Poster: Revisiting Virtual Channel Memory for Performance and Fairness on Multi-core Architecture

Licheng Chen[1,2], Yongbing Huang[1,2], Yungang Bao[1], Onur Mutlu[3], Guangming Tan[1], Mingyu Chen[1]
[1]Key Laboratory of Computer System & Architecture, Institute of Computing Technology, Chinese Academy of Sciences
[2]Graduate School of Chinese Academy of Sciences
[3] Carnegie Mellon University
{chenlicheng, baoyg, cmy}@ict.ac.cn {huangyongbing, tgm}@ncic.ac.cn onur@cmu.edu

Categories and Subject Descriptors: B.3.2, shared memory; B.3.1, dynamic memory (DRAM)
General Terms: Experimentation, Measurement, Performance

1. Introduction

In modern multi-core chip architecture, the DRAM system is shared by more and more cores and high bandwidth I/O devices. This trend raises the memory contention problem and the memory QoS problem. Meanwhile, we find that multicore architecture also brings a large amount of unexploited memory-level parallelism (MLP). In order to exploit the MLP, we revisit the obsolete Virtual Channel Memory (VCM) technology [3]. The experimental results show that VCM is a good alternative to traditional DRAM chip on multicore architecture because it can not only improve performance but also reduce unfairness. Thus, we suggest memory chip vendors reconsider the VCM technology for multicore architecture.

2. Implementing VCM on a multi-core architecture

Figure 1 illustrates VCM's conceptual organization. Two commands, *Prefetch* and *Restore*, are introduced to transfer data between row buffers and channel buffers, each row buffer is divided into 4~16 segments which is the basic transfer data size. Foreground operations and background operations can also be executed independently, so VCM can exploit more MLP than traditional DRAM chip. Figure 2 illustrates the VCM memory controller, the memory request buffers are distributed as a number of separated channel request buffers and the physical memory address is translated into VCM address in the form of <*bank*, *row*, *segment*, *column*>. Based on this, we further implement the state of the art scheduling algorithms on VCM, including FR-FCFS [4], PARBS [2], and ATLAS [1].

3. Experimental Results

Figure 3 shows the normalized IPC (NIPC) speedups of memory intensive applications running on a 16-core system with 8 memory banks. On average, VCM with 32 1KB-channel buffers achieves 2.08X performance speedup against the baseline FR-FCFS system, while eight 8KB-row buffers provide only 1.15X improvements.

Figure 4 shows the weighted speedup (system throughput) of heterogeneous memory intensive workloads on a 16-core system. For VCM with FR-FCFS, it improves system throughput by 1.66X. Since applying PAR-BS and ATLAS to VCM changes system throughput slightly, by only 5.9% and 0.03% respectively. This means that VCM is inherently able to eliminate unfairness and improve system throughput.

From our perspective, we suggest memory chip vendors reconsider the VCM technology for multicore architecture.

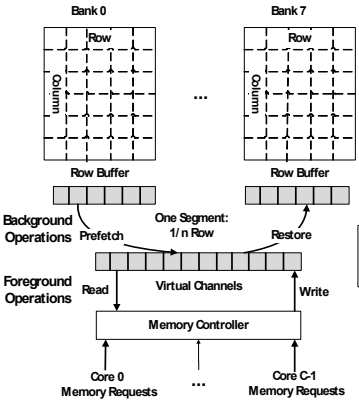

Figure 1. Organization of VCM

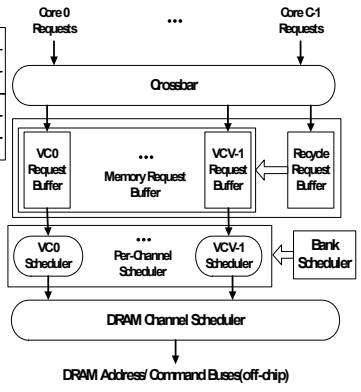

Figure 2. VCM Memory Controller

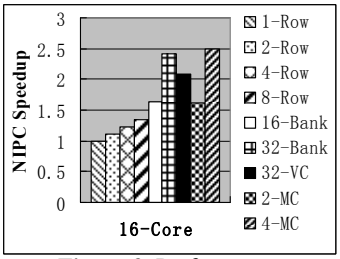

Figure 3. Performance improvements

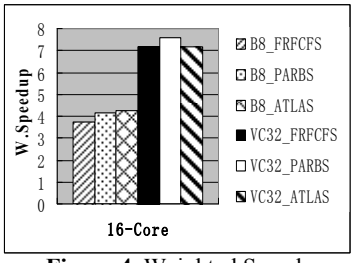

Figure 4. Weighted Speedup

References

[1] Y. Kim, D. Han, O. Mutlu, and M. Harchol-Balter. ATLAS: A Scalable and High-Performance Scheduling Algorithm for Multiple Memory Controllers. in Proceedings of the 16th International Symposium on High-Performance Computer Architecture (HPCA). 2010.

[2] O. Mutlu and T. Moscibroda, Parallelism-Aware Batch Scheduling: Enhancing both Performance and Fairness of Shared DRAM Systems, in Proceedings of the 35th Annual International Symposium on Computer Architecture, 2008.

[3] Nec, 64M-bit Virtual Channel SDRAM data sheet, 1998.

[4] S. Rixner, W. J. Dally, U. J. Kapasi, P. Mattson, and J. D. Owens. Memory Access Scheduling. in Proceedings of the 27th annual international symposium on Computer architecture. 2000.

ICS'11, May 31-June 4, 2011, Tucson, Arizona, USA.
ACM 978-1-4503-0102-2/11/05.