A Superblock-based Memory Adapter Using Decoupled Dual Buffers for Hiding the Access Latency of Non-volatile Memory

Kwang-Su Jung, Jung-Wook Park, Charles C. Weems and Shin-Dug Kim, *Member, IAENG*

***Abstract*— This paper presents a superblock-based memory adapter that allows access transformation between superblock-based pages from the main memory and L2 cache blocks. Recently, non-volatile memories such as PRAM, FeRAM, and MRAM have been considered for use as main memory components because of advantages such as low power consumption, higher density, and non-volatility compared with DRAMs, which have been the dominant main memory component technology. Despite these merits, access latencies of both PRAM and FeRAM are to slow to simply replace DRAMs. The access latency of new memories must be improved before they can be adopted for the main memory. In this paper, we propose a superblock-based adapting buffer located between on-chip cache and main memory. The adapting buffer is comprised of decoupled dual buffers: one is to utilize spatial locality aggressively, and the other exploits temporal locality adaptively. The proposed structure is implemented and evaluated by using a trace-based simulator using SPEC 2000 traces. The experimental results show that the proposed architecture improves buffer miss rate by about 47 percent, compared with the same-sized uniform buffer, making it possible to hide access latency of non-volatile memory effectively.**

***Index Terms*— non-volatile memory, off-chip cache, main memory, buffer management**

I. Introduction

RECENTLY, non-volatile memories such as PRAM, FeRAM, and MRAM have emerged in the commercial area, especially starting with PRAM mass production of 512MB components by Samsung in 2009 [3]. The characteristics of non-volatile memories are shown in Table 1 [2][5][6][7][10]. Although non-volatile memory technology is being developed very quickly, some tradeoffs can be summarized as follows. First, MRAM is the best memory cell compared to other memories. It has lower access latency than that of DRAM, but it cannot replace DRAM because of its high price. For this reason, MRAM is receiving significant attention as a replacement for on-chip SRAM. Both FeRAM and PRAM could potentially replace DRAM within the next decade because these memories have characteristics similar to DRAM [1][4].

TABLE I
CHARACTERISTICS OF NON-VOLATILE MEMORIES AND DRAM

	DRAM	MRAM	FeRAM	PRAM
Non-volatile	X	O	O	O
Read latency	20ns	10~50 ns	30~100 ns	300 ns
Write latency	20ns	10~50 ns	30~100 ns	3us
Endurance	N/A	10^{15}	10^{12}	10^{6}~10^{8}
Density	Unknown	Poor	Poor	Good
Power consumption	300㎽	30㎼	10㎼	30㎼

In general, non-volatile memories have some common advantages compared with DRAM. First, the memories consume lower power. Second, data in the new memory is not lost even after system power is turned off, which means that main memory data doesn't need to be saved to disk when power is turned off, and that significantly affects boot-up. Finally, while DRAM is approaching the density scaling limit, PRAM has higher density and can store more bits per cell.

Use of consumer devices such as smart phones and tablet PCs is increasing rapidly, and requiring higher performance in mobile devices, while also demanding low power consumption to increase battery life. DRAM does not satisfy these needs because it consumes almost 40 percent of the system energy [8]. Thus, research to replace DRAMs with non-volatile memories is under way. Studies of utilizing non-volatile memory as main memory commonly encounter high access latency because latencies of both PRAMs and FeRAMs are significantly greater than DRAMs. In this paper,

Manuscript received July 27th, 2011; revised August 18th, 2011. 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(2011-0002536).

K. S. Jung is a master candidate in Department of Computer Science, Yonsei University, School of Engineering, C532, Shinchon-dong, 134, Seoul 120-749, Republic of Korea. (e-mail: ksjung@cs.yonsei.ac.kr).

J. W. Park is a computer science BK21 Post Doc. in Department of Computer Science, Yonsei University, School of Engineering, C532, Shinchon-dong, 134, Seoul 120-749, Republic of Korea. (e-mail: pjppp@cs.yonsei.ac.kr).

