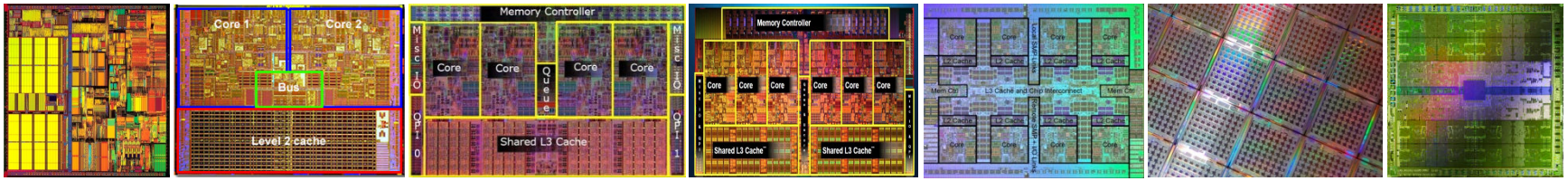
Demand-Driven Software Race Detection using Hardware Performance Counters

Joseph L. Greathouse†, Zhiqiang Ma‡, Matthew I. Frank‡
Ramesh Peri‡, Todd Austin†

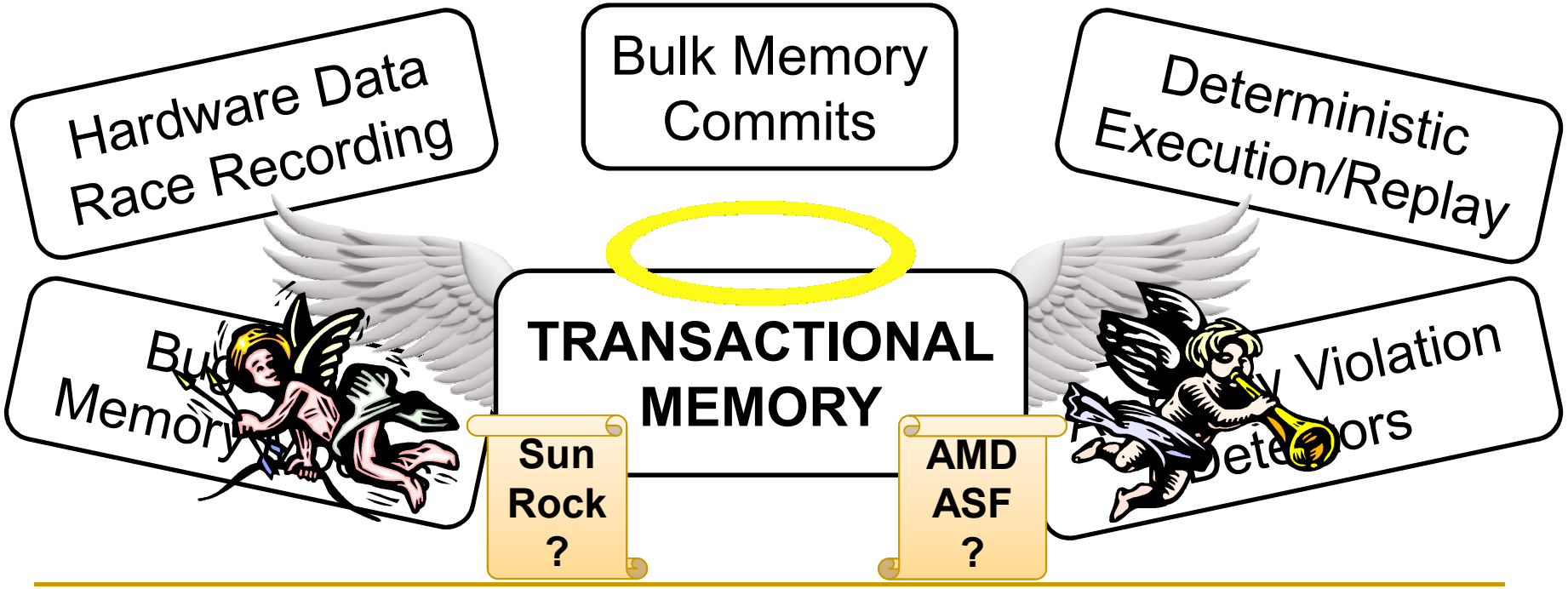
†University of Michigan ‡Intel Corporation

