Investigating the Effects of Wrong-Path Memory References in Shared-Memory Multiprocessor Systems

Ayse Yilmazer[1], Resit Sendag[1], Joshua J. Yi[2], and Augustus K. Uht[1]

[1] - Department of Electrical and Computer Engineering
University of Rhode Island, Kingston, RI
yilmazer,sendag,uht@ele.uri.edu

[2] - Networking and Computing Systems Group
Freescale Semiconductor, Inc., Austin, TX
joshua.yi@freescale.com

Abstract

Uniprocessor studies have shown that wrong-path memory references pollute the caches by bringing in data that are not needed for the correct execution path and by evicting useful data or instructions. Additionally, they also increase the amount of cache and memory traffic. On the positive side, however, they may have a prefetching effect for loads and instructions on the correct path. While the wrong-path effects are well studied for uniprocessors, there is no work on its effects on multiprocessor systems. In this paper, we explore the effects of wrong-path memory references on the memory system behavior of shared-memory multiprocessor (SMP) systems with broadcast (snoop-based) and directory-based cache coherence. We show that in contrast to uniprocessor systems, these wrong-path memory references can increase the amount of cache-to-cache transfers by 32%, invalidations by 8% and 20% for broadcast and directory-based SMPs, respectively, and the number of write-backs by up to 67% for both systems. In addition to the extra coherence traffic, wrong-path memory references also increases the number of cache line state transitions by 21% and 32% for broadcast and directory-based SMPs, respectively.

1 Introduction

Shared-memory multiprocessor (SMP) systems are typically built around a number of high-performance out-of-order superscalar processors, each of which employs aggressive branch prediction techniques in order to achieve high issue rate. During execution, these processors speculatively execute the instructions following the direction and target of a predicted branch instruction. If later detected incorrect, these wrong-path memory references do not change the processor's architectural state, however, they do change the data and instructions that are in the memory hierarchy, which can affect the processor's performance.

Several authors have studied the effects that speculatively executed memory references have on the performance of out-of-order superscalar processors [2, 3, 4] and have shown that wrong-path memory references may function as indirect prefetches by bringing data into the cache that are needed later by instructions on the correct execution path [1, 3, 4]. However, these wrong-path memory references also increase the amount of memory traffic and can pollute the cache with cache blocks that are not referenced by instructions on the correct path [1, 3].

In this paper, we focus on the effect that wrong-path memory references have on the memory system behavior of SMP systems, in particular, broadcast-based and directory-based SMP systems. For these systems, not only do the wrong-path memory references affect the performance of the individual processors, they also affect the performance of the entire system by increasing the number of coherence transactions, the number of cache line state transitions, the number of write-backs and invalidations due to wrong-path coherence transactions, and the amount of resource contention (buffer usage, bandwidth, *etc.*).

2 Evaluating Wrong-Path Effects

In this section, we discuss the potential effects that wrong-path memory references can have on the memory behavior of SMP systems. To measure the various wrong-path effects, we track the speculatively generated memory references, and mark them as being on the wrong-path when the branch misprediction is known. Due to the space limitation, we only give a subset of the effects. For a complete version refer to [5].

2.1 L1 and L2, and Coherence Traffic

We observe that wrong-path loads increase the total number of memory references issued to the memory system on average by 17% and 14%, respectively, for broadcast and directory-based SMPs. Additionally, these loads increase the percentage of L2 cache accesses by 23% and 21% for broadcast and directory-based SMP systems, respectively. Our results also show that wrong-path loads increase the number of coherence transactions by an average of 32%.

2.2 Replacements and Write-backs

A speculatively-executed (*i.e.*, later detected as wrong-path) load instruction may bring a cache block into processor's data cache that replace another block that may be needed later by a correct-path load. Due to this replacement, these wrong-path load can cause extra cache misses *i.e.* pollution [3]. Wrong-path replacements may also cause extra write-backs that would not occur otherwise. For example, if the requested wrong-path block has been modified by another processor, *i.e.*, its cache coherence state is M, a shared copy of that block is sent to the requesting processor's cache, which subsequently may cause a replacement. When the evicted block has a cache coherence state of M (exclusive, dirty) or O (shared, dirty) state, this also causes a write-back.

Figure 1 shows the percentage increase in the number of *E* (for directory MOESI) and *S* line replacements. *E*→*I* transitions – which increased by 2% to 63% – are particularly important since the processor loses the ownership of a block and the ability to silently upgrade its value, potentially significantly increasing the number of invalidations for write upgrades. The number of *S* line replacements account for a significant fraction of the total number of the replacements due to wrong-path load in broadcast SMPs; in directory-based SMPs, they are relatively insignificant.

In Figure 2, we observe that wrong-path reads increase the number of write-backs from 4% to 67%.