C. C. Weems is an associate professor in Department of Computer Science, University of Massachusetts, Amherst, MA 01003-4610, USA. He is also a codirector of the Architecture and Language Implementation Research Group at the University of Massachusetts. (e-mail: weems@cs.umass.edu).

S. D. Kim is a professor in Department of Computer Science, Yonsei University, School of Engineering, C532, Shinchon-dong, 134, Seoul 120-749, Republic of Korea. (corresponding author phone: 02-2123-2718; e-mail: sdkim@yonsei.ac.kr).

we propose a superblock-based adaptive buffer located between on-chip cache and the main memory. The adaptive buffer is comprised of decoupled dual buffers: one exploits spatial locality aggressively, and the other exploits temporal locality adaptively. The buffers don't have any additional write buffer to hide asymmetric read/write access patterns so that we could effectively reduce the size of buffer space for hiding the access latency of non-volatile memory. The proposed structure is implemented and evaluated by using a trace-based simulator with SPEC 2000 traces.

According to our simulation results, miss rate can be reduced by around 47 percent compared to a uniform buffer of the same size. Therefore, the proposed superblock-based memory adapter can effectively hide the access latency of non-volatile memory using only small, decoupled, dual buffers.

The rest of this paper is organized as follows: Section 2 presents the related work to adapt non-volatile memory for main memory using buffer structures to reduce access latency. Section 3 presents the superblock-based memory adapter structure and its management algorithm. In Section 4, performance analysis of this system is shown. Finally, we conclude in Section 5.

II. Related Work

There are many studies regarding use of non-volatile memories in the existing memory hierarchy; disk cache and write buffers over NAND flash memory [13], hybrid main memory systems [7], and homogenous non-volatile memory structures [9][12][14]. In this section, new memory architectures using PRAM will be reviewed briefly along with work on using off-chip cache memory to reduce the miss rate.

PRAM, which is commercially available, is being variously applied to existing memory systems ranging from embedded systems to high-end computing systems. Ferreira proposed a new memory system called PMMA to effectively use PRAM for main memory in next-generation embedded systems [9]. In this architecture, PRAM is used both as a storage medium and as a main memory component. Because the density of PRAM is considerably higher than that of DRAM, it can be used as a backing store in many embedded computing applications. But, the access latency of PRAM is significantly slower than DRAM. Thus, Acceleration and Endurance Buffer (AEB), located between PRAM and CPU is proposed to reduce the access latency of PRAM. The AEB performs the role of a write buffer and high speed data cache with respect to the PRAM. The AEB consists of a page cache area to hold application data at page granularity and a spare table to store metadata. Because of capacity and energy considerations, however, the AEB uses DRAM rather than SRAM, and uses a large amount of memory (4Gbits).

Qureshi proposed a PRAM-based hybrid main memory system coupled with a small DRAM buffer [7]. In this architecture, the DRAM buffer acts as a cache with respect to the PRAM to hide the read latency. A write buffer is used to hide the write latency. In addition to the write buffer, fine-grained wear-leveling is proposed the extend the endurance limits of PRAM, which employs a line size write unit. The PRAM-based hybrid main memory architecture is composed 32 Gbyte PRAM with a 1Gbyte DRAM buffer. Their hybrid main memory shows results similar to a 32Gbyte DRAM.

Beyond architectures using PRAM as the basic component of main memory, there is relevant research in reducing the access latency of main memory with off-chip cache. Zhang proposed a new memory hierarchy that uses cached DRAM to construct a large, low overhead off-chip cache [11]. The cached DRAM is composed of a large DRAM space to hold large working sets and a small SRAM space to exploit the spatial locality in L2 miss streams to reduce access latency. The cache DRAM consists of 64 Mbytes of DRAM with 128Kbytes of on-chip cache, and its performance result offers better execution time than an 8Mbyte L3 off-chip cache alone.

