CSE 231 Paper Discussion
Vijay S. Pai and Sarita Adve,
Code Transformations to Improve Memory Parallelism
Proceedings of the 32nd Annual IEEE/ACM International Symposium on Microarchitecture, pages 147-155. November 1999.

Motivation

- Mismatch between processor speed and the speed of interconnects and memory.
- Processors can execute instructions out of order and speculatively, but can only retire them in order.
- Fixed, relatively small (32 to 80 instruction) window of instructions that have been issued but not retired.
- A cache miss can take 200 to 300 processor cycles, several times the time to execute the next 80 instructions if they are not cache misses. Processor idles for remaining time.

Solution Overview

- Cluster cache read misses, so that multiple misses are in the same window, while preserving locality.
- Use unroll-and-jam to make each inner loop iteration do its work for several outer loop iterations.

Base Code   Interchange   Unroll-and-jam
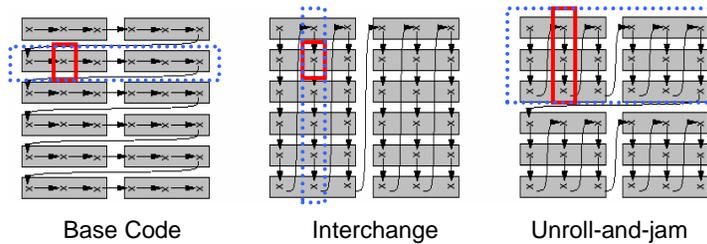
Solution details

- Apply to reads, assuming writes can be buffered.
- Evaluate whether to cluster.
  - Calculate number of outer iterations to unroll
- Use existing algorithms to check legality and do the transformation.
- Reorder, scalar replacement, within enlarged inner loop iteration.

Results

Hand implemented optimization, Latbench and several scientific applications

- Simulated uniprocessor

  Latbench 5.34x speedup, Applications 11-48% reduction, average 30%

- Simulated SMP

  Applications 9-39% reduction, average 23%

- Convex Exemplar

  Latbench 5.77x speedup, Applications 9-38% reduction, except for 3% degradation in Ocean.

  (Exemplar application data from extended report)

Issues and Questions

- Increased bus and memory contention.
- Interaction between miss clustering and prefetch.
  - See "Comparing and Combining Read Miss Clustering and Software Prefetching" by same authors.
- Arrays with outer dimension stride not a multiple of cache line size?

  "double precision a(1001,1000),b(1000),x(1000)"
  (http://www.netlib.org/benchmark/1000d)

- Multiple arrays with cache misses in different outer loop iterations?
- Number of outstanding misses may be processor model dependent. What will the effect be on processors with fewer?