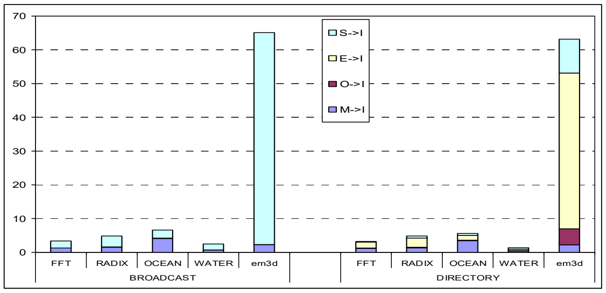

Figure 1 Percentage increase in the number of *replacements* due to wrong-path references in broadcast and directory-based SMPs.

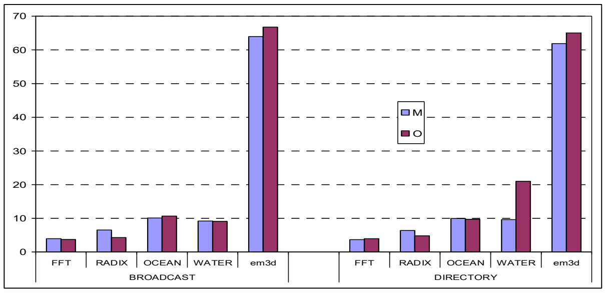

Figure 2 Percentage increase in the number of *write-backs* due to wrong-path references in broadcast and directory-based SMPs.

2.3 Cache Line State Transitions

Finally, Figure 3 shows the impact of wrong-path memory references on the number of cache line state transitions. The results show that the number of cache line state transisions increases by 20% to 24% for broadcast SMPs and by 27% to 44% for directory-based SMPs.

An exclusive cache block (modified or clean) loses its ownership when another processor attempts to load that cache block. In order to gain ownership again, that processor has to first invalidate all other copies of that cache block, *i.e.,* Shared→Invalidate for all other processors. In Figure 4, we can see that for broadcast SMPs, there is 8% to 11% increase in the number of write misses, each of which subsequently causes an invalidation. This percentage is higher, 15% to 26%, for the directory SMPs.

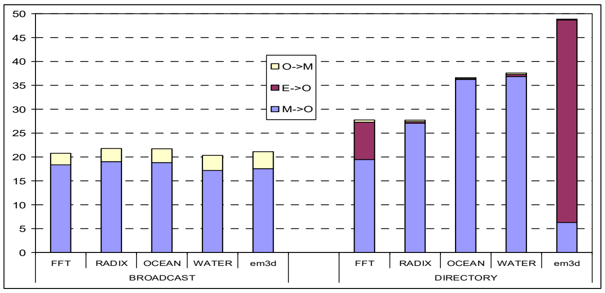

Figure 3 Percentage increase in the number cache line *transitions* for MOSI broadcast and MOESI directory SMPs

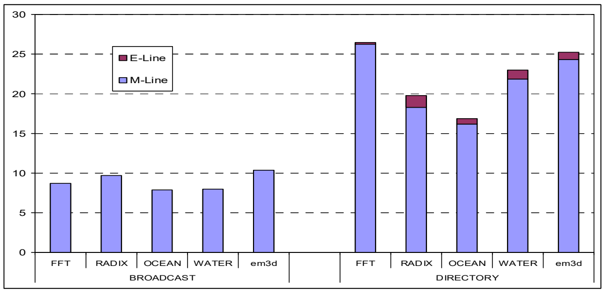

Figure 4 Increase in the *write misses* and extra *invalidations* due to wrong-path references.

3 Conclusion

In this paper, we evaluate the effects of executing wrong-path memory references on the memory behavior of cache coherent multiprocessor systems. Our evaluation reveals the following conclusions: (1) Modeling wrong-path memory references in a cache coherent shared memory multiprocessor is important and not modeling them may result in wrong design decisions, especially for future systems with longer memory interconnect latencies and processors with larger instruction windows. (2) In general, wrong-path memory references are beneficial because they prefetch data into caches. However, there can be significant amount of pollution caused by these references. (3) For a workload with many cache-to-cache transfers, the coherence actions can be significantly affected by wrong-path memory references.

References

[1] O. Mutlu *et al.* Understanding the effects of wrong-path memory references on processor performance. WMPI 2004.

[2] J. Pierce and T. Mudge. Wrong-path instruction prefetching. In MICRO, pages 165–175, 1996.

[3] R. Sendag *et al.* Exploiting the prefetching effect provided by executing mispredicted load instructions. Euro-Par 2002.

[4] R. Sendag *et al.,* The Impact of Incorrectly Speculated Memory Operations in a Multithreaded Architecture, IEEE TPDS, 2005.

[5] R. Sendag *et al.*, Quantifying and Reducing the Effects of Wrong-Path Memory References in Cache-Coherent Multiprocessor Systems, IPDPS 2006.