Drawing from both the use of buffers to hide PRAM access latency and the use of both large DRAM and small SRAM in caches, we propose the use of small, decoupled dual buffers for hiding the access latency of non-volatile memory while minimizing miss rate as much as possible. The studies using PRAM as a main memory component and an off-chip cache with low overhead have required a large amount of buffer space to attain performance comparable to DRAM, while our proposal achieves this with two, much smaller, buffers.

III. Superblock-based Memory Adapter Architecture

A. Overall Architecture

Conventional memory hierarchies are typically constructed as L1 and L2 caches, main memory, and disk storage. When a cache miss occurs at L2, access proceeds to main memory. As explained before, our goal is to replace volatile DRAMs with new, non-volatile technologies for main memories. If these memories are to be used as main memory components, their access latency should be less than or equal to that of DRAM. Furthermore, because PRAM has asymmetrical access latency, a write buffer must be used to its long write time. To address these issues, we present a superblock-based memory adapter. As much as possible, the adapter should maintain the data that L2 needs to get from main memory on a cache miss. One approach to achieving this is to have the superblock-based memory adapter aggressively prefetch a set of pages, called a superblock, from the main memory. It then classifies the fetched data as valid or invalid. In addition to this basic technique to reduce low access latency of non-volatile memory, where others have used homogeneous buffers, we use heterogeneous buffers that emphasize spatial and temporal locality: a spatial locality based superblock buffer (SLSB) and a temporal locality based adaptive buffer (TLAB). We exploit the TLAB as a write buffer by decoupling the buffers in the proposed adapter. The sub-blocks that are accessed once before being evicted from the SLSB are promoted into the TLAB. For write requests to the SLSB, the sub-blocks are moved to the TLAB, where they can hide the write latency of the slower memory medium while increasing the probability of a hit on access by L2 cache.

Figure 1 shows the superblock-based memory adapter architecture in the context of a memory hierachy. The architecture is constructed as follows:

