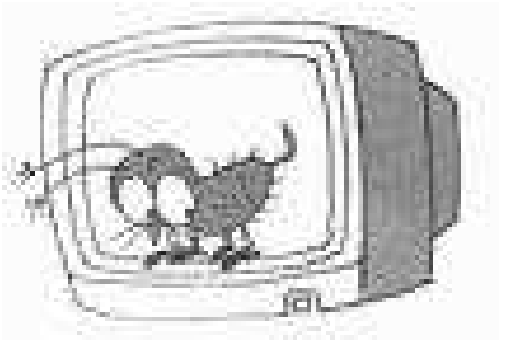
Architectural Support for Software Bug Detection

Yuanyuan (YY) Zhou and Josep Torrellas

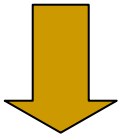
University of Illinois at Urbana-Champaign

{yyzhou,torrellas}@cs.uiuc.edu



Why Architectural Support?

- Motivation
  - Ever so many transistors in a single chip
  - Performance is reasonably good for some applications
  - Software bugs accounts for 40% of system failures
  - Limitations with software-only approaches



- New opportunity: Using a few transistors for software bug detection
  - New trade-offs between performance, robustness, etc
  - New types of bugs detected
  - New possibility: detect bugs on production-runs

Existing Hardware Support

- WatchPoint Registers
  - Mainly used for interactive debugging
  - Only support several (4 in x86) watched locations
  - Raise expensive OS exceptions upon accesses to watched locations

Recent Solutions From Our Group (1)

- ReEnact [ISCA'03]
  - Goal: Detect and correct data races on the fly
  - Idea: Dynamically analyze memory accesses to shared data and capture violations to happen-before order
  - Overhead: 1%-13%

- iWatcher [ISCA'04]
  - Goal: Efficiently and accurately monitor memory accesses to detect bugs and attacks
  - Idea
    - Allow programmers or automated tools to associate monitoring functions with monitored locations
    - When a monitored location is accessed, the monitoring function is automatically executed on-the-fly without going through OS
  - Overhead: 4% -80%

Recent Proposals From Our Group (2)

- AccMon [Micro'04]
  - Goal: Detect general memory bugs
  - Idea:
    - **PC-based invariants**: the small set of PCs that access a given memory location
    - Use hardware support for efficient invariant extraction and detection
  - Overhead: 24% - 288%
- SafeMem [HPCA'05]
  - Goal: Detect memory leaks and corruption on production runs
  - Idea: exploit ECC-memory to prune false positives in memory leak detection and detect memory corruptions
  - Overhead: 1-29%
- Undo support (presented this morning)

Proposals by Others

- Flight Data Recorder (Wisconsin-ISCA'03)
- Reduced Flight Data Recorder (UCSD-ISCA'05)
- Dynamic Instrumentation via DISE (UPenn-HPCA'05)

Advantages of Hardware Support

- Efficiency
  - Low overhead
  - Can potentially be used for production runs
- Accuracy
  - Accurate execution information (e.g. all true memory accesses
  - Accurate thread inter-leaving information
- Portability
  - Language independent
  - Cross-library, cross-modules and cross-programmers
  - Applicable to low-level code (e.g. OS)

Main Disadvantages

- Require hardware extension
- One more barrier for deployment
- Mostly limited to dynamic analysis
- Cannot work alone--- require software cooperation

Deployment in Future Processors

- Challenge: need to convince hardware vendors
- Solution:
  - Architecture and debugging communities collaborate to
    - Identify the perfect match between debugging needs and architectural support
  - Software company pushes processor companies with incentives
    - Testing machines with higher prices
    - Lowering license fees for customers that purchase testing machines
  - More research exploration on this topic
- What architectural supports will likely succeed?
  - Simple, general, reconfigurable, leverage existing hardware

Conclusions

- Message 1:
  - Architectural support opens up a new possibility in software bug detection
- Message 2:
  - The two communities should work together to make deploy such support in future processors