Precise and Accurate Processor Simulation

Harold Cain, Kevin Lepak, Brandon Schwartz, and Mikko H. Lipasti

University of Wisconsin—Madison

http://www.ece.wisc.edu/~pharm
Performance Modeling

- Analytical models
- Queuing models
- Simulation
  - Trace-driven
  - Execution-driven
  - Full system
- Why?

Perceived accuracy and precision
Precision, Accuracy, Flexibility

- **Precision**
  - How closely simulator matches design
    - Latency, bandwidth, resource occupancy, etc.

- **Accuracy**
  - How closely simulation matches reality
    - Requires precision
    - Also requires replication of real-world conditions, inputs

- **Flexibility?**
  - Enables exploration of broad design space
Uses for Simulation

Academic Research

Design Space Exploration

Quantitative Tradeoff Analysis

Performance Validation

Accuracy???

Flexibility

Precision

High-level Design

Microarchitectural Definition

Design and Implementation

Verification
Causes of Inaccuracy

- Many possible causes
  - Software differences
  - Hardware differences
  - System effects
  - Time dilation: interaction with physical world

- Here, we consider:
  - Operating system code
  - DMA traffic (in paper)
  - Wrong-path effects
Validating Accuracy

- How do we validate?
  - Against real hardware with perf. counters
    - Different “input” since O/S now present
  - Against HDL
    - Same input as timer model, same error?
- Without full system simulation, cannot:
  - Replicate runtime environment
  - Cannot really validate accuracy
- Compensating errors mask inaccuracy
- Hence: build simulator that does not cheat
PharmSim Overview

- Device simulation, etc. from SimOS-PPC
- PharmSim replaces functional simulators
  - Full OOO core model, values in rename registers
  - Based on SimpleMP [Rajwar]
    - Adds VM, TLB, exceptions, interrupts, barriers, etc.
PharmSim Pipeline

- Substantially similar to IBM Power4
  - Some instructions “cracked” (1:2 expansion)
  - Others (e.g. lmw) microcode stream

- Mem Stage
  - Interface to 2-level cache model
  - Sun Gigaplane XB snoopy MP coherence
  - Caches contain values, must remain coherent

- No cheating!
  - No “flat” memory model for reference/redirect

February 2, 2002 Precise and Accurate Processor Simulation--Mikko Lipasti
Operating System Effects

- Fairly well-understood for commercial:
  - Must account for O/S references
- For SPEC? Widely accepted:
  - Safe to ignore O/S paths
  - Most popular tool (Simplescalar)
    - Intercepts system calls
    - Emulates on host, updates “flat” memory
    - Returns “magically” with caches intact
- Is this really OK?
## Operating System Effects

<table>
<thead>
<tr>
<th>References Modeled</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>User-mode only</td>
<td>Atom</td>
</tr>
<tr>
<td>User + Shared library</td>
<td>Simplescalar with static link</td>
</tr>
<tr>
<td>User + Sh Lib + O/S</td>
<td>H/W bus trace</td>
</tr>
<tr>
<td>User + Sh Lib + O/S + cache control ops</td>
<td>PharmSim</td>
</tr>
</tbody>
</table>
Operating System Effects

- Dramatic error (5.8x in mcf, 2-3x commonplace)
- Note compensating errors (e.g. crafty, gzip, perl)
- IPC error > 100% (more detail at ISCA)
Wrong-path Execution

- Multiple effects on unarchitected state
  - Pollute/prefetch I-cache, D-cache, TLB
  - Pollute/train branch predictor (BHR, PHT, RAS)

- PharmSim:
  - BHR is updated and repaired
  - PHT is not updated speculatively
  - RAS is updated, no repair
  - No speculative TLB fill

- How can we filter wrong-path instructions?
  - No “cheating”: don’t know branch outcomes

- 25% - 40% instructions are wrong-path
Wrong-path Memory Stalls

- Minor effect: better or worse

February 2, 2002

Precise and Accurate Processor Simulation--Mikko Lipasti
Wrong-path RAS Accuracy

- Prediction accuracy degrades up to 29%
- Could add fixup logic
Wrong-path IPC

Negligible effect (0.9%)
- RAS mispredictions overlapped

February 2, 2002  Precise and Accurate Processor Simulation--Mikko Lipasti
Summary

- PharmSim
  - Simulator that does not cheat
  - Can be used to validate assumptions, simplifications, abstractions
- Evaluated three effects on accuracy
  - O/S: dramatic error, even for SPECINT
  - DMA: not important for uniprocessors
    - MP, bus-constrained results TBD
  - Wrong path: unimportant
Conclusions

- Ignoring O/S effects fraught with danger
  - Should **always** model O/S effects
- Trace-driven vs. execution-driven
  - Traces with O/S much better
  - Invest in
    - **Trace quality** vs.
    - **Complexity** of execution-driven simulation
- Precision without accuracy?
  - Of questionable value
- Validation difficult due to compensating errors
  - Hard to know if model is precise or accurate
Wrong-path Instructions

- Aggressive core model; 25%-40% wrong-path
DMA Traffic

- How do we support DMA?
  - No “flat” memory image in simulator
  - Lines may be in caches
    - Invalidate
    - Read
- Must use existing coherence
  - Everything has to work correctly
    - No subtle coherence bugs
- How much does this matter?
  - Affects cache miss rates
  - Introduces bus contention
DMA Traffic

- PharmSim incorporates accurate DMA engine:
  - Issues bus invalidates, snoops
  - Concurrent data transfer: No “magic” flat memory

- Bottom line:
  - Unimportant for SPEC
  - Unimportant for SPECWEB, SPECJBB
    - Others in progress
  - Contrived multiprogrammed workload
    - 4.8% of all coherence traffic due to I/O, 1% IPC effect

- Results understated due to “overbuilt” MP bus
  - MP workloads likely much more sensitive
  - Additional evaluation in progress