ISBN: 978-988-19251-7-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

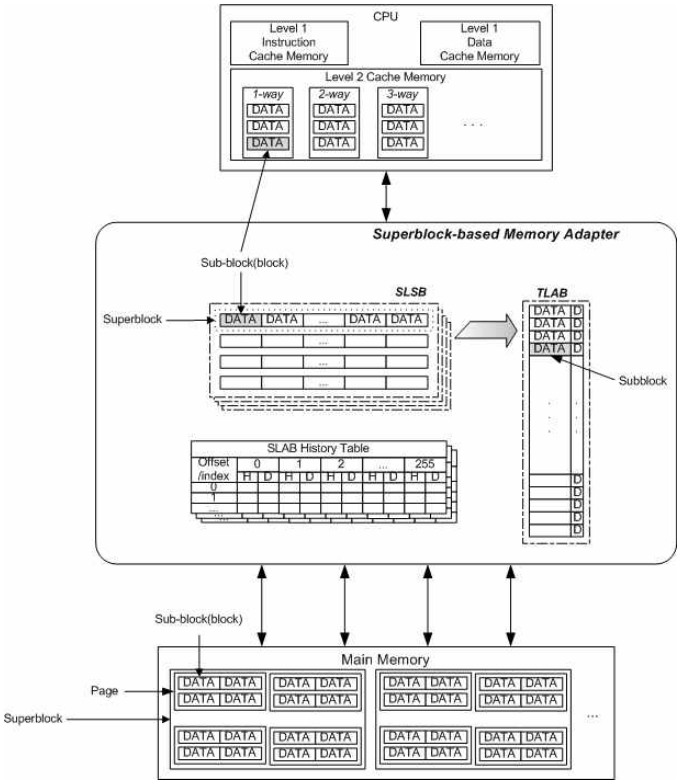

Fig. 1. Superblock-based memory adapter architecture

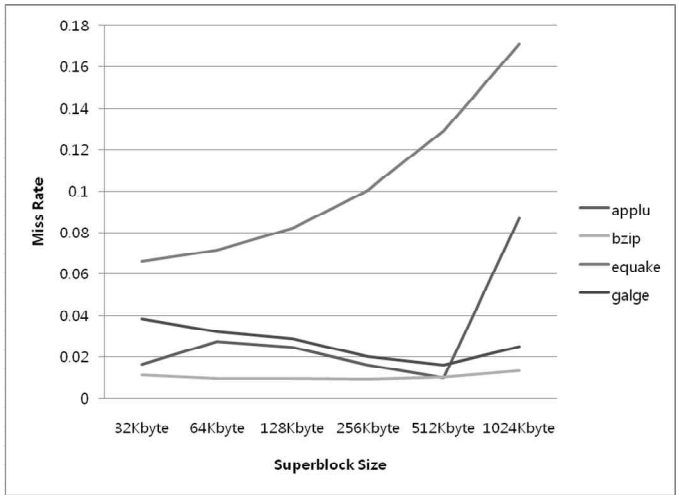

Fig. 2. Miss rate in terms of superblock size composition

- ***Sub-block, superblock, page :*** As shown in Figure 1, a sub-block is a basic data unit managed by this adapter and is the same size as an L2 cache block. We set the size of a sub-block to the typical width of 128Bytes as a test example. A page is the minimal unit of memory logically transferred between main memory and disk by the operating system in supporting virtual memory, and consists of a group of sub-blocks. A superblock is defined as a unit of data formed from a set of pages transferred between the proposed superblock memory adapter and the non-volatile main memory.
- ***Decoupled Dual Buffers :*** As mentioned previously, the proposed memory adapter has two buffers. The first buffer an SLSB that fetches superblocks from the main memory aggressively. The second is a TLAB that manages useful data extracted from the SLSB. Because the superblock-based memory adapter fetches a group of pages at once, these pages may contain reusable data and/or unnecessary data. Because of temporal locality, reusable data tend to be accessed again from L2 cache. Unnecessary data, on the other hand, tend to be flushed to main memory. The size of a sub-block is the same as an L2 cache block, and a superblock holds multiple sub-blocks. Basically, superblocks are fetched by and managed in the SLSB. The TLAB, on the other hand, deals with accesses based on sub-block units.
- ***SLSB History Tables :*** When a miss occurs in the proposed adapter, a superblock must be evicted from the SLAB. Data within the superblock will be either reusable or unnecessary, as determined from the SLSB history table. The table also manages a hit bit and a dirty bit associated with each sub-block in the SLSB. If the hit bit and/or the dirty bit is set, it means that access from a higher memory layer has happened. Therefore, the accessed sub-blocks will be moved to the TLAB upon eviction from the SLSB. Each entry in the TLAB also has a dirty bit to identify sub-blocks to be stored in the next lower memory layer when the sub-block is finally evicted from the TLAB. Because a superblock consists of a few sub-blocks, the SLSB history table also maintains a hit count for each of the sub-block units.
- ***Victim algorithm :*** The eviction algorithm applied to the SLSB is least recently used (LRU). Based on LRU status, the SLSB evicts a superblock among the existing superblocks in the SLSB. In contrast, the TLAB uses a first-in-first-out policy for eviction. When a miss occurs in the proposed adapter, sub-blocks that are valid according to the SLSB history table are moved into the TLAB in a queuing fashion.

B. Superblock Size

Determining the number of pages to be fetched as a unit from main memory cannot be determined by theoretical calculation. Thus, we empirically determined the proper size of a superblock based on a set of traces: applu, bzip, equake, and galge. Based on preliminary analysis we constrained the range of superblock sizes from 32Kbytes to 1024Kbytes. Figure 2 shows the experimental results used to select an optimized SLSB superblock size. The x-axis corresponds to various sizes, while the y-axis gives the miss rate for each size. From the figure, it is clear that there is no single best superblock size because the access patterns of the benchmarks are quite different. The miss rate of equake simply increases with increasing superblock size. Whereas, the miss rate of applu is irregular, and the others are stable or are not seriously affected by superblock size. From these results, we choose 32Kbytes as a good compromise for the SLAB superblock size.

