ILP vs. Consistency

An Evaluation of Memory Consistency Models for Shared-Memory Systems with ILP Processors

Vijay S. Pai, Parthasarathy Ranganathan, Sarita V. Adve, and Tracy Barton

Department of Electrical and Computer Engineering
Rice University
Houston, Texas 77251-1892
{vijaypai|parthas|sarita|harton}@rice.edu

What’s the big question?

Architectural Parameters

- SC and RC
- Write-through vs write-back caches
- Load prefetching
- Write prefetching
- Speculative loads
HW-generated Prefetching

• How does this hide consistency-related latency?
  • For Loads?
  • For Stores?
  • Better with WT or WB cache?
  • What about pointer chasing applications?

Speculative Loads

• How does this hide consistency-related latency?
  • For Loads?
  • For Stores?
  • Better with WT or WB cache?
  • What about pointer chasing applications?

RC with Prefetching?

RC with Speculative Loads?
More aggressive protocol

- Originally – cannot overlap ownership request with shared read request.
- Optimization?
- Why Hard?
- Why helpful?

Still a gap between RC and SC

- Why?

Fuzzy Acquires

Code for Processor Pt

```
for(i < N/1); W = WaitFlagFlag[i];
ACquireMemory[Flag][i];
DoWork[Flag][i];
ReleaseMemory[Flag];
SetFlag[Flag][i] == 0;
```

Figure 4: Application of fuzzy acquires.
Selective Acquires

Is SC + ILP = RC?

Chris Gniady, Babak Falsafi, and T. N. Vijaykumar
School of Electrical & Computer Engineering
Purdue University
1285 EE Building
West Lafayette, IN 47907
impetus@ecn.purdue.edu, http://www.ece.purdue.edu/~impetus

Big Idea?

Speculative Execution

FIGURE 1: Speculative execution in current microprocessors.
How do you

• Make speculative loads work in a SC system?

• Do write buffering in a SC system?

In this paper

• What do you need to make SC match RC?

Reordering stores

• What do you need to make this work in SC?
FIGURE 3: Comparison of SC, RC, and SC++.

FIGURE 4: Impact of network latency.

FIGURE 5: Impact of reorder buffer size.

FIGURE 6: Impact of speculative stores.

FIGURE 7: Impact of the L2 cache size.