PS-Dir: A Scalable Two-Level Directory Cache

Joan J. Valls*, Alberto Ros†, Julio Sahuquillo*, María E. Gómez*, and José Duato*

*Department of Computer Engineering
Universitat Politècnica de València (Spain)
joavalmo@fiv.upv.es, {jsahuqui,megomez}@disca.upv.es

† Dept. de Ingeniería y Tecnología de Computadores
Universidad de Murcia (Spain)
aros@ditec.um.es

ABSTRACT

As the number of cores increases in both incoming and future chip multiprocessors, coherence protocols must address novel hardware structures in order to scale in terms of performance, power, and area. It is well known that most blocks accessed by parallel applications are private (i.e., accessed by a single core). These blocks present different directory requirements and behavior than shared blocks. Based on this fact, this paper proposes a two-level directory cache that tracks shared blocks in a small and fast first-level cache and private blocks in a larger and slower second-level cache, namely Shared and Private caches, respectively. Speed and area reasons suggest the use of eDRAM technology much dense but slower than SRAM technology for the Private cache, which in turn brings energy savings. Experimental results for a 16-core system show improvements in performance by 11.1%, in area by 25.4%, and in energy consumption by 20.5% compared to a conventional directory cache.

Categories and Subject Descriptors

B.3.2 [**Memory Structures**]: Design Styles—*Cache memories*; C.1.2 [**Processor Architectures**]: Multiple Data Stream Architectures (Multiprocessors)

Keywords: Multicore, cache coherence, directory protocol, two-level directory, private/shared blocks.

1. INTRODUCTION AND MOTIVATION

Chip-multiprocessors (CMPs) have dominated a wide spectrum of the microprocessor market, ranging from embedded to supercomputers, since a few years ago. The on-chip core counts in CMPs have rapidly grown and it is expected to reach several hundreds of cores in the following years [1]. Most CMPs allow the shared memory programming model and implement a cache coherence protocol to maintain data coherence among the processor caches. The use of directory-based protocols is the commonly preferred approach, since this approach represents the most scalable alternative by keeping track of every cached memory block in the system to avoid broadcasting messages.

Traditionally, directory caches have kept track of both shared and private blocks all together. However, both types of blocks have different requirements in terms of area and latency. First, directory entries of shared blocks are accessed by several cores while entries of private blocks are accessed only in they are evicted from the processor cache and requested again. Second, to keep track of private blocks, the area-demanding sharing vector is not required. Finally, as recently stated [2], most of the accessed blocks in parallel workloads are private.

PACT'12, September 19–23, 2012, Minneapolis, Minnesota, USA.
ACM 978-1-4503-1182-3/12/09.

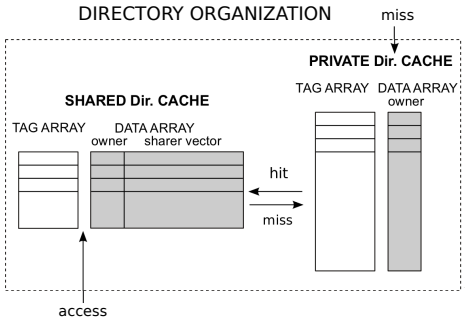

Figure 1: Private-Shared directory organization.

In this paper we propose a novel directory cache organization which discerns between private and shared blocks. PS-Dir (Private/Shared Directory) is organized in two independent caches, namely the Private cache and the Shared cache, aimed at tracking coherence of private and shared blocks, respectively. The Shared cache is designed with the same fields as a typical directory cache but with much lower number of entries, attending to the low fraction of expected shared blocks; but fast due to the high lookup rate expected. In contrast, the Private cache has a higher number of entries but it does not includes a sharing vector, thus allowing scalability. Additionally, while the Shared cache should be implemented in typical SRAM technology for speed reasons, the Private cache can use eDRAM technology [5], which although slower, it incurs less energy consumption and area than SRAM. eDRAM technology has been already used to implement large caches in some recent commercial processors like the IBM Power7 [3].

Experimental results for a 16-core CMP show that PS-Dir improves performance by 11.1% compared to a conventional directory due to the separate treatment of private and shared blocks, while reducing its area by 18.85%. In addition, when eDRAM technology is considered for the Private cache this area reduction becomes 25.39%. In terms of energy, PS-Dir allows energy savings by 20.5%.

2. THE PRIVATE/SHARED DIRECTORY

Figure 1 depicts the proposed organization consisting of the Private cache and the Shared cache, as well as the main fields required for each cache structure. As observed, both

the height (number of entries) and the width (bits per entry) of both directory caches differ. The different height is because most of the blocks are private, so the Private cache has been designed with more entries. The width differs because the sharer vector, whose size does not scale with the core count, is only implemented in the small Shared cache.

The proposal considers the Private cache as the default cache, that is, on a directory miss the block is assumed to be private and fetched into the Private cache. On subsequent accesses to a private block, the processor will find it in its L1 cache (in case of no eviction), so no additional directory access will be done. On the other hand, when a private block is removed from the processor cache, the protocol can invalidate the associated directory entry. Thus, a subsequent access to that block would result in a directory miss. This means that the Private cache access time does not affect the performance of private blocks since these blocks are provided directly to cores by the NUCA slice or main memory.

