Multithreaded Value Prediction

N. Tuck and D.M. Tullesn

HPCA-11 2005

CMPE 382/510 Review Presentation
Peter Giese
30 November 2005
Outline

• Motivation
• Multithreaded & Value Prediction Architectures
• Single Fetch Simulation and Experiments
• Reported Performance
• Conclusions & Review Comments
Motivation – What is the problem?

• **Overcoming long memory latencies**
  – Relative performance gap between memory access time and microprocessor cycle time increasing
  – Approaching 1000 cycle memory access latency

• **Single threaded (traditional) approaches are ineffective**
  – Out-of-order execution not effective for latencies of this size
  – Single-threaded value predication schemes quickly fill up instruction windows
  – Cannot make more than a single prediction for any single dynamic instruction
Proposed Solution

• Combine value predication (VP) and multithreading on SMT processor to improve performance
• Overcome long memory latencies with load-based VP
• Approach:
  – Use single and multi-value predictions for load instructions
  – Use value predication to decouple speculative stream from non-speculative stream
  – Multithreading applied to a single program
Multithreading Discussion

• Instruction window size (large) is a key to achieving high performance
  – Long speculative execution paths

• Reviewed the ability of multi-threading architectural frameworks for:
  – Execution of multiple independent paths
  – Decoupled execution occurring in arbitrarily separated windows
  – Allowing speculative execution

• Checkpoint architecture share similar goals …
Checkpointing (Review)

- Checkpoint created at certain instructions
- Allow instructions to commit in order they finish, but don’t update memory
- Update memory only when checkpoints commit and always commit checkpoints in order
- Rollback to previous checkpoint to handle exceptions and branch misprediction

Source: Out-of-order Commit Processor (HPCA –10)
Proposed Frameworks

**Simple Threaded Value Prediction (STVP)**

- Load VP
- Dependent instructions proceed in parallel within the instruction window of a single thread
- Commitment only after predicated load is validated

**Multithreaded Value Prediction (MTVP)**

- Load VP spawns a speculative thread
- Separate non-speculative and speculative thread context
- Speculation distance limited by size of store buffer (memory writes)
- Mechanisms:
  - Flash copy register map to replace register state (STM Arch)
  - Separate context maintained for a tree of speculative threads
  - Separate store buffer per thread context (PolyPath Arch)
Baseline Simulation

• “Specialized” version of MTVP used in experiments
  – Single Fetch Path: Sacrifices speculative benefits but simplifies implementation

• Simplified implementation:
  – Simple thread tracking, linear rather than tree
  – No fetch interruption, only thread context change
  – Single store buffer with thread-based tags
Value Prediction

• **Context-based data value prediction**
  – Learns the values that follow a particular sequence of data values (context) and predicates a value

• **Predicts the actual data to be brought in from memory**
  – Value is inserted into the instruction’s allocated register
  – Miss-speculation check preformed when real data becomes available
Context-based Data Prediction

Context Methods

Data Sequence: 111231123111

0\(^{\text{th}}\) Order

\[\begin{array}{c|c|c|c}
1 & 2 & 3 \\
\hline
1 & 6 & 2 & 0 \\
2 & 0 & 0 & 2 \\
3 & 2 & 0 & 0 \\
\end{array}\]

1\(^{\text{th}}\) Order

Wang-Franklin

Based on Two-level Value Prediction

• Different types of value prediction referenced:
  – Finite Context Method (FCM): uses past data values
  – Differentiated FCM (DFCM): uses difference between past values
  – Wang-Franklin VP: uses pattern of previous value predictions to make next prediction
Deciding when to make a prediction

Tested several approaches:

1) Based on expected cache (miss) behavior
   - Results not shown due to “inferior” performance

2) ILP-pred approach
   - Tracks forward progress between VP start & confirmation
   - Counts issued instructions and “learns” from previous VP

3) Second ILP-pred approach
   - Gauges forward progress on committed rather than issued instructions
   - “Comparable” to IPL-pred so no result reported
Simulation

• Modified version of SMTSIM multithreading simulator

• Modeling assumptions:
  – Immediate stall detection
  – Infinite store buffer
  – Variable number of threads (2, 4, 8)

• Use the SPEC CPU2000 Benchmarks
  – Both INT and FP benchmarks
Perfect Prediction

Experimental Setup

- STVP: 1 thread
- MTVP: Single Fetch Path
  - 2, 4 or 8 thread

- Normalized to baseline without value prediction
- IPC performance (geometric mean):
  - MTVP speedup of 40% (50% FP)
  - STVP speedup of 24% (5% FP)
Sensitivity: Perfect Prediction

Spawning Penalties of 1, 8, 16 cycles

1) Thread spawning sensitivity
   - No significant impact <= 8 cycles
   - FP 16 cycles results are not as sensitive as INT results
   - Assume 8 cycle spawn latency

2) Store buffer size sensitivity
   - No results presented
   - Set store buffer to 128 entries
**Realistic Value Predictor**

**Hybrid Wang-Franklin Value Predictor**

- Includes stride value in VHT
- Five learn values
- Large PHT & VHT tables (160/244 KB)
- Confidence threshold of 12 out of 32
Realistic Value Predictor

- **Greater variation in results**
  - Average speed up 40% INT (40% FP)
  - Significant improvement in the INT VPR/MCF benchmarks

- **Experiments also conducted with DFCM value predictor**
  - Results (not reported) not as good as the hybrid Wang-Franklin results
Fetch Policies

• Single versus multiple fetch path experiment
• Multiple Fetch Path MTVP
  – Parent thread continues to fetch and execute instructions
• Allowing fetch to continue in parent after value predication is counterproductive
  – Limits forward speculation
  – Competition during incorrect predictions
Multiple Value Predications

- Ability to follow multiple predictions for a single instruction
- Percentage fraction: Predication was incorrect but the correct value was present and over threshold

- Evidence indicates multi-value predication may be profitable
- Multi-value MTVP experimental results not presented
  – Limited by their implementation of Wang-Franklin VP algorithm
Checkpoint Comparison

**STVP-base Architecture**

- **Idealized Checkpoint**
  - 8192 entry ROB
  - Unlimited registers

**MTVP-base Architecture**

- **Spawn Only**
  - Spawns without VP
  - Split-Window Commit

- **Best MTVP**
  - Single fetch path
Checkpoint Results

- When abundant parallelism exists, wide-window architectures are best at exploiting it especially in FP benchmarks
- When parallelism is hard to find MTVP is effective
Stated Conclusions

• MTVP overcomes the barriers of traditional VP
  – Speeds of over 100% on a few benchmarks
  – Improvement of IPC over 40% compared to no VP

• Unlike traditional VP, MTVP is effective for both integer and floating point benchmarks

• Single fetch path MTPV out performs general multithreaded architecture
Review Comments

• Thrust of paper is VP in a generalized MTVP architecture, but results mainly report STVP
• Model large memory latencies
• No VP performance data reported
• Window ROB versus memory commit buffer sensitivity not covered – the main theme of the paper
• Anomalies in a few benchmarks produce good averages
• Most effective to go with architectures (Window/Threaded) that exploit highest level of ILP

• For a few benchmarks, STVP overcomes ILP limitations under large load-latency conditions

• Use VP is special cases – when ILP is hard to find