C. Superblock-based Memory Adapter Operation

When a data request comes from L2 cache, the superblock-based memory adaptor begins its operation.

Its basic operational flow is as follows:

- ***Step 1.*** First, a data request is generated by L2 cache

ISBN: 978-988-19251-7-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

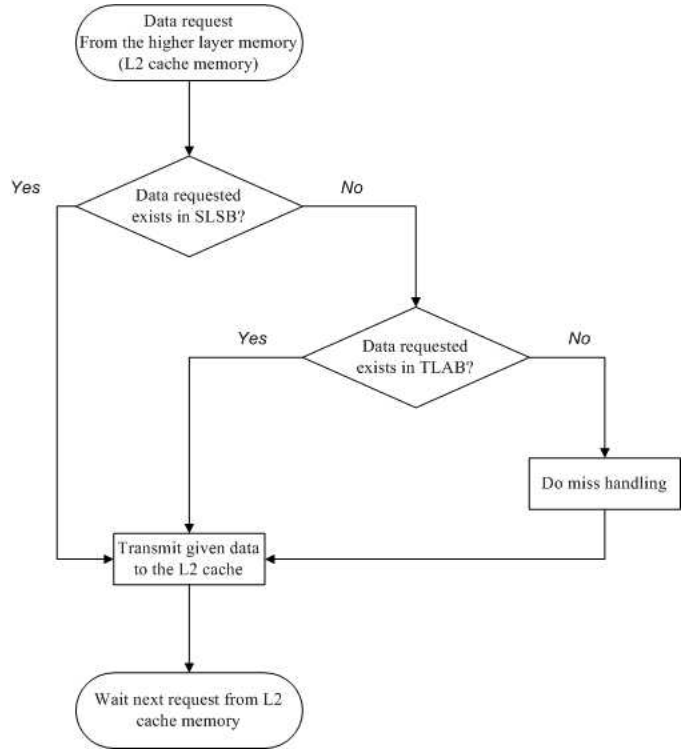

Fig. 3. Superblock-based memory adapter operation

according to the type of miss.

- ***Step 2.*** Check whether the requested data is in the SLSB. If the requested data is in the SLSB, go to step 5. Otherwise go to step 3.
- ***Step 3.*** Check if the data is in the TLAB. If the requested data is in the TLAB, go to step 5. Otherwise, a miss has occurred, and we go to the next step.
- ***Step 4.*** The LRU block will be evicted. The adapter identifies sub-blocks in the victim superblock that have been accessed from L2 while in the SLSB and transfers them to the TLAB. Other blocks are just discarded. The evicted space is replaced with the requested data from main memory.
- ***Step 5.*** The memory adapter transmits the requested data to the L2 cache.

Figure 3 shows this process.

When a miss occurs in the proposed memory adapter, we should determine a superblock to be evicted from SLSB and then valid sub-blocks are moved to TLAB. In this process, the proposed superblock-based memory adapter takes advantage of temporal locality in the TLAB. To do so, we extract sub-blocks from a flushed superblock depending on records in the SLSB history table.

Detailed miss handling operation is shown as follows:

- ***Step 1.*** The adapter selects the LRU superblock in the SLSB from among the associative set of superblocks.
- ***Step 2.*** The adapter classifies sub-blocks in a victim superblock as valid or invalid according to their hit counts in the SLAB history table. A valid sub-block has been accessed from L2 cache at least once. Invalid sub-blocks are discarded because a clean copy exists in main memory. Valid, sub-blocks are moved to the TLAB. There, the sub-block can be accessed again from the L2 cache. If it is changed while in the TLAB, then the TLAB plays a role similar to a write buffer.

