



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

Ayse Yilmazer(1), Resit Sendag(1), Joshua Yi(2), and August Uht(1) (1) Microarchitecture Research Institute, University of Rhode Island (2) Freescale Corporation

# Outlines

- Wrong Path Effects on Shared-Memory Multiprocessor Systems (SMPs)
  - Broadcast (snoop-based) and directory-based SMPs
- Simulation Methodology
- Evaluation Results
- Summary

# Motivation

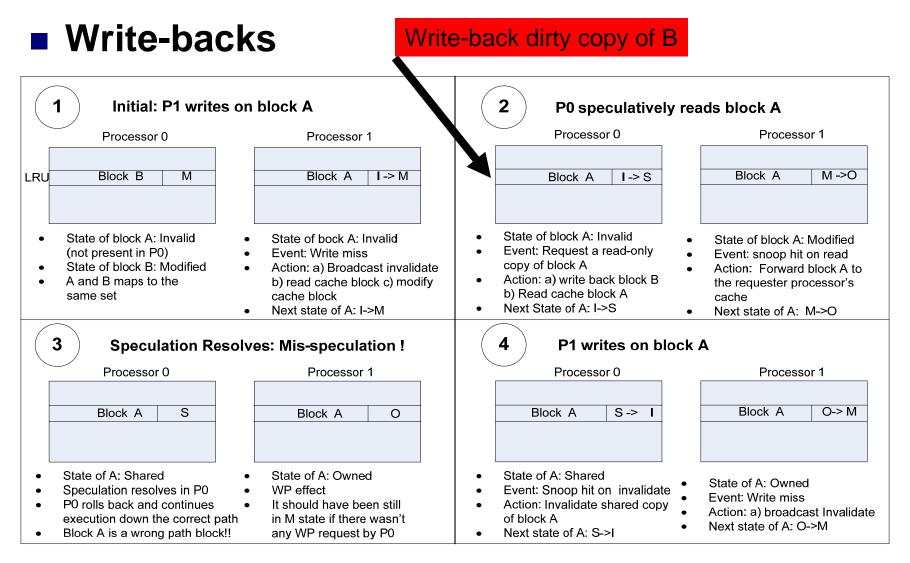
### Wrong-path (WP) effects on Uniprocessors

- Negative Effects: Pollution
  - L1 and L2 cache pollution
- Positive Effects: Prefetching
  - Up to 20% better performance for mcf
- Important to simulate WP for some applications

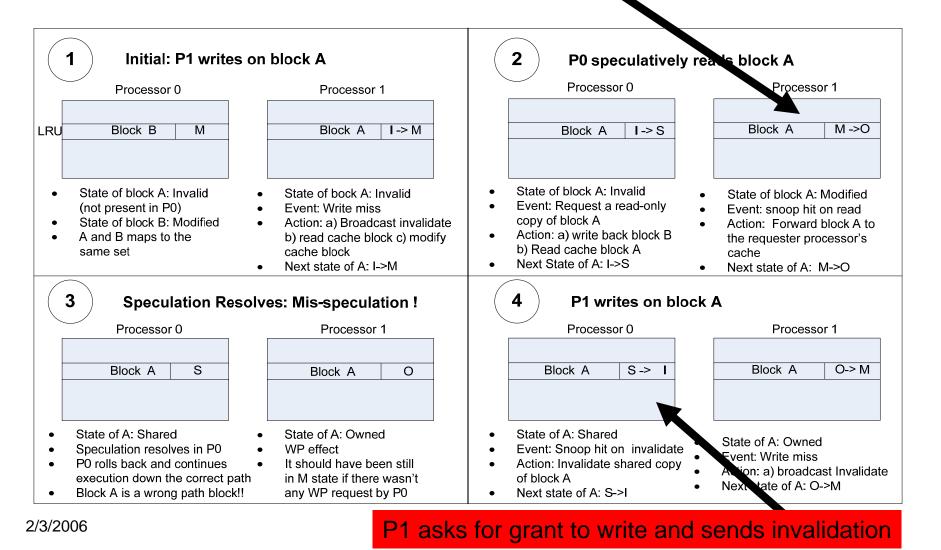
### No work on WP effects on Multiprocessors