CSCADS
Aug 2, 2011

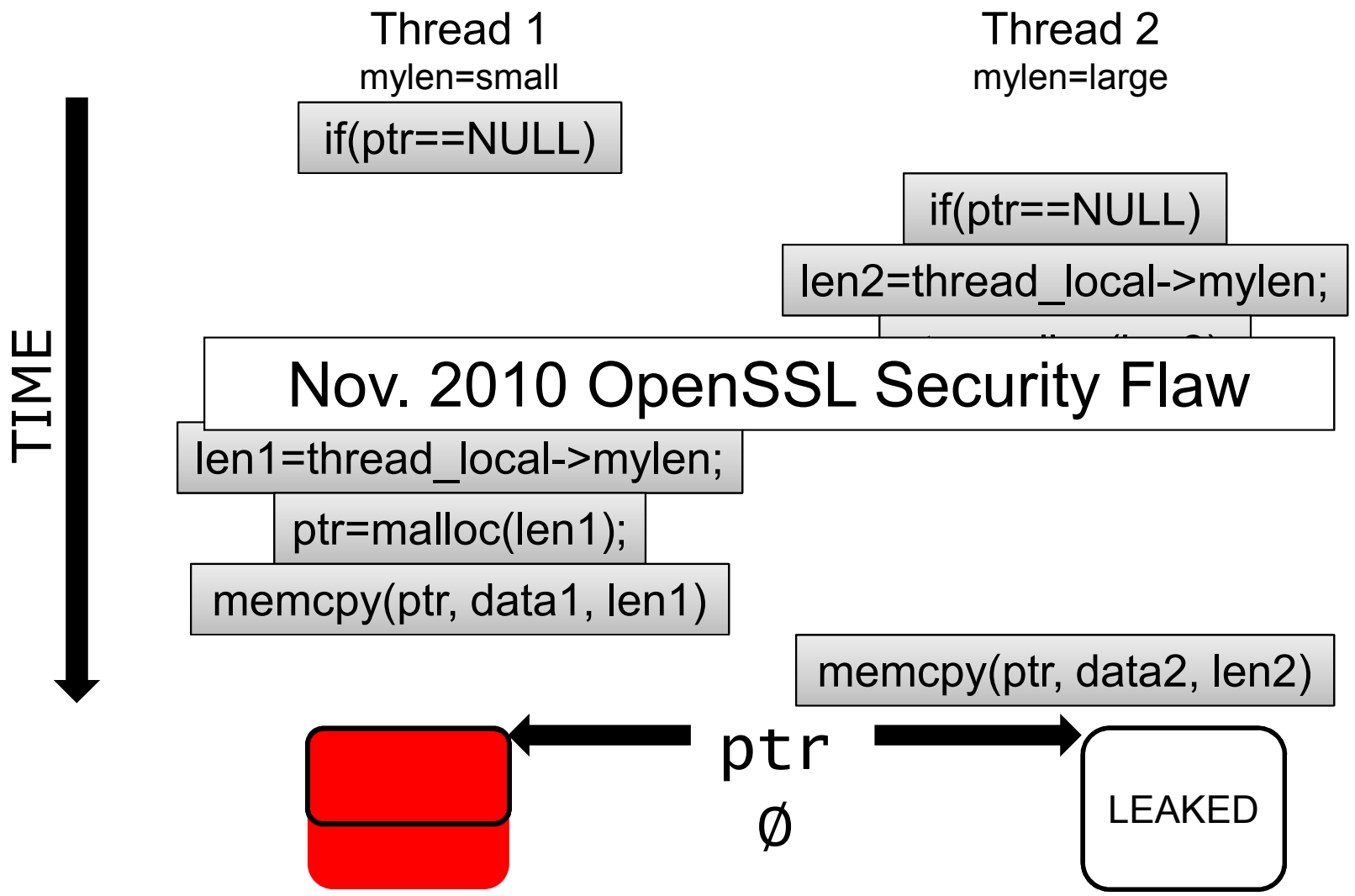
Concurrency Bugs Still Matter



In spite of proposed hardware solutions



Concurrency Bugs Matter NOW



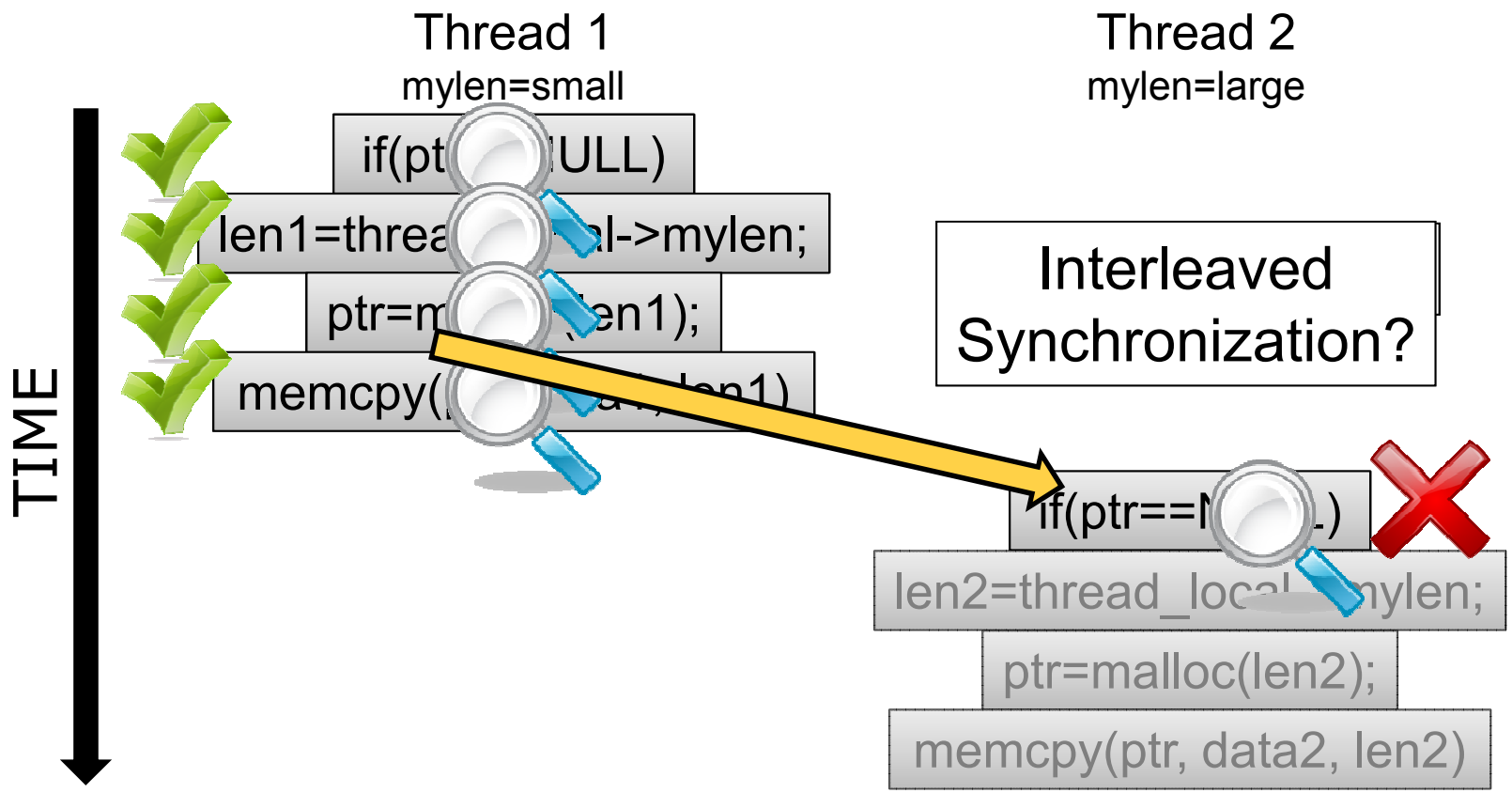
This Talk in One Sentence

Speed up software race detection with existing hardware support.