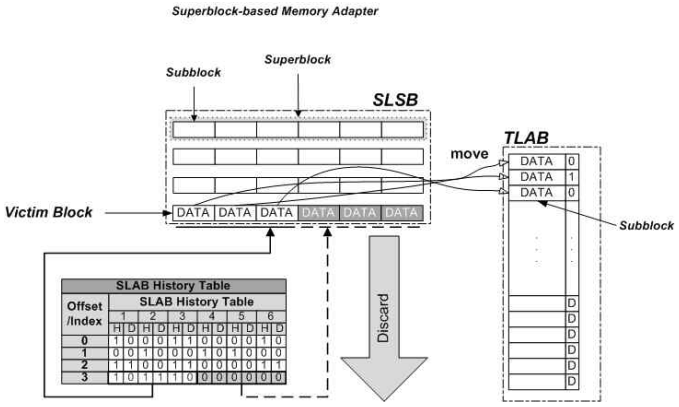

Fig. 4. Miss handling process in superblock-based memory adapter

- ***Step 3.*** The victim block is replaced with a new superblock brought from a lower memory layer.

Figure 4 shows these processes.

IV. Experimental Evaluation

The simulator for the proposed superblock-based memory adapter was developed to evaluate the miss rate between L2 cache and main memory. We use SimpleScalar 3.0 [15] and a subset of the SPEC 2000 benchmark. SimpleScalar's L1 instruction cache and L1 data cache consist of 32KBytes, 4-way set associative, with a 64 Byte block size. L2 is configured as a unified 1MByte, 8-way set associative cache, with a 128 Byte block size. We extracted the traces used as input to the simulator for the proposed adapter from the L2 miss routine in SimpleScalar.

Three size configurations for the proposed memory adapter are considered: 1MBytes for SLSB plus 8MBytes for TLAB, 8MBytes for SLSB plus 1MBytes for TLAB, and 4MBytes for SLSB plus 16MBytes for TLAB, with 8-way set associativity. We also implemented a simple unified buffer simulator to compare against the proposed adapter. The unified buffer simulator fetches superblocks that are the same size as those in the proposed memory adapter.

A. Reducing Miss Rate

To evaluate the performance of the proposed superblock-based memory adapter, we cmpare it with a unified buffer having the same space and twice the total space of the largest configuration of the proposed adapter.

Figure 5 shows the miss rate of each buffer structure. The lowest miss rate is for the configuration with 4MBytes of SLSB plus 16MBytes of TLAB, with an average of about 0.49 percent. Compared with the unified 40MBytes buffer structure, the proposed superblock-based memory adapter shows an average improvement of about 7.5 percent lower miss rate even though it has half as much storage space overall..

Furthermore, the miss rate of the proposed superblock-based memory adapter is about 49.1 percent lower than that of the same-sized unified buffer. Another configuration for the proposed superblock-based memory adapter, 1Mbytes of SLSB and 8Mbytes of TLAB, has an average miss rate of 1.29 percent, while the last configuration, 8Mbytes of SLSB and 1Mbytes of TLAB, shows an average miss rate of 1.01 percent.

