Energy-Efficient Cached DIMM Architecture

Mu-Tien Chang, Joe Gross, and Bruce Jacob
Dept. ECE, University of Maryland

***Abstract*—This paper presents a cached DIMM architecture – a low-latency and energy-efficient memory system. Two techniques are proposed: the on-DIMM cache and the on-DIMM cache-aware address mapping scheme. These two techniques work together to reduce the memory access latency. Based on the benchmarks considered, our experiments show that compared to a conventional DRAM main memory, the proposed architecture reduces memory access latency by up to 30% (25% on average), reduces system execution time by up to 25% (10% on average), achieves up to 12% energy savings (5% on average), and improves the energy delay product by up to 27% (14% on average).**

I. Introduction

DRAM (dynamic random access memory) continues to be the primary technology for main memories due to its high capacity, fast access time, infinite endurance, and high reliability features [14]. However, the latency gap between CPU and DRAM has been increasing over time. Moreover, energy efficiency has become an important concern for modern computer systems. In particular, memory is one of the main sources of the system power dissipation. For instance, [6] [5] [13] show that 15% to 40% of a server power consumption is due to the memory system.

This paper proposes techniques to reduce the latency and the energy consumption of the main memory system. We make the following contributions:

- We propose a low-latency energy-efficient cached DIMM architecture, which is a DRAM main memory with SRAM caches integrated on the DIMMs. The proposed architecture thus provide expandable cache capacity without modifications to the DRAM devices.
- An on-DIMM cache aware address mapping scheme is proposed to enable higher on-DIMM cache hit ratio.
- We show that the proposed architecture reduces memory access latency by up to 30% (25% on average), reduces system execution time by up to 25% (10% on average), and achieves up to 12% main memory energy savings (5% on average), compared to a conventional main memory without on-DIMM caches.

The remainder of the paper is organized as follows. Section 2 provides the related work. Section 3 describes the proposed architecture. Section 4 presents our experimental methodology. Section 5 shows the simulation results. Finally, section 6 concludes this paper.

II. Related Work

Cached DRAM (CDRAM) was first manufactured by Mitsubishi for use as a texture buffer in high-end graphics cards with the goal of inexpensively bridging the growing performance gap between the CPU and DRAM [10]. The idea is to place a small SRAM next to the DRAM core. The SRAM performs like an on-memory cache, which reduces the average memory access latency if it has good hit ratio. A variety of DRAM architectures employ on-memory caches [10] [11] [12] [16] [8] [2]. However, all the prior cached DRAM architectures require modifications to the DRAM device, which is highly discouraged in the DRAM community. The proposed cached DIMM integrates SRAMs on the DIMM. No changes are made to the DRAM devices. Moreover, we propose an address mapping scheme that is essential for on-memory caches to work effectively.

III. Cached DIMM Architecture

A. The On-DIMM Cache

The on-DIMM cache acts like a last-level cache. However, its capacity expands with the number of DIMMs. Therefore, the cache capacity can be increased without replacing the CPU. Since power density has become a problem for high-performance processors, the proposed scheme allows increases in cache capacity without increasing the processor power usage.

The on-DIMM SRAM cache hit ratio determines the performance of the cached DIMM architecture: if the hit ratio is low, in addition to the extra step of checking the cache tag, most requests still require long access to DRAM devices. Moreover, due to the power overhead introduced by the on-DIMM cache, the energy consumption of a cached DIMM is higher than conventional DIMMs without on-memory caches if the system execution time is not reduced. Therefore, our design focuses on improving the on-DIMM cache hit ratio.

Conventional cached DRAM architectures place the on-memory cache at the device granularity. However, in addition to the need for DRAM device modification, a set of multiple split small caches have higher miss ratio than a unified large cache in general [9]. Therefore, we propose a cached DIMM architecture that has a cache at a coarser granularity – instead of placing a small SRAM cache next to each of the DRAM arrays, we integrate a large cache on the DIMM, as illustrated in Figure 1. When a new memory request arrives, the on-DIMM cache tag are checked before accessing the DRAM. If the on-DIMM cache hits, the requested data is returned immediately without accessing the DRAM. If the on-DIMM cache misses, the DRAM row is activated and the DRAM sends the requested data to the on-DIMM cache. The on-DIMM cache then returns the data to the CPU. A on-DIMM cache hit therefore results in shorter memory access latency.