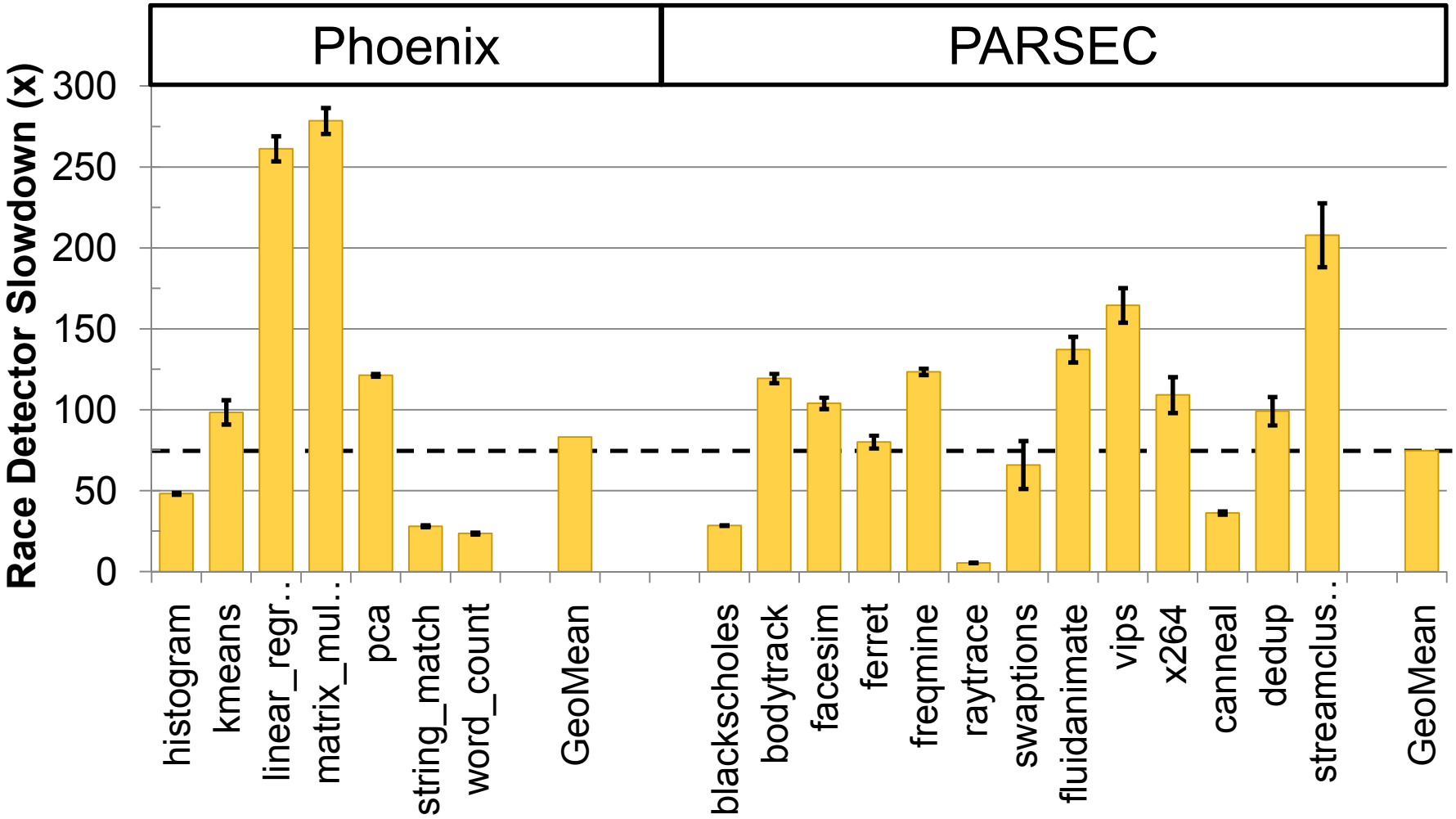
Software Data Race Detection

- Add checks around every memory access
- Find inter-thread sharing events
- Synchronization between write-shared accesses?
  - No? Data race.