- □ In contrast to uniprocessor effects, WP cause:
  - Extra coherence traffic:
    - Data, invalidations, write-backs, acknowledgements
  - Additional cache state transitions

#### **Initial States** A speculatively replaces B Replacements Initial: P1 writes on block A 1 P0 speculatively reads block A Processor 0 Processor 1 Processor 0 Processor 1 Block B Μ Block A I -> M Block A M ->O LRU Block A I-> S State of block A: Invalid State of block A: Invalid State of bock A: Invalid ٠ State of block A: Modified Event: Request a read-only (not present in P0) Event: Write miss . Event: snoop hit on read copy of block A State of block B: Modified Action: a) Broadcast invalidate Action: Forward block A to Action: a) write back block B A and B maps to the b) read cache block c) modify ٠ • the requester processor's b) Read cache block A same set cache block cache Next State of A: I->S Next state of A: I->M ٠ Next state of A: M->O 3 **Speculation Resolves: Mis-speculation !** 4 P1 writes on block A Processor 0 Processor 0 Processor 1 Processor 1 O-> M Block A S Block A S-> I Block A Block A Ο tate of A: Shared State of A: Shared State of A: Owned ٠ State of A: Owned Speculation resolves in P0 WP effect Event: Snoop hit on invalidate Event: Write miss P0 rolls back and continues It should have been still Action: Invalidate shared copy Action: a) broadcast Invalidate oution down the correct nath in M state if there wasn't of block A Next state of A: O->M equest by P0 Next state of A: S->I . A is a Wrong-path Block !

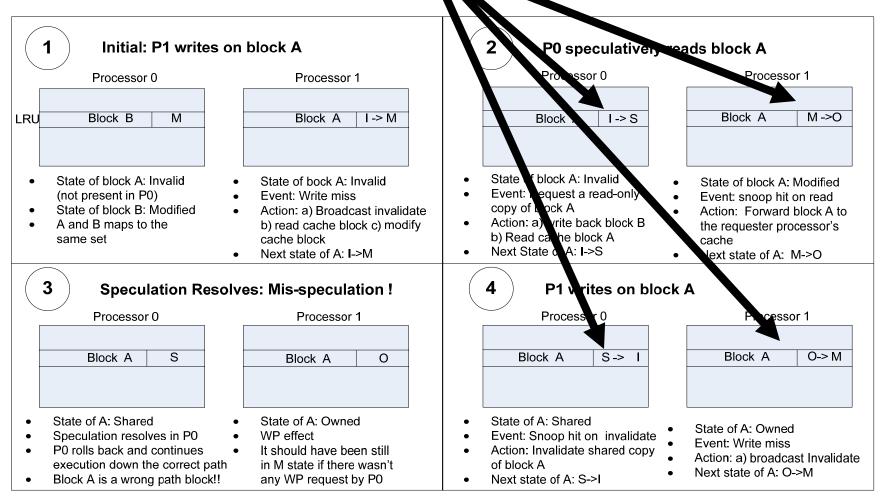


Write misses and invalid P1 loses its write privileges for block A



States Change

### Cache block state transitions



#### Data/Bus and Coherence Traffic Increases

- L1 references
- L2 references
- □ Coherence traffic
  - Snoop, directory requests for data and invalidations

#### Power Consumption Increases

Due to extra cache accesses, coherence traffic and cache line state transitions

#### Resource Contention

- Competing with Correct-path resources
  - Full service buffers, critical when many cache-to-cache transfers

# Simulation Methodology

- GEMS simulator Wisconsin Multifacet Group
  - Based on Virtutech SIMICS
  - □ Aggressive out-of-order superscalar processor
  - Detailed Shared-Memory Model
- We evaluate 16-processor SPARC V9 system running unmodified Solaris 9
- Evaluated both Snoop-based MOSI and Directory-based MOESI coherence
  - □ MOSI: Modified, Owned, Shared, Invalid
  - □ MOESI: Modified, Owned, Exclusive , Shared, Invalid
- We track the speculatively generated memory references
  - And mark them as being on the wrong-path when the branch misprediction is known

