SNIPER: EXPLORING THE LEVEL OF ABSTRACTION FOR SCALABLE AND ACCURATE PARALLEL MULTI-CORE SIMULATION

TREVOR E. CARLSON, WIM HEIRMAN, LIEVEN ECKHOUT

TREVOR.CARLSON@ELIS.UGENT.BE
HTTP://WWW.ELIS.UGENT.BE/~TCARLSON
WEDNESDAY, NOVEMBER 16TH, 2011
SC11, SEATTLE, WA
**Need for High-Abstraction Models**

We are here today, but how do we get to the next step?

- **Core/Tile:** 1 to 10 threads
- **Die:** 100 to 1000 threads
- **Socket/Blade:** 500 to 5000 threads
- **Rack:** $10^4$ to $10^5$ threads
- **HPC Cluster:** $10^9$ threads
- **1Pflop/s**
- **1Eflop/s**
- **1 to 10Gflop/s**

Photos used under Creative Commons from Flickr users jeremybrooks, st-stev, bike, pchow98 and brookhavenlab
DEMANDS ON SIMULATION ARE INCREASING

• Number of cores per node is increasing (100s)
  – Requiring parallel simulation to keep pace

• Cache sizes are increasing
  – Requires longer benchmark runtimes
**Time To Market Matters**

- **Industry**
  - Tend to have enough compute bandwidth
  - Simulation latency limited
  - Results should be ready when you arrive at work in the morning

- **Academia**
  - Not enough compute power
  - Both latency and bandwidth limited
  - Exploration and investigation work

- **How many scenarios can I run?**
  - N = total number of simulation scenarios
  - d = days until SC deadline
  - t = average time per simulation
  - B = number of benchmarks
  - A = number of architectures

\[ N = \frac{d}{t \times B \times A} \]

minimize \( t \) to maximize \( N \)
Fast and Accurate Simulation is Needed

• Sniper Simulator
  – Interval core model
  – Accurate structures (caches, branch predictors, etc.)
  – Parallel simulator scales with the number of simulated cores

• Key Questions
  – What is the right level of abstraction?
  – When to use these abstraction models?
• Motivation
• Research Background
• Experimental Setup
• Results
• Conclusions
**Needed Detail Depends on Focus**

<table>
<thead>
<tr>
<th>Component</th>
<th>Single-event time scale</th>
<th>Required sim time</th>
</tr>
</thead>
<tbody>
<tr>
<td>RTL</td>
<td>single clock cycle</td>
<td>millions of cycles</td>
</tr>
<tr>
<td>OOO execution</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Core memory ops</td>
<td></td>
<td></td>
</tr>
<tr>
<td>L1 cache access</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LLC access</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Off-socket</td>
<td>microseconds</td>
<td>seconds</td>
</tr>
</tbody>
</table>

**Too slow**

- cycle-accurate models
- interval core model

**Not accurate enough**

- simple core models
One-IPC Modeling — Too Simple?

• Simple high-abstraction model
  – Widely used

• One-IPC modeling
  – Scalar, in-order issue
  – Account for non-unit instruction exec latencies
  – Perfect branch prediction
  – L1 D-cache accesses are assumed to be hidden
  – All other cache accesses incur penalty
• Out-of-order core performance model with in-order simulation speed

D. Genbrugge et al., HPCA’10
S. Eyerman et al., ACM TOCS, May 2009
T. Karkhanis and J. E. Smith, ISCA’04, ISCA’07
KEY BENEFITS OF THE INTERVAL MODEL

• Models superscalar OOO execution
• Models impact of ILP
• Models second-order effects: MLP

• Allows for constructing CPI stacks
MULTI-CORE INTERVAL SIMULATION

memory hierarchy simulator

branch predictor simulator

functional simulator

processor cores

next instruction to dispatch

old window

upcoming instructions

dispatched instructions
CORE-LEVEL TIMING
NO MISS EVENTS

Instantaneous dispatch rate is determined by the longest critical path in the old window:

Instantaneous dispatch rate = \( \min \left( \frac{W}{L}, D \right) \)

Little’s law
Assumes a balanced architecture

\( L \) = longest critical path length in cycles
\( W \) = instructions in the old window (max = ROB length)
\( D \) = maximum dispatch rate (processor width)
LONG BACK-END MISS EVENTS

ISOLATED LONG-LATENCY LOAD

S. Eyerman et al., ACM TOCS, May 2009
LONG BACK-END MISS EVENTS

OVERLAPPING LONG-LATENCY LOADS

S. Eyerman et al., ACM TOCS, May 2009
If long-latency load (LLC miss):

\[
\text{core sim time } += \text{ miss-latency}
\]

**AND** walk the window to issue independent miss events:
these are hidden under the long-latency load – second-order effects
**Cycle Stacks**

- Where did my cycles go?
- CPI stack
  - Cycles per instruction
  - Broken up in components
- Normalize by either
  - Number of instructions (CPI stack)
  - Execution time (time stack)
- Different from miss rates: cycle stacks directly quantify the effect on performance
**Constructing CPI Stacks**

- Interval simulation: track why time is advanced
  - No miss events
    - Issue instructions at base CPI
    - Increment base component
  - Miss event
    - Fast-forward time by X cycles
    - Increment component by X

---

- L2 cache
- I-cache
- Branch
- Base
**EXPERIMENTAL SETUP**

- **Benchmarks**
  - Complete SPLASH-2 suite
    - 1 to 16 threads
    - Linux pthreads API

- **Graphite simulation infrastructure**
  - Fast simulation
  - Parallel and modular base
  - Pin-based user-level functional-first simulation
EXPERIMENTAL SETUP: ARCHITECTURE
SNIPER SIMULATION ENVIRONMENT

• User-level, x86-64, parallel
• Branched from MIT Graphite in August 2010
• Adds interval core model, CPI stacks, modern branch predictor, shared cache models, DVFS, OpenMP support, etc.
• Hardware-validated against a 16-core Intel Xeon X7460 Dunnington machine
SNIPER SIMULATION ENVIRONMENT

- Code will be available this week
- Open source (MIT license, interval model with academic license) available at

http://snipersim.org
SIMULATION CONSIDERATIONS

• Is a one-IPC core model accurate enough?

• What are the available abstraction levels?

• What is the speed/accuracy trade-off for each level?
The interval core model provides consistent accuracy of 25% avg. abs. error, with a minimal slowdown.
INTERVAL: GOOD OVERALL ACCURACY

Good accuracy for the entire benchmark suite
Sniper currently scales to 2 MIPS

Typical simulators run at 10s-100s KIPS, without scaling
Variability due to relaxed synchronization is application specific
FLEXIBILITY TO CHOOSE NEEDED FIDELITY
CONCLUSIONS

• Faster simulation methodologies are needed to bridge the gap to 100s and 1000s of cores per socket
• Many performance/accuracy trade-offs, and application and optimization-target specific
• The interval model is a good option for fast and accurate core simulation
  – Simulated vs. HW error of only 25% for 16-thread applications
• Download the Sniper simulator soon at snipersim.org
SNIPER: EXPLORING THE LEVEL OF ABSTRACTION FOR SCALABLE AND ACCURATE PARALLEL MULTI-CORE SIMULATION

TREVOR E. CARLSON, WIM HEIRMAN, LIEVEN EEEKHOUT

snipersim.org

TREVOR.CARLSON@ELIS.UGENT.BE
HTTP://WWW.ELIS.UGENT.BE/~TCARLSON
WEDNESDAY, NOVEMBER 16TH, 2011
SC11, SEATTLE, WA