Example of Data Race Detection



SW Race Detection is Slow



Goal of this Work

Accelerate Software Data Race Detection

Technique #1: *Making it Fast*

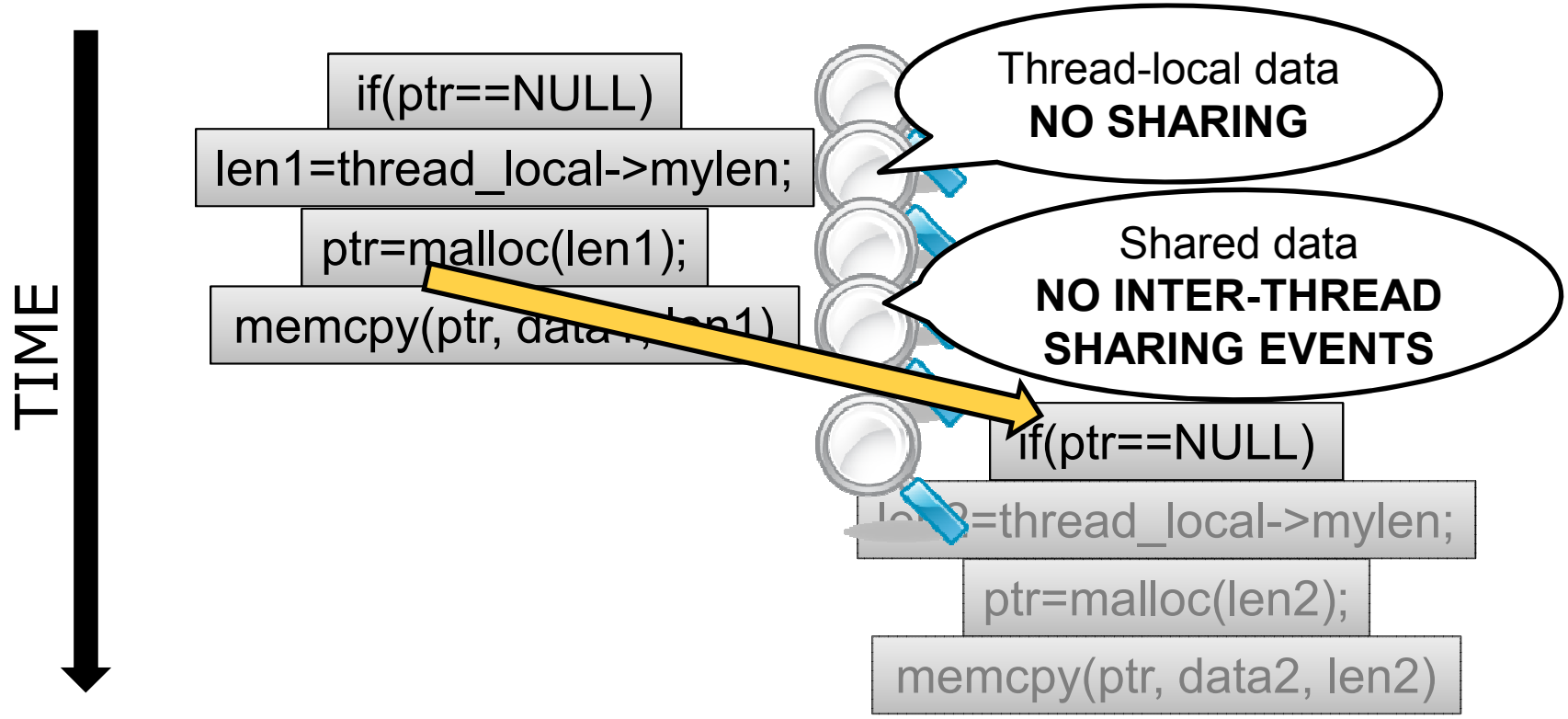
Demand-Driven Data Race Detection

Technique #2: *Keeping it Real*

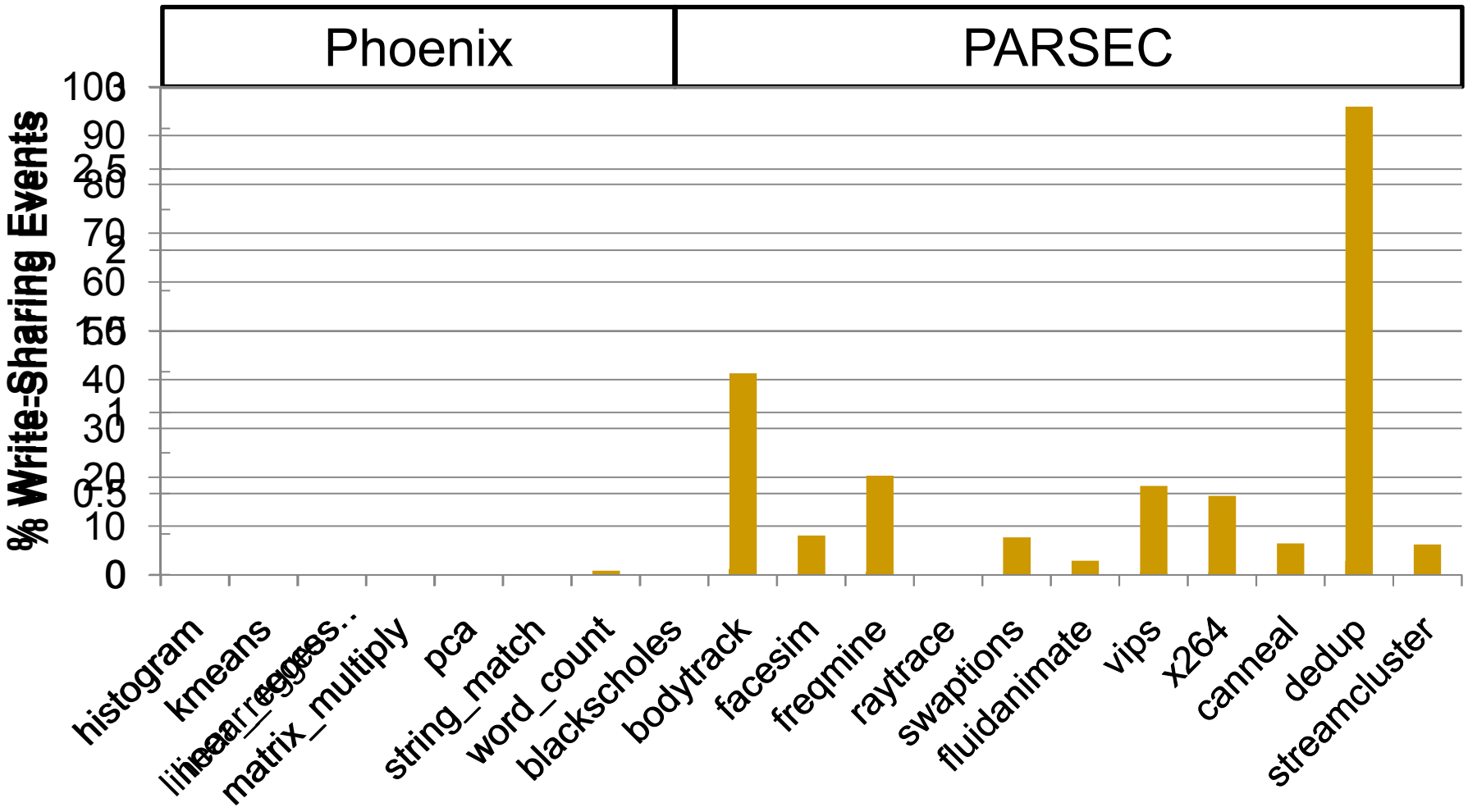
Find sharing events with existing HW

Inter-thread Sharing is What's Important

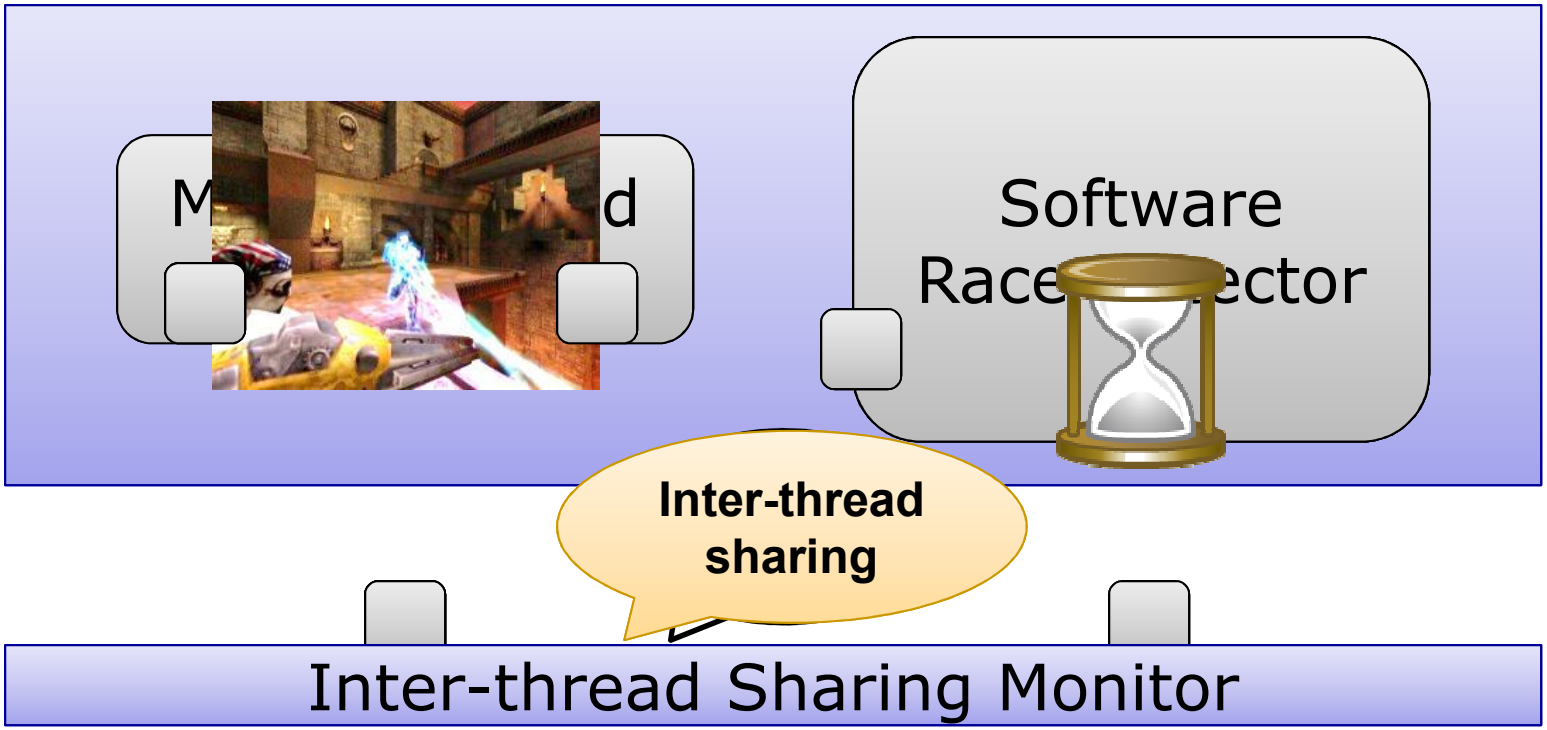
"Data races ... are failures in programs that **access and update shared data** in critical sections" – Netzer & Miller, 1992