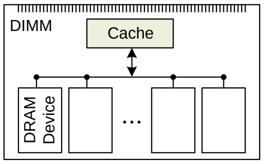
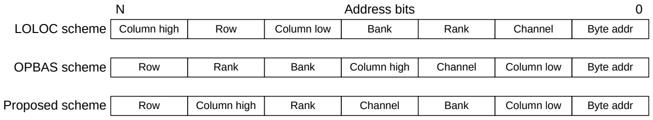


Fig. 1. Proposed cached DIMM architecture.



	N			Address bits			0
LOLOC scheme	Column high	Row	Column low	Bank	Rank	Channel	Byte addr
OPBAS scheme	Row	Rank	Bank	Column high	Channel	Column low	Byte addr
Proposed scheme	Row	Column high	Rank	Channel	Bank	Column low	Byte addr

Fig. 2. Address mapping schemes. LOLAC: SDRAM low locality address mapping scheme. OPBAS: SDRAM high performance address mapping scheme. The proposed scheme is the on-DIMM cache aware address mapping scheme.

It also results in lower dynamic energy since the DRAM does not need to be activated.

B. An On-DIMM Cache-Aware Address Mapping Scheme

The memory controller employs an address mapping scheme to determine the mapping between a memory request and the DRAM physical location. A good address mapping scheme for cached DRAM architectures enables good on-memory cache hit ratio while maintaining memory interleaving to preserve bandwidth. In addition to the address mapping schemes provided in [7], we propose an on-DIMM cache-aware address mapping scheme, as shown in Figure 2. The proposed scheme enables better on-DIMM cache hit ration while maintaining memory interleaving.

IV. Experimental Methodology

A. Configurations

For our study, we modify DRAMSimII [7], a detailed cycle-accurate memory system simulator. DRAMSimII is integrated with M5 [4], a full-system simulator. The baseline configuration is a 3.2 GHz quad-core system with cache organization similar to the Intel Core i7.

A 400 MHz DDR SDRAM memory system of 4 GB capacity (1 GB per DIMM) is considered, organized as 2 channels, 2 DIMMs per channel, 2 ranks per DIMM, and 16 banks per rank. The timing and power parameters are modeled based on Micron's specification. Moreover, a cache module is integrated into DRAMSimII to simulate the proposed architecture. The configuration for the on-DIMM cache is 8 MB (per DIMM), 128 B block size, 8-way associative. Additionally, we use CACTI [15] to calculate the performance and power of the on-memory caches.

B. Benchmarks

We use six benchmarks to evaluate our system. They are mcf, milc, lbm, libquantum from the SPEC CPU 2006 suite [1], and canneal, ferret from the PARSEC 2.1 suite [3]. We use the input sets `ref` and `medium` for the SPEC and PARSEC benchmarks, respectively. These benchmarks are chosen because the are more memory intensive.

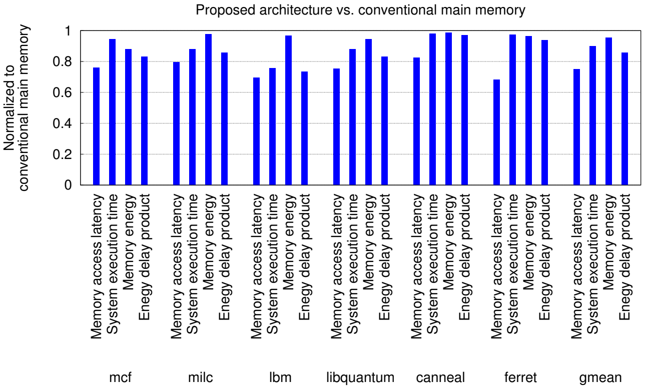


Fig. 3. Performance results of the proposed architecture normalized to a conventional main memory without on-DIMM caches.

