Efficient GPU Hardware Transactional Memory through Early Conflict Resolution

Sui Chen and Lu Peng
Division of Electrical and Computer Engineering, Louisiana State University

Accepted & presented at the 22th IEEE Symposium on High Performance Computer Architecture (HPCA), Barcelona, Spain

Motivation and Goal

Goal: Conflict \(\Rightarrow\) Performance

Parallel programs with critical sections

**Coarse-grained locks**
- easy to use
- slow (serializes everything)

**Fine-grained locks**
- fast
- hard to program (example: RBTree)
- error-prone

GPU: massively parallel

**Transactional Memory**
- easy to use
- good performance

TM: Get the best of both coarse- and fine-locks

GPU-Specific Architecture & Conflict Types

SIMT execution gives 3 spatial types of conflicts

**Type 1**: Intra-warp (ex. A and C)
**Type 2**: Inter-warp (ex. A and B)
**Type 3**: Inter-block (ex. A and B)

Interaction with Hardware

Transaction Execution

**Value-Based Validation**

- One set of hardware change facilitates both approaches
- Modified transaction execution flow

Opportunity for Speedup

**Early abort**

- SIMT Cores *cannot see each other* directly

**Pause-and-Go**

- Aborting and retrying may be expensive
- One small conflict wastes all transactional work

Results

- 1.41x speedup
- 0.8x energy consumption
- Table size chosen using sensitivity study
- 5 workloads are CU-heavy; the rest are SIMT Core-heavy