Very Little Inter-Thread Sharing



Technique 1: Demand-Driven Analysis

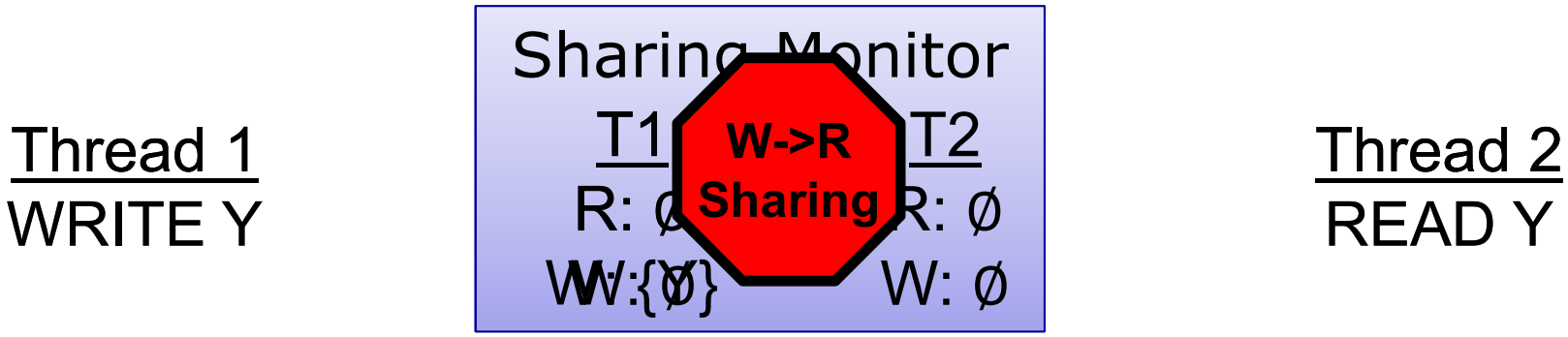


Inter-thread Sharing Monitor

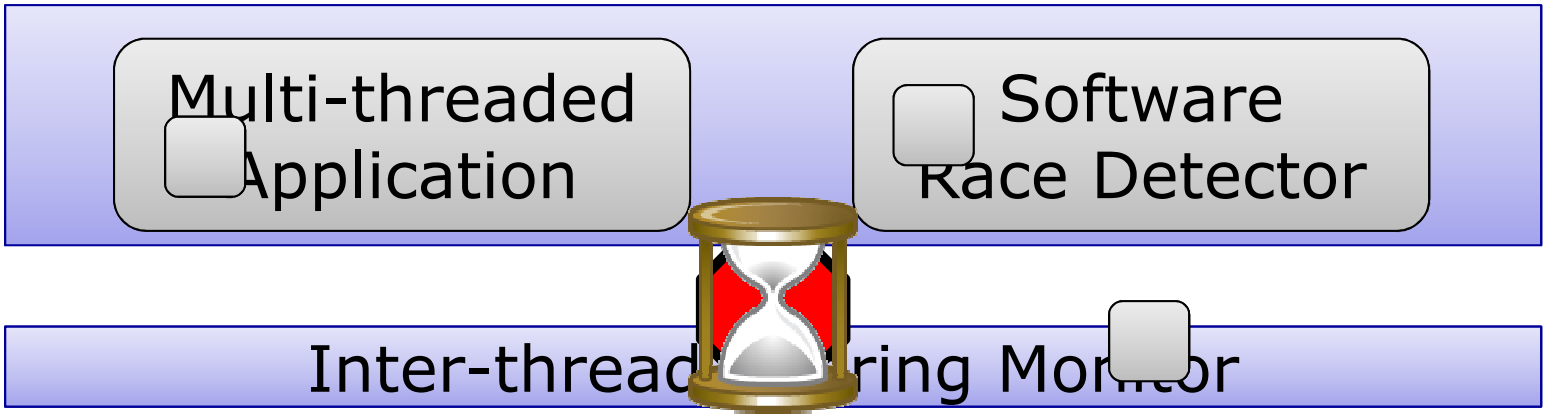
- Check each memory op. for write-sharing
- Signal software race detector on sharing
- Possible to do in software
  - \+ Can be built now with instrumentation
  - – Slow. May take as long as race detection

