the overall performance of the proposed page replacement algorithm. This is one of typical evaluation criteria for memory systems. We used the parameters of table 1 for DARM and PCM. And the memory access time of the lowest hierarchy is obtained by using a tool [20] when a miss occurs in the hybrid memory, and is defined as 15*ms* in this paper.

Fig. 5 shows the average memory access time of hybrid memory that has a different block size in PCM.

As shown Fig. 5, the best performance has been improved with a structure in which one page consists of a block size of 64bytes. However, 'mcf' and 'lbm' have the worst performance with a block size of 64bytes compared with other block sizes. Analysis of the simulation results showed that 'lbm' and 'mcf' in a 64bytes block had high spatial locality for write operation. Also, frequent requests were also made for reading operation. The structures with 64byte blocks have high memory latency because the pages are replaced to the DRAM after many write requests in the PCM compared with other blocks.

If the page stored in PCM has high spatial locality, PCM with a block size of 64bytes will save the page for a long time. This means that pages with high spatial locality stored in PCM are likely to be selected the victim pages compared with other blocks.

As a result, the structure with a block size of 64byte reduces the average memory access time by about 11% compared with other blocks. Therefore, we chose 64bytes in the block size for the spatial locality of the hybrid memory.

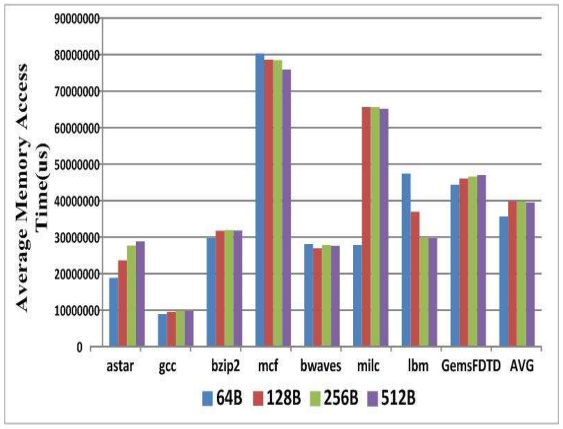

Figure 5: Average Memory Access by the different block size of PCM

In this paper, we compared the average memory access time with CLOCK-DWF that has the same structure and purpose for performance evaluation.

Fig. 6 shows the average memory access time both the proposed structure and CLOCK-DWF. In Fig. 6, all benchmarks except for 'lbm' displayed a better performance than CLOCK-DWF. Based on an analysis of the simulation results, in the case of 'lbm', the miss count during the entire 100 million data address execution was less than the suggested memory capacity. Therefore, 'lbm' did not have additional page fault. And as mentioned above, 'lbm' showed characteristics with high spatial locality for write operation and frequent read operation. Because the 'lbm' page has high spatial locality, the PCM of the proposed hybrid memory generated more the hits than CLOCK-DWF. Therefore, many write operations occurred in PCM. As shown in Table 1, the write operation has latency approximately 10 times higher than read operation. According to simulation, when the counter of block_write of 'lbm' in the propose structure was change to 4Kbyte, it became the same as the counter of the page_write. Therefore, the proposed structure caused more of the write operation in PCM than CLOCK-DWF. This is the reason that CLOCK-DWF has a better average memory access time than the proposed hybrid memory at 'lbm'.

As shown in Fig. 5, 'mcf' as well as 'lbm' also improved the performance of average memory access time as the block size increase. However, in 'mcf', the proposed structure improved performance better than CLOCK-DWF unlike 'lbm'. According to simulation result, the proposed structure reduced page write of PCM compared to CLOCK-DWF with 'mcf'. Therefore, 'mcf' has a working–set larger than the suggested memory capacity. So a page fault will occur. Therefore, we confirmed the hybrid memory miss ratio. In 'mcf', the proposed structure achieved performance improvement of miss ratio of about 12% compared with CLOCK-DWF. For these reasons, unlike the results in Fig. 5, the proposed structure achieved an effective performance improvement of the average memory access time with 'mcf' compared to CLOCK-DWF.

In particular, the proposed structure achieves effective performance improvement compared with CLOCK-DWF in 'astar' and 'milc'. Two benchmarks have a shorter working-set than the proposed memory capacity. Therefore, they did not have the page fault. Only the proposed structure achieved the performance improvements by reduced the number of page write of the PCM that is by 70% and 67% compared with CLOCK-DWF.

As a result of the simulation, the proposed structure has a slightly higher miss ratio than CLOCK-DWF. Nonetheless, the proposed structure achieved performance improvements in average memory access time of about 30% compare with CLOCK-DWF.

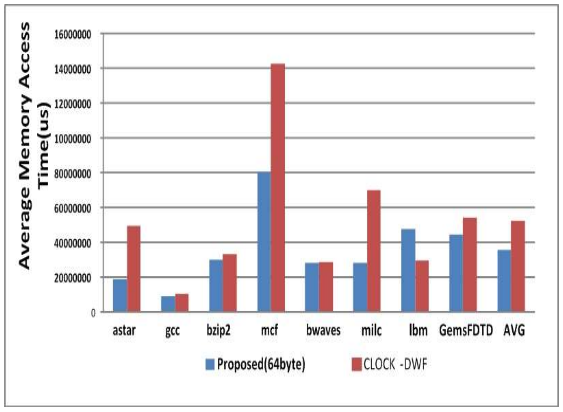

Figure 6: Average memory access time

5. CONCLUSION

In this paper, to achieve efficient page replacement of hybrid memory, we proposed an effective algorithm for DRAM and PCM with the DRAM stored pages requested from the lowest layer memory and pages of the PCM with a temporal locality. The page in which a write operation has occurred at a specific time interval will be stored in the DRAM for a long time. In order to reduce frequent page replacement with DRAM, we have proposed an algorithm to perform write operations according to temporal locality and spatial locality. A page with spatial locality will perform write operations to the PCM. On the other hand, pages with temporal locality will be exchanged with DRAM through the proposed algorithm.

