ASeD: Availability, Security, and Debugging Support using Transactional Memory
JaeWoong Chung, Woongki Baek, Nathan G. Bronson, Jiwon Seo, Christos Kozyrakis, Kunle Olukotun
Stanford University

Motivation
• Transactional Memory (TM)
  • Simplifies parallel programming using atomic blocks
  • Easy to use & high performance

• TM systems should provide ACI features
  • Asymmetry: rollback to a safe system state
  • Consistency: guarantee system-level invariants
  • Isolation: limit the propagation of side effects

• Key insights: ACI of TM can also be used for ASeD
  • Availability
  • Security
  • Debugging

Availability Features
• ASeD addresses both permanent & transient faults
  • Permanent: loss in cores or caches, etc.
  • Transient: packet loss, logic errors, etc.

• Availability primitives
  • Global checkpoint: system-wide state
  • AI (atomic/isolated) regions: thread-specific state

• Use case
  • Permanent faults
  • Global checkpoints are periodically taken
  • Upon faults, roll-back to the latest global checkpoint
  • Transient faults
  • Code fragments are enclosed by AI regions
  • Upon faults, just roll-back the faulty thread
  • Significantly reduced MTTR

Security Features
• Security primitives
  • Fine grained read/write barriers
  • Accessing a specific address is detected & notified
  • Isolated execution
  • Similar to the AI regions

• Use case
  • Buffer overflow detection
  • Mark the address of canaries with write barriers
  • Overwrite to a canary is detected by the ASeD
  • Significantly lower overhead than SW canaries

Debugging Features
• Debugging primitives
  • Global checkpoint
  • Fine grained read/write barriers

• Use case
  • Scalable watchpoints
  • Provides arbitrary # of watchpoints using RW barriers
  • Negligible performance impact
  • Supports coarse-grain watchpoints with the runtime
  • Parallel bookmark & step-back are also supported

Feature Decomposition

ASeD on Transactional Memory
• Implementing the ASeD on HTM using similarity of ACI
  • Reduces HW costs of ASeD support
  • An integrated system of TM with the advantages of ASeD

ASeD vs HTM

Availability Experiments

Security Experiments

Debugging Experiments

Conclusions
• We proposed & evaluated the ASeD on top of HTM
  • Availability using global & local checkpoints
  • Security using RW barriers & isolated executions
  • Debugging: using global checkpoints
  • Overall, enhanced HTM with ASeD with minimal HW costs