Ideal Hardware Sharing Detector

- Follow read/write sets of threads

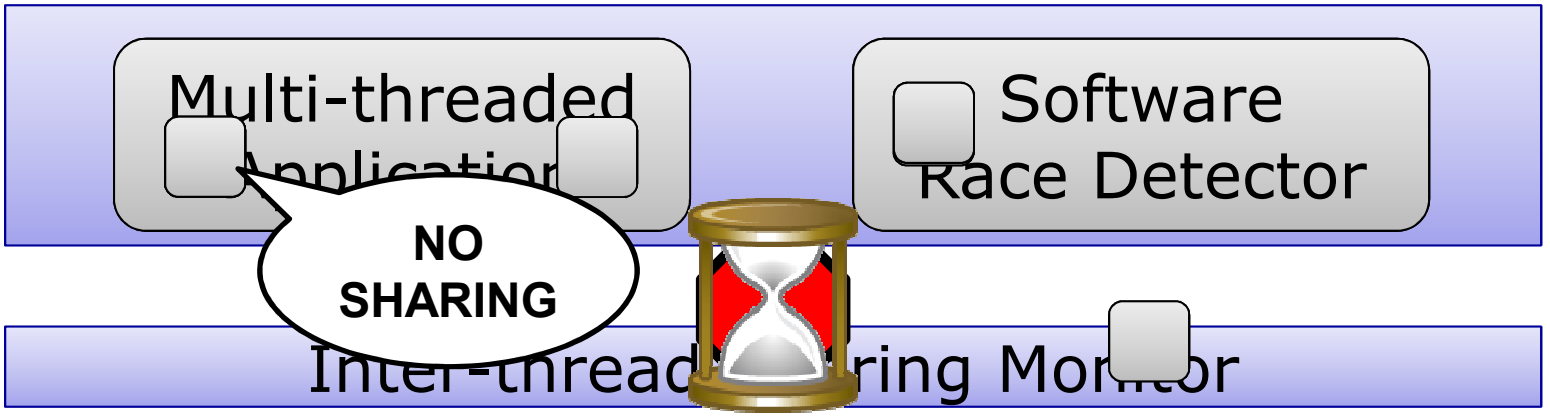


- Fast user-level faults



Limitations of Existing Hardware

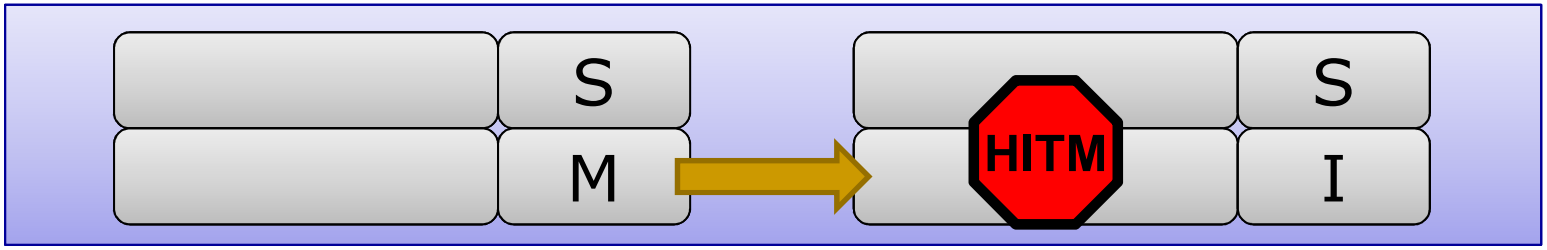
- Fast faults
  - Solution: Enable detector for long periods of time



- Read/write sets
  - Solution:

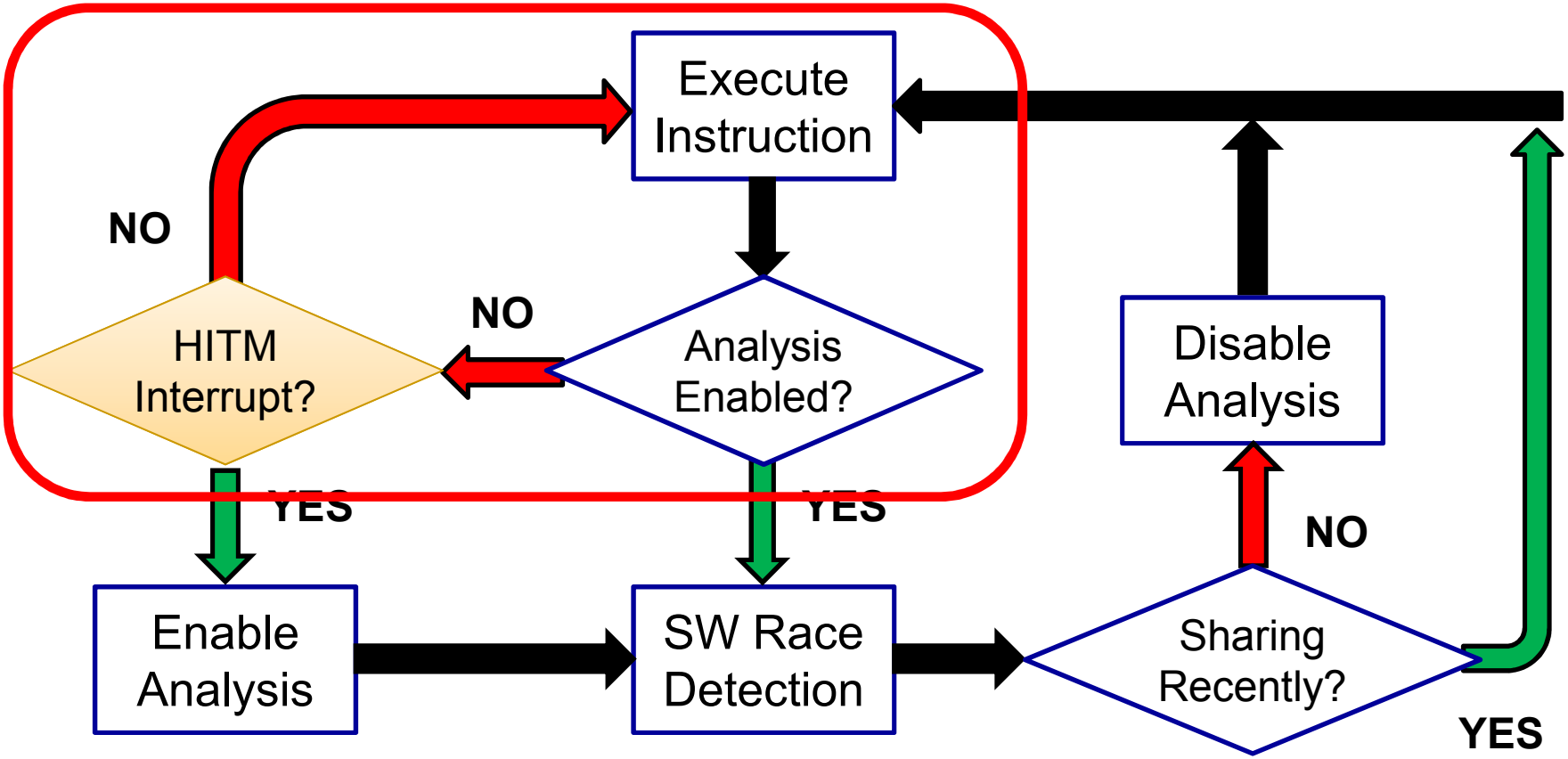
Technique 2: Hardware Sharing Detector

- Hardware Performance Counters
  - Interrupt on cache coherency events
  - Intel's HITM event: W→R Data Sharing