If a block in the Private cache is accessed by a core other than the owner, the block becomes shared and therefore, it is moved to the Shared cache and the sharing vector is updated accordingly. From then on and until eviction, coherence of this block is tracked in the Shared cache. Evictions from the Shared cache are removed from the directory.

The proposal reduces area by design with respect to conventional caches since entries in the Private cache are narrower. In addition power is also reduced by accessing smaller cache structures sequentially. Nevertheless, the use of two independent organizations with different design goals (speed for the Shared and capacity for the Private) suggests that the use of specific technologies could provide our proposal further energy and area savings. This work explores the use of eDRAM technology in the Private cache which leads to important area and leakage savings.

3. EXPERIMENTAL EVALUATION

The proposal has been evaluated with full-system simulation using Virtutech Simics, the GEMS toolkit [4], and a wide range of scientific applications from both the SPLASH-2 and the PARSEC benchmark suite. We simulate a 16-tile CMP with 2-cycle, 64KB, 4-way L1 instruction and data caches, and a shared (NUCA) LLC cache with 512KB and 8-way per tile. Each LLC bank is accessed in 6 cycles (2 for accessing only the tags).

Different configurations for PS-Dir have been evaluated and compared to a conventional directory (*single cache* configuration) with the same number of entries as an L1 processor cache, i.e., 1× coverage ratio. The coverage ratio indicates the number of directory entries per processor cache entry. Different coverage ratios have been evaluated ranging from 1× to 0.125×. We used the CACTI 6.5 toolto estimate access time, area requirements, and power consumption of the different cache structures for a 32nm technology node.

Performance. Every time a directory entry is evicted, invalidation messages are sent to processor caches for coherence purposes. These invalidations will cause *coverage* misses upon a subsequent memory request to those blocks, impacting on the final performance. Reductions in the number of coverage misses translate into improvements in execution time. As observed in Figure 2, a PS-Dir with a 1× coverage ratio reduces execution time on average by 11.1%. Alternatively, if reducing silicon area is a design goal, one can opt for reducing the area overhead of the directory without losing performance. The 0.125× PS-Dir achieves the same performance as a single cache.

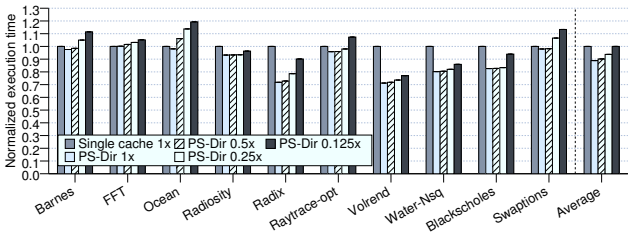

Figure 2: Normalized performance with respect to a conventional single-cache directory.

Area. PS-Dir reduces area even for the 1× configuration. This is because the Private cache does not include a sharing vector. In fact, the 1× PS-Dir with the SRAM Private cache saves 18.8% of area compared to the single cache. In addition, when eDRAM technology is considered these reductions increase up to 25.4%. Moreover, when the directory coverage ratio is reduced (0.5× and 0.25×), area savings significantly increase up to 80.2% for the 0.25× configuration, while still improving the system performance.

Energy. The PS directory also attacks by design the energy consumption, specially leakage, due to its area reduction. The 1× and 0.5× PS-Dir configurations consume more dynamic energy per access than the conventional cache, but this is highly offset by the much lower leakage consumed. Leakage is reduced from 19% for the SRAM 1× PS-Dir up to 86% for the eDRAM 0.25× PS-Dir. Regarding total directory energy, the 1× PS-Dir can save around 20.5% of the energy consumption of the single cache directory. Smaller coverage ratios lead to less energy consumed at the cost of performance degradation.

4. CONCLUSIONS

This work proposes the PS directory, which uses two different directory cache structures tailored to the behavior of the blocks they track: the Shared directory cache, aimed at keeping track of shared blocks is small and fast, and a larger Private directory cache, aimed at tracking private blocks, which does not store the sharing vector. Compared to a single directory cache with the same number of entries and considering a 16-core system, the PS directory improves performance by 11.1%, area by 25.4%, and energy by 27%.

5. ACKNOWLEDGMENTS

This work has been supported by the Spanish MICINN, as well as European Commission FEDER funds, under Grants CSD2006-00046 and TIN2009-14475-C04.

6. REFERENCES

[1] S. Bell, et al. TILE64^{TM} processor: A 64-core SoC with mesh interconnect. In *ISSCC*, pages 88–598, Jan. 2008.

[2] B. Cuesta, A. Ros, M. E. Gómez, A. Robles, and J. Duato. Increasing the effectiveness of directory caches by deactivating coherence for private memory blocks. In *38th ISCA*, pages 93–103, June 2011.

[3] R. Kalla, B. Sinharoy, W. J. Starke, and M. Floyd. Power7: IBM's Next-Generation Server Processor. *IEEE Micro*, 30:7–15, 2010.

[4] M. M. Martin, et al. Multifacet's general execution-driven multiprocessor simulator (GEMS) toolset. *Computer Architecture News*, 33(4):92–99, Sept. 2005.

[5] R. E. Matick and S. E. Schuster. Logic-based eDRAM: Origins and rationale for use. *IBM Journal of Research and Development*, 49(1):145–165, 2005.