ISBN: 978-988-19251-7-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

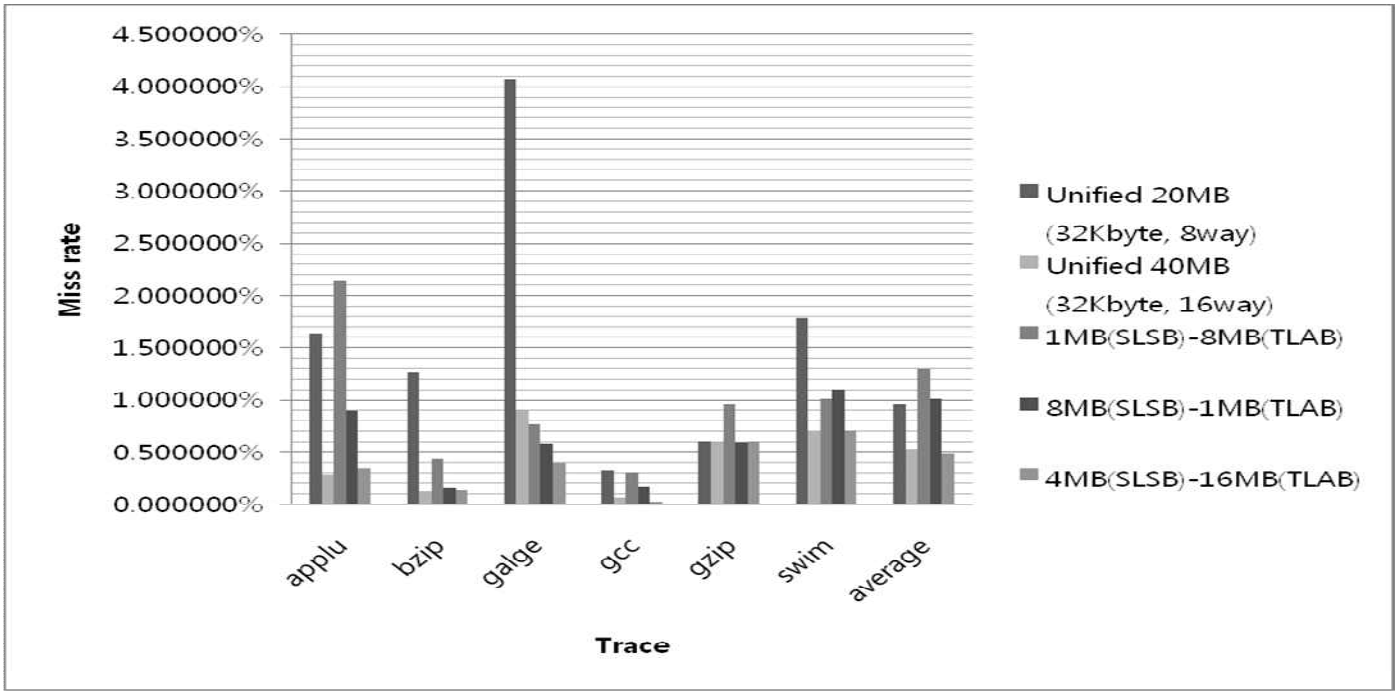

Fig. 5. Miss rate of each buffer configuration

B. *Decoupled Dual Buffers Hit Rate Portion*

We also analyzed the hit rate portion for each of the buffers in the proposed adapter, for the configuration with 4MBytes for SLSB and 16MBytes for TLAB. Figure 6 shows this result. On average, the hit rate portion for the SLSB is about 58 percent, while the hit rate fraction for TLAB is about 42 percent. The traces for bzip, galge, have a high hit rate in the TLAB, indicating greater temporal locality. The others have a more balanced hit rate distribution. On the other hand, the gzip trace takes greater advantage of the spatial locality in the SLSB. This illustrates how the proposed memory adapter is able to adapt to different locality mixes in disparate applications.

V. Conclusion

In this paper, we proposed a superblock-based memory adapter that can reduce access latency from L2 cache memory to a high latency main memory such as PRAM. By using a pair of decoupled buffers, an effective compromise between temporal and spatial locality can be achieved with a relatively small amount of overall buffer space. Also, the proposed adapter can conceal asymmetric read/write access speed without the need for a separate write buffer. The experimental results show that the proposed architecture outperforms a unified buffer of the same size by about 47 percent in terms of buffer miss rate, making it possible to more effectively hide access the latency of a non-volatile memory, without resorting to the larger buffer sizes of prior studies.