- Limitations of this method:
  - SMT sharing can't be counted
  - Cache eviction
  - Others in paper

Demand-Driven Analysis on Real HW

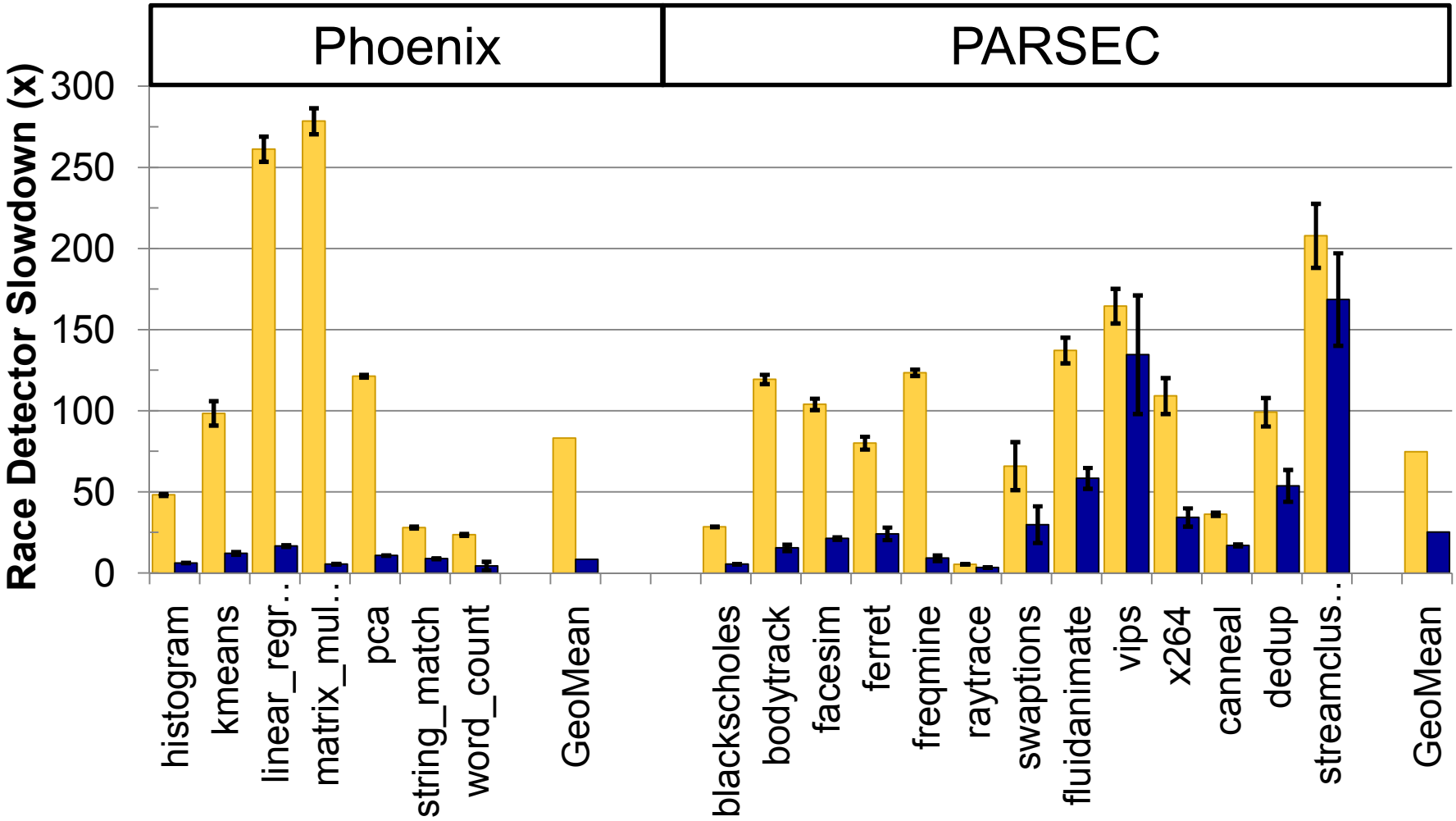


Experimental Evaluation

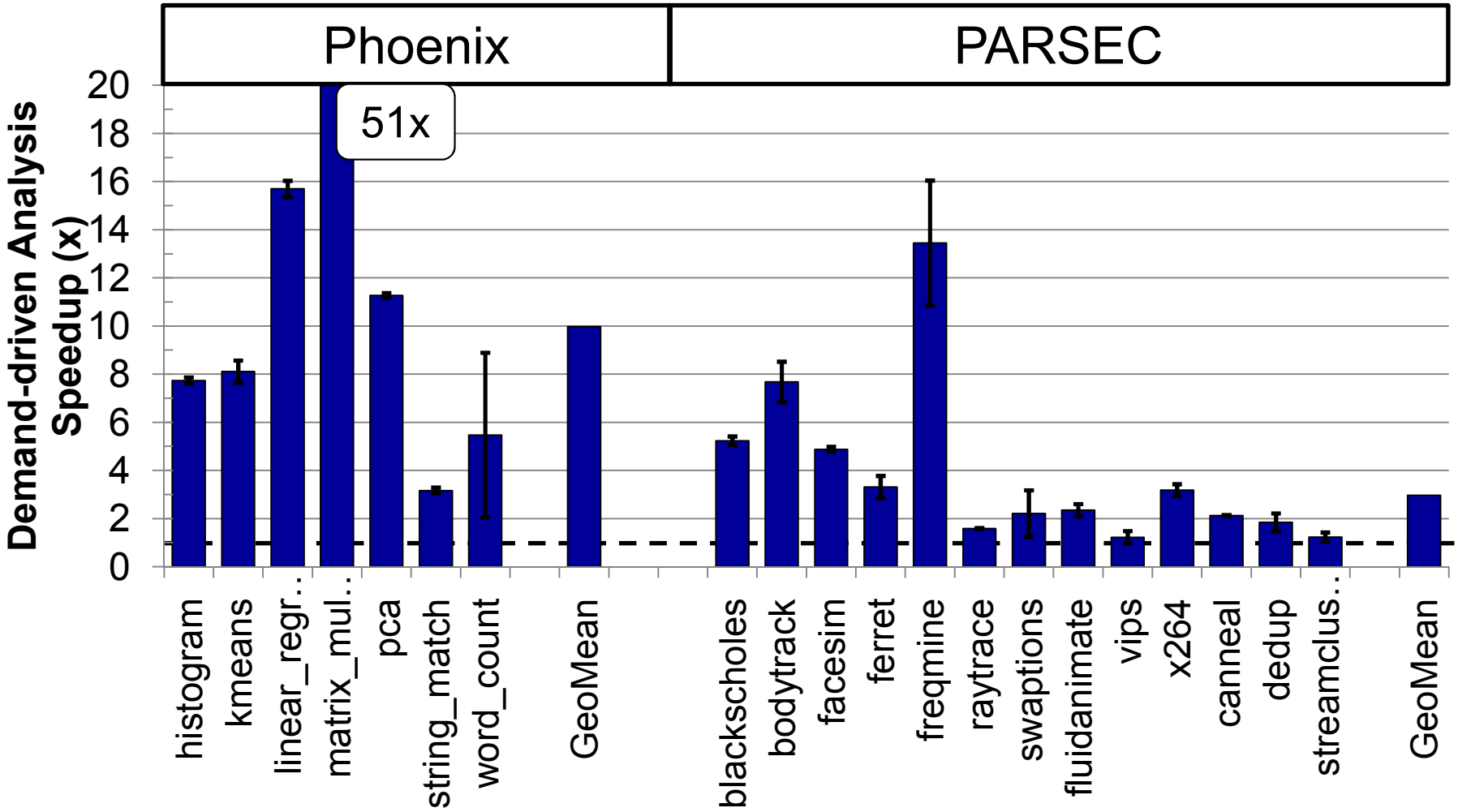
- Modified Intel Inspector XE Race Detector
- Linux on 4-core Core i7, no Hyper-Threading

- Performance Tests:
  - Phoenix Suite
  - PARSEC
- Accuracy Tests:
  - Phoenix Suite
  - PARSEC
