

Motivation

- Chip Multiprocessors (CMPs) are ubiquitous
- Sharing of data among cores is a critical bottleneck
- Q1: Role of interference
 - OS memory references
 - Prefetch requests and page table walks caused by TLB miss handling
- Q2: TLB cooperation and management
 - Sharing
 - Prefetching

Methodology

Real-System Infrastructure: Intel Nehalem (Core i7)

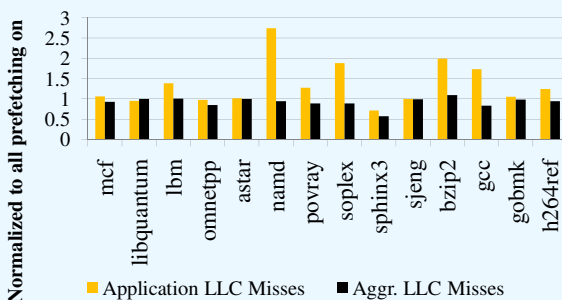
- 128-entry L1 ITLB, 64-entry (small page) +32-entry (large page) L1 DTLB, 512-entry L2 DTLB
- 32KB L1 I/D cache, 256KB L2 cache, 8MB L3 cache

Full-System Simulation: Virtutech Simics with Multifacet GEMS (UltraSPARC III Cu, Sun Solaris 10)

- Parallel applications: PARSEC benchmark suite
- Sequential applications: SPEC CPU2006

Prefetch Requests Cause Cache Interference

Intra-Application LLC Interference Caused by MLC Prefetchers



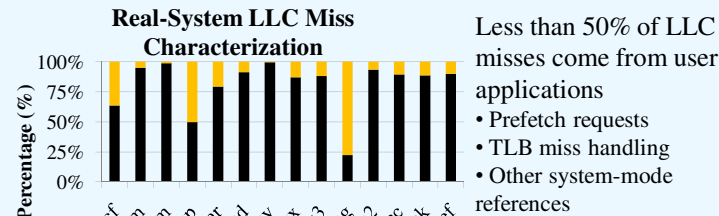
Acknowledgements

The authors acknowledge the support of the Gigascale Systems Research Center, one of six research centers funded under the Focus Center Research Program, a Semiconductor Research Corporation program. In addition, this work was supported in part by National Science Foundation grant CCF-0916971.

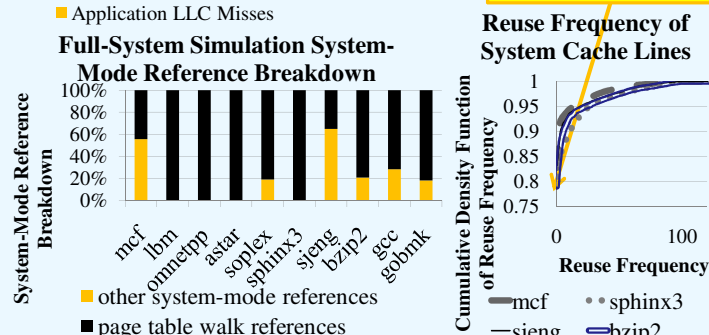
Intra-Application Last-Level Cache Interference

Objective:

- Characterize real-system intra-application LLC interference
- Eliminate intra-application interference via dynamic cache management

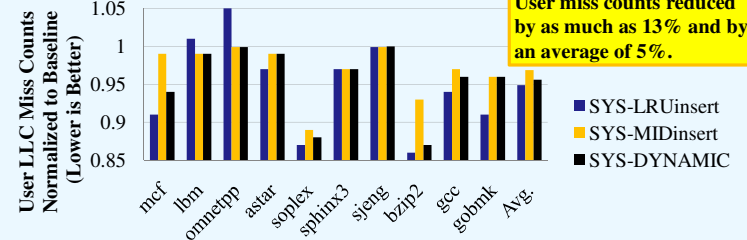


More than 75% of system cache lines are never reused before eviction

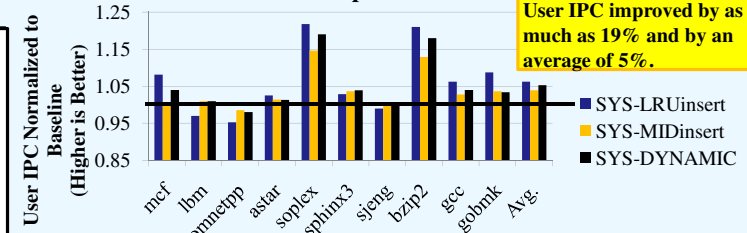


Eliminating Interference via Dynamic Cache Management

User LLC Miss Count Reduction



User IPC Improvement



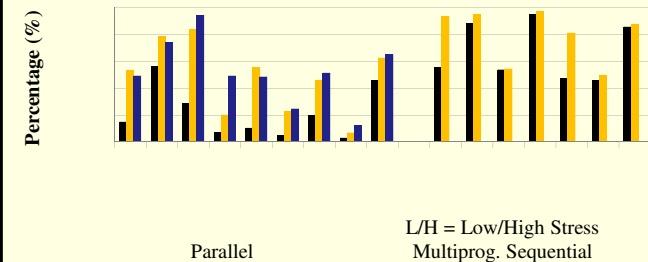
Shared Last Level TLBs

Objective: Improve performance by replacing standard private L2 TLBs with Shared Last-Level (SLL) TLBs

- Eliminate unnecessary redundancy in multiple cores
- More flexibility in number of entries allocated per core

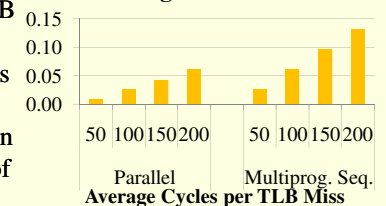
Results

- Workloads of parallel (from PARSEC) and multiprogrammed sequential applications (from SPEC CPU06) show dramatic improvements in hit rate



- Hit rate improvement is as good as or better than previously proposed ICC prefetchers alone
- Simple stride prefetching further improves TLB hit rates
- Performance models show consistent cycles-per-instruction savings, regardless of TLB miss penalty

Average CPI Saved



Conclusions

- Real-system characterization for LLC misses
 - Degree of intra-application interference on Nehalem.
 - The number of app. LLC misses is reduced by as much as 13% (an avg. of 5%) via dynamic cache management.
- Shared Last Level TLB increase hit rates by an average of 21% for sequential and 27% for parallel workloads over conventional private L2 TLBs per core