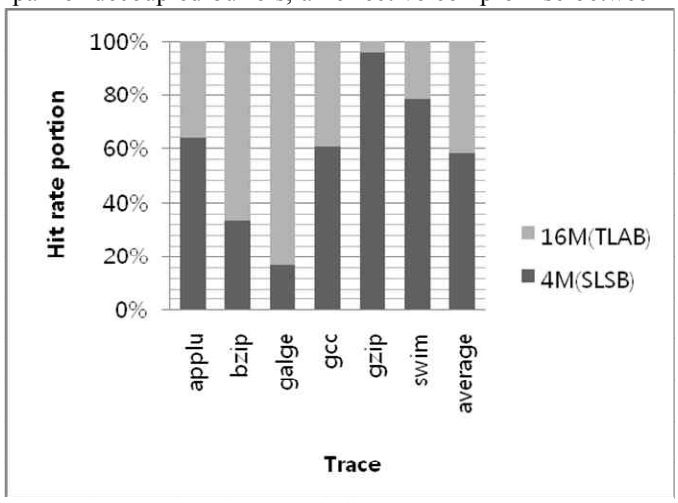

Fig. 6. Hit rate portion of 4M-16M buffer structure

References

[1] R. F. Freitas and W.W. Wilcke, "Storage-class memory: The next storage system technology", *IBM Journal of Research and Development*, Vol.52, Issue:4.5, pp.439-447, July 2008.

[2] NcPRAM, http://www.samsung.com/global/business/semiconductor/products/fusionmemory/Products_NcPRAM.html

[3] Samsung PRAM memory, http://www.photokinashow.com/0369/samsung/storage/flashmemorycard/

[4] Samsung CEO: Headwinds hindeer PRAM, http://www.eetimes.com/electronics-news/4213362/Samsung-CEO--Headwinds-hinder-PRAM-

[5] FRAM Datasheet, http://www.ramtron.com/lib/literature/datasheets/FM24C512ds_r1.0.pdf.

[6] MRAM Datasheet, http://www.freescale.com/files/microcontrollers/doc/data_sheet/MR2A16A.pdf.

[7] M.K. Qureshi, V. Srinivasan, and J.A. Rivers, "Scalable High Performance Main Memory System Using Phase-Change Memory Technology", *ACM SIGARCH Computer Architecture News*, Vol.37, Issue 3, pp.24-33, June 2009.

[8] C.Lefurgy, K.Rajamani, F.Rawson, W.Felter, M.Kistler, and T.W.Keller. "Energy management for commercial servers", *IEEE Computer*, 36(12):39-48, Dec. 2003.

[9] A.P. Ferreira, B. Childers, R. Melhem and D. Mosse, "Using PCM in Next-generation Embedded Space Applications", *Real-Time and Embedded Technology and Applications Symposium*, pp.153-162, April 2010

[10] 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, Issue:4.5, pp.449-464, July 2008

ISBN: 978-988-19251-7-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

[11] Z. Zhang, Z. Zhu, and X. Zhang, "Design and Optimization of Large Size and Low Overhead Off-Chip Caches", *Computers, IEEE Transactions on*, Vol.53, Issue:7, pp. 843-855, July 2004.

[12] A. Seznec, "A Phase Change Memory as a Secure Main Memory", *Computer Architecture Letters*, Vol.9, Issue:1, pp.5-8, January 2010.

[13] C.W. Smullen, J. Coffman, and S. Gurumurthi, "Accelerating Enterprise Solid-State Disks with Non-Volatile Merge Caching", *Green Computing Conference, 2010 International*, pp.203-214, August 2010.

[14] M.K. Qureshi, M.M. Franceschini, and L.A. Lastras-Montano, "Improving Read Performance of Phase Change Memories via Write Cancellation and Write Pausing", *High Performance Computer Architecture, 2010 IEEE 16th International Symposium on*, pp.1-11, January 2010.

[15] SimpleScalar, http://www.simplescalar.org/, SPEC2000 binaries, 2011.

ISBN: 978-988-19251-7-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)