Rollback-Free Value Prediction with Approximate Loads
Bradley Thwaites, Gennady Pekhimenko, Amir Yazdanbakhsh, Jongse Park, Girish Mururu, Hadi Esmaeilzadeh, Onur Mutlu, Todd C. Mowry

Motivation
Perfect Prediction
- Mitigate long latency memory accesses

Microarchitecturally-triggered approximation
- Predict the value of an approximate load when it misses in the cache
- Do not check for mispredictions
- Do not rollback from mispredictions

RFVP Overview
Quickly Predict Values for Approximate Load Misses

Design Principles
- Maximize opportunities for performance and energy benefits
- Minimize the adverse effects of approximation on quality degradation

Target Performance-Critical Safe Loads
- Profile-directed compilation
- Usually, < 32 loads cause 80% of cache misses

Design Challenges
- Utilize Fast-Learning Predictors
- Two-delta stride predictor
- Prediction: table lookup plus an addition

Key Experimental Results
2MB LLC, 4-Wide, Performance Results

Value Prediction - Quality Loss
- Stride
- TwoDelta

Value & Predictor

Ongoing Work
- Extend rollback-free value prediction to GPUs
- Drop a fraction of the missed requests
- Preliminary results: Up to 2x improvement in energy and performance with only 10% quality degradation

(Performance Gain, Energy Gain, Quality Loss)
(0%, 0%, 0%) (1.06x, 1.03x, 2%)
(1.13x, 1.06x, 6%) (1.27x, 1.10x, 11%)

MoMvaMon
Core
I$ D$ LLC
Avoid Long-Latency Memory Accesses

Extend rollback-free value prediction to GPUs
Drop a fraction of the missed requests
Preliminary results: Up to 2x improvement in energy and performance with only 10% quality degradation

(Performance Gain, Energy Gain, Quality Loss)
(0%, 0%, 0%) (1.06x, 1.03x, 2%)
(1.13x, 1.06x, 6%) (1.27x, 1.10x, 11%)

Integration RFVP with existing architecture

Last Value Stride 1 Stride 2
Hash(PC)

Predicted Value

Value Prediction - Quality Loss
- Stride
- TwoDelta

Value & Predictor