  - Pre-release version of RADBench

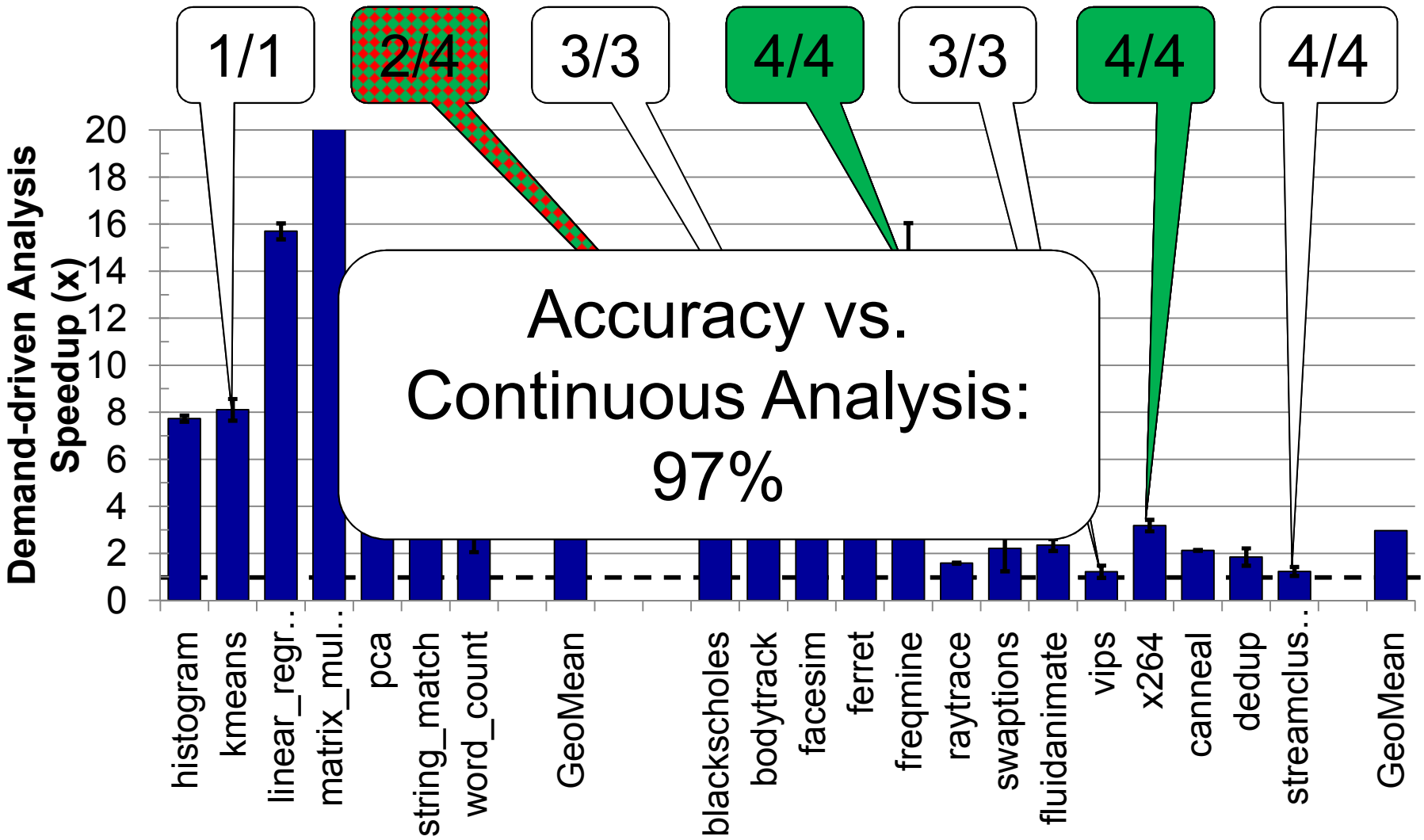
Performance Difference



Performance Increases



Demand-Driven Analysis Accuracy



Future Directions

- Better Performance
  - Fast user-level faults
  - Application specific hardware
- More Accuracy
  - Better performance counters
  - Inform SW on cache evictions/misses
- Smooth transition to ideal hardware

- Combine sampling & demand-driven analysis