### Evaluation Results (Cont'd) -- L1 and L2, and Coherence Traffic



**Broadcast and Directory – Coherence Traffic** 

### Evaluation Results (Cont'd) -- L1 and L2 Cache WP Replacements

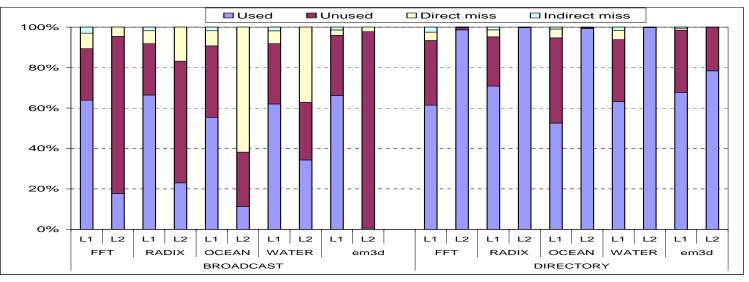
4 Categories:

1. Unused : evicted before being used or never used by a correct-path

2. used: used by a correct-path reference

3. *direct-miss*: Replaces a cache block that is needed by a later correct-path load, but are evicted before being used.

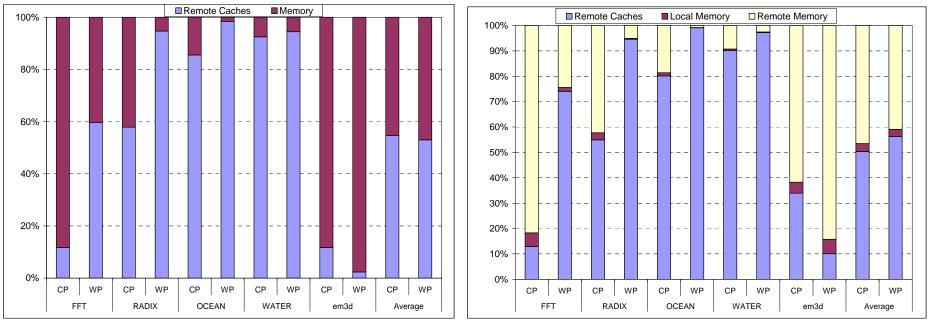
*4. indirect-miss*. LRU changes in a set may eventually cause correct-path misses.



55-67% L1 and 12-35% L2 replacements are used in broadcast SMPs.

## Evaluation Results (Cont'd)

#### -- Servicing Coherence Transactions

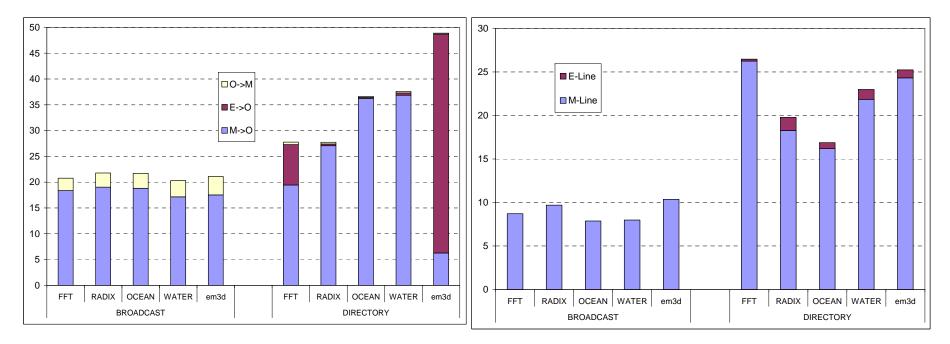


Broadcast

Directory

### Evaluation Results (Cont'd)

#### -- Cache Line State Transitions



State Transitions

#### Write misses and invalidations

# Summary of Effects

- Uniprocessor effects (i.e., pollution & refetching) apply. Moreover,
- Increase in Coherence Traffic
  - □ Cache-to-cache transfers by 32%
  - Invalidations by 8% and 20% for broadcast and directory-based SMPs, respectively
  - □ Write-backs by up to 67% for both systems
- Extra Cache Line State Transitions
  21% and 32% for broadcast and directory-based SMPs, respectively