For PCM, block sizes from 64 bytes to 512 bytes are used to select an effective block size for the proposed algorithm and propose additional state bits to select the page of the write operation according to the time interval for the DRAM. As a result, the proposed hybrid memory with block sizes from 64 bytes to 512 bytes displayed better performance than CLOCK-DWF.

6. ACKNOWLEDGMENT

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 (NRF-2014R1A1A4A01008504).

REFRENCES:

[1] S.J. Im, D.K. Shin, "Differentiated space allocation for wear leveling on phase-change memory-based storage device," IEEE Transactions on Consumer Electronics, Vol. 60, Issue 1, 2014, pp: 45-51.

[2] Y.Xie, "Future memory and interconnect technologies," in Proc. Des. Autom. Test Eur. Conf.Exhib., 2013, pp. 964-969.

[3] I.S. Choi, S.I. Jang, "A dynamic adaptive converter and management for PRAM-base main memory," Microprocessors and Microsystems, Vol. 37, Issue 6-7, 2013, pp. 554-561.

[4] S.I. Jang, S.K. Yoon, K. H. Park, "Data Classification Management with its Interfacing Structure for Hybrid SLC/MLC PRAM Main Memory," The Computer Journal, Vol. 58, issue 11, 2015, pp. 2852-2863.

[5] E. K¨ult¨ursay, M. Kandenmir, A. Sivasubramaniam, "Evaluating STT-RAM as an Energy-Efficient Main Memory Alternative," Performance Analysis of Systems and Software (ISPASS), 2013 IEEE International Symposium on.

[6] A. Hassan, H. Vandierendonck, D.S. Nikolopoulos, "Softerware-managed energy-efficient hybrid DRAM/NVM main memory,"

Proceedings of the 12th ACM International Conference on Computing, No. 23, 2015.

[7] Sparsg Mittal, Jeffrey S. Vettr, “A Survey of Software Techniques for Using Non-volatile Memories for Storage and Main Memory Systems," IEEE Transactions on Parallel and Distributed Systems, Vol. 27, Issue 5, 2016, pp. 1537-1550.

[8] B.S. Jung, J.H. Lee, “Analysis on the Effectiveness of the Filter Buffer for Low Power NAND Flash Memory," IEMEK J. Embed. Syst. Appl., Vol. 7, No. 4, 2012, pp. 201-207.

[9] Y. Liu, C. Zhou, X. Cheng, “Hybrid SSD with PCM,” 2011 11th Annual Non-Volatile Memory Technology Symposium Proceeding.

[10] H. C. Seok, Y.W. Park, K.W. Park, K.H. Park, “ Efficient page Caching Algorithm with Prediction and Migration for a Hybrid Main Memory,” ACM SIGAPP Applied Computing Review, Vol. 11, Issue 4, 2011, pp. 38-48.

[11] A.N. Jacobvite, R. Calderbank, D.J. Sorin, “Cost coding to extend the lifetime of memory,” HPCA 2013, 2013 IEEE 79th International Symposium on, 2013.

[12] R. Maddah, et.al, “CAFO: Cost Aware Flip Optimization for Asymmetric Memories,” 2015 IEEE 21st International Symposium on.

[13] M.K. Qureshi, V. Srinivasan, and J. A. Rivers, “Scalable high performance main memory system using phase-change memory technology,” Int’l Symp. Computer Archi. , 2009, pp. 24-33.

[14] K.H. Park, S.K. Yoon, S.D. Kim, “Selective data buffering module for unified hybrid storage system,” IEEE/ACIS 14th International Conference on. Computer and Information Science, 2015.

[15] G. Dhiman, R Ayoub, and T. Rosing, “PDRAM: A Hybrid PRAM and DRAM Main Memory System,” Proceedings of Design Automation Conference, 2009, pp. 664-669.

[16] S. Lee, H. Bahn, and S. H. Noh, “CLOCK-DWF:A Write-History-Aware Page Replacement Algorithm for Hybrid PCM and DRAM Memory Architecture," IEEE Transactions on Computers, Vol. 63. No. 9, 2013, pp. 2187-2200.

[17] M. Lee, D.H. Kang, J. King, “M-CLOCK: Migration-optimized page replacement algorithm for hybrid DRAM and PCM memory architecture,” Proceedings of the ACM Symposium on Applied Computing, 2015.

[18] SpecCPU benchmark, http://www.spec.org/

[19] N. Nethercote and J. Seward, "Valgrind: A Program Supervision Framework“, Elsevier Electronic Notes in Theoretical Computer Science, Vol. 89, No. 2, 2003, pp.44-66.

[20] HD Tune Pro, http://www.hdtune.com