Software bugs cause big problems
- Account for as much as 40% of computer system failures
- Cost the US economy an estimated $59.5 billion annually

Most existing dynamic monitoring methods are ineffective and inefficient
- Induce large overhead (10X-100X)
- Hard to find the root cause of the bug due to imperfect pointer analysis (as shown in the example)

The bug is detected at line B instead of after line A!
- Cannot detect program-specific bugs

Software is ineffective and inefficient

Hardware is used to detect triggering accesses and trigger monitoring functions on the fly

Programmers can use architectural support for software debugging

PC-based invariants
- Training: Collect the set of PCs accessing a monitored object
  - Normal run: AccSet(total) = {PC1, PC2}
  - Bug-detection: If an unusual PC is detected → likely bug

A bug will be caught by AccMon when PC5 accesses total object during detection!

A good framework for general-purpose debugging
- Low overhead profiling
- Performance debugging
- Interfacing to interactive debuggers

The bug will be caught at line A by iWatcher!

The bug will be missed by many tools using programming rules!

Advantages of iWatcher
- Catches hard-to-find bugs by monitoring all accesses to the watched memory locations
- Induces low execution overhead with architectural support
- Flexible, language independent, cross-module and cross-developer
- A good framework for general-purpose debugging
  - Low overhead profiling
  - Performance debugging
  - Interfacing to interactive debuggers

What is iWatcher?
- iWatcher (Intelligent Watcher) provides efficient and flexible architectural support for software debugging
- iWatcher monitors all accesses to the watched memory locations with low overhead

What is AccMon?
- AccMon is an automatic tool to detect memory-related bugs
  - Use PC-based invariants to detect bugs
  - Use architectural support to reduce overhead

Idea of AccMon?
- PC-based invariants
  - Training: Collect the set of PCs accessing a monitored object
  - Normal run: AccSet(total) = {PC1, PC2}
  - Bug-detection: If an unusual PC is detected → likely bug

The bug will be caught by AccMon when PC5 accesses total object during detection!

Architectural support
- Leverage iWatcher with a simple extension called CheckLook-aside Buffer (CLB)
  - CLB: filter most valid accesses to monitored objects
    - Small cache to store AccSets
    - Bloom filter: space efficiency and fast lookup

Architectural Design of iWatcher
- Hardware is used to detect triggering accesses and trigger monitoring functions on the fly
- Software is used to manage the associations between watched locations and monitoring functions

Bug Detection & Overhead
- iWatcher detects all the evaluated bugs with low overhead (4-80%)