V. Results

For each of the benchmarks, we compare the performance of the proposed architecture and a conventional main memory system. The conventional main memory has the same capacity and organization as the proposed architecture, but it does not have the on-DIMM cache integrated. We evaluate the average memory access latency, the system execution time, the energy consumed by the main memory and the on-DIMM cache, and the energy delay product, as shown in Figure 3. The *gmean* is the geometric mean of the six evaluated benchmarks. The proposed architecture reduces average memory access latency by 17% to 31%, which is due to to the fast data response of the on-DIMM caches. The reduction of average memory access latency translates to better system execution time, where 2% to 24% execution time reduction is observed. Note that the reduction of execution time is affected by both the reduction of the memory access latency and the number of requests going to the memory system. Therefore, even if we see significant memory access latency improvement, if the number of memory requests is not that many, the execution time is only slightly improved. Moreover, although the on-DIMM cache introduces both active and standby power overhead, the execution time reduction compensates the on-DIMM cache power overhead. As a result, the proposed architecture reduces the energy of the memory system by 1% to 12%. Finally, the energy delay product is reduced by 4% to 27%.

VI. Conclusions

In this paper, we present a cached DIMM architecture that places an SRAM cache on a DIMM to improve the latency and energy consumption of memory systems. The capacity of cache thus scales with the number of DIMMs, where no modifications are required to the DRAM devices. In addition, an on-DIMM cache-aware address mapping scheme is proposed to increase the on-DIMM cache hit ratio. We show that the proposed architecture effectively reduces the memory access latency, the system execution time, and the energy consumption.

References

[1] SPEC CPU 2006.

[2] N. AbouGhazaleh, B. R. Childers, D. Mossé, and R. G. Melhem. Near-Memory Caching for Improved Energy Consumption. *IEEE Trans. Comput.*, 56:1441–1455, November 2007.

[3] Christian Bienia. *Benchmarking Modern Multiprocessors*. PhD thesis, Princeton University, January 2011.

[4] N. L. Binkert, R. G. Dreslinski, L. R. Hsu, K. T. Lim, A. G. Saidi, and S. K. Reinhardt. The M5 Simulator: Modeling Networked Systems. *IEEE Micro*, 26:52–60, July 2006.

[5] D. Economou, S. Rivoire, and C. Kozyrakis. Full-System Power Analysis and Modeling for Server Environments. In *Workshop on Modeling Benchmarking and Simulation (MOBS)*, 2006.

[6] X. Fan, W.-D. Weber, and L. A. Barroso. Power Provisioning for a Warehouse-Sized Computer. In *ISCA*, 2007.

[7] J. Gross. High-performance DRAM system design constraints and considerations. Master's thesis, University of Maryland, College Park, August 2010.

[8] A. Hegde, N. Vijaykrishnan, M. Kandemir, and M. J. Irwin. VL-CDRAM: variable line sized cached DRAMs. In *Proceedings of the 1st IEEE/ACM/IFIP international conference on Hardware/software codesign and system synthesis*, CODES+ISSS '03, pages 132–137, New York, NY, USA, 2003. ACM.

[9] J. L. Hennessy and D. A. Patterson. *Computer Architecture: A Quantitative Approach, 4th Edition*. Morgan Kaufmann, 2006.

[10] H. Hidaka, Y. Matsuda, M. Asakura, and K. Fujishima. The Cache DRAM Architecture: A DRAM with an On-Chip Cache Memory. *IEEE Micro*, 10:14–25, March 1990.

[11] W.-C. Hsu and J. E. Smith. Performance of cached DRAM organizations in vector supercomputers. In *Proceedings of the 20th annual international symposium on Computer architecture*, ISCA '93, pages 327–336, New York, NY, USA, 1993. ACM.

[12] R. P. Koganti and G. Kedem. WCDRAM: a fully associative integrated cached-DRAM with wide cache lines. In *Proceedings of the 11th Annual International Symposium on High Performance Computing Systems*, HPCS '97. IEEE, 1997.

[13] C. Lefurgy, K. Rajamani, F. Rawson, W. Felter, M. Kistler, and T. Keller. Energy Management for Commercial Servers. *IEEE Computer*, 36:39–48, Dec. 2003.

[14] S. Natarajan, S. Chung, L. Paris, and A. Keshavarzi. Searching for the Dream Embedded Memory. *IEEE Solid-State Circuits Magazine*, 1(3):34–44, Summer 2009.

[15] S. Thoziyoor, J. H. Ahn, M. Monchiero, J. Brockman, and N. P. Jouppi. A Comprehensive Memory Modeling Tool and Its Application to the Design and Analysis of Future Memory Hierarchies. In *ISCA*, 2008.

[16] Z. Zhang, Z. Zhu, and X. Zhang. Cached DRAM for ILP Processor Memory Access Latency Reduction. *IEEE Micro*, 21:22–32, July 2001.