ZSim Tutorial

Validation
Introduction

- How accurate is a simulator?
- What are the sources of inaccuracies?
- What kind of workloads and studies is a simulator intended for?
- Important to do validation before using a simulator.

Tony Nowatzki et.al, Architectural Simulators Considered Harmful, IEEE MICRO 2015
Validation in ZSim

- Micro-benchmarks that stress different micro-architectural structures and events.
  - Ex. Time taken to do integer add, multiply.
  - Lets us catch even minor modeling inaccuracies.

- Wide range of workloads from different benchmark suites
  - Single threaded – SPECCPU2006
  - Multi threaded – PARSEC, SPLASH2, SPECCOMP 2001
Comparison to other simulators

- ZSim has an average error of 10% for both single-threaded and multi-threaded workloads.

- **MARSS**
  - Cycle accurate OOO x86 model
  - Performance differences range from -59% to 50% with only 5 benchmarks being within 10%

- **Sniper**
  - Approximate OOO model
  - Absolute errors over 50% on SPLASH2 benchmarks

- **Graphite, Hornet, SlackSim** – no known validation study
Methodology

- Zsim models an x86 core model.
  - It is possible to validate against real hardware system.
- We run each application on the real machine and also simulate it on zsim.
- We record several relevant performance counters on the real machine.
  - Compare them against zsim’s results.
- We perform multiple profiling and simulation runs to avoid noisy comparisons.
System Configuration

We validate ZSim against a Westmere system.

<table>
<thead>
<tr>
<th>HW</th>
<th>Xeon L5640 (6-core Westmere), 24GB DDR3-1333, no hyperthreading, turbo/DVFS disabled</th>
</tr>
</thead>
<tbody>
<tr>
<td>SW</td>
<td>Linux 3.5 x86-64, gcc 4.6.2, Pin 2.12</td>
</tr>
<tr>
<td>Bound-weave</td>
<td>1000-cycle intervals, 6 weave threads</td>
</tr>
<tr>
<td>Cores</td>
<td>6 x86-64 OOO cores at 2.27GHz</td>
</tr>
<tr>
<td>L1I caches</td>
<td>32KB, 4-way, LRU, 3-cycle latency</td>
</tr>
<tr>
<td>L1D caches</td>
<td>32KB, 8-way, LRU, 4-cycle latency</td>
</tr>
<tr>
<td>L2 caches</td>
<td>256KB, 8-way, LRU, 7-cycle latency, private</td>
</tr>
<tr>
<td>L3 cache</td>
<td>12MB, 16-way, hashed, 6 2MB banks, 14-cycle bank lat, shared, inclusive, MESH coherence w/in-cache directory, 16 MSHRs</td>
</tr>
<tr>
<td>Network</td>
<td>Ring, 1-cycle/hop, 5-cycle injection latency</td>
</tr>
<tr>
<td>Mem ctrl</td>
<td>1 controller, 3 DDR3 channels, closed page, FCFS scheduling, fast powerdown with threshold timer = 15 mem cycles</td>
</tr>
<tr>
<td>DRAM</td>
<td>24GB, DDR3-1333, 2 4GB RDIMMs per channel</td>
</tr>
</tbody>
</table>

Hardware and Software Configuration of the real system and the corresponding ZSim configuration
## Single-threaded validation

- Validate OOO core model with the full SPEC CPU2006 suite.
- Run each application for 50 billion instructions using ref(largest) input set.

---

### Comparing instruction range: 0-50 B

<table>
<thead>
<tr>
<th>App</th>
<th>Real/Inst IPC</th>
<th>Real/Inst UPC</th>
<th>PerfErr</th>
<th>R/S</th>
<th>BrMPKI</th>
<th>UopErr</th>
<th>approxIns</th>
</tr>
</thead>
<tbody>
<tr>
<td>434.zeusmp-ref</td>
<td>1.05</td>
<td>1.05</td>
<td>1.22</td>
<td>1.25</td>
<td>0.00</td>
<td>0.18</td>
<td>0.32</td>
</tr>
<tr>
<td>459.Gem5PFD-ref</td>
<td>0.66</td>
<td>0.66</td>
<td>0.85</td>
<td>0.89</td>
<td>0.00</td>
<td>0.19</td>
<td>0.24</td>
</tr>
<tr>
<td>473.istar-ref</td>
<td>0.56</td>
<td>0.56</td>
<td>0.63</td>
<td>0.62</td>
<td>0.00</td>
<td>38.82</td>
<td>45.65</td>
</tr>
<tr>
<td>418.waves-ref</td>
<td>1.49</td>
<td>1.48</td>
<td>1.78</td>
<td>1.77</td>
<td>0.01</td>
<td>4.64</td>
<td>4.14</td>
</tr>
<tr>
<td>433.mlcc-ref</td>
<td>0.64</td>
<td>0.63</td>
<td>0.74</td>
<td>0.73</td>
<td>0.01</td>
<td>0.10</td>
<td>0.15</td>
</tr>
<tr>
<td>464.bar2-ref</td>
<td>1.85</td>
<td>1.87</td>
<td>2.17</td>
<td>2.38</td>
<td>-0.01</td>
<td>1.56</td>
<td>1.56</td>
</tr>
<tr>
<td>454.calcilix-ref</td>
<td>1.71</td>
<td>1.74</td>
<td>1.81</td>
<td>1.84</td>
<td>-0.01</td>
<td>1.23</td>
<td>1.51</td>
</tr>
<tr>
<td>456.nmer-ref</td>
<td>1.82</td>
<td>1.70</td>
<td>2.63</td>
<td>2.59</td>
<td>0.02</td>
<td>0.19</td>
<td>0.21</td>
</tr>
<tr>
<td>445.gobmk-ref</td>
<td>1.13</td>
<td>1.11</td>
<td>1.38</td>
<td>1.27</td>
<td>0.02</td>
<td>0.29</td>
<td>23.28</td>
</tr>
<tr>
<td>481.bzip2-ref</td>
<td>1.21</td>
<td>1.29</td>
<td>1.48</td>
<td>1.48</td>
<td>-0.06</td>
<td>11.67</td>
<td>12.75</td>
</tr>
<tr>
<td>416.gromacs-ref</td>
<td>1.88</td>
<td>1.77</td>
<td>2.26</td>
<td>2.12</td>
<td>0.06</td>
<td>1.36</td>
<td>1.72</td>
</tr>
<tr>
<td>462.libquantum-ref</td>
<td>0.55</td>
<td>0.59</td>
<td>0.48</td>
<td>0.52</td>
<td>-0.06</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>481.lwrf-ref</td>
<td>1.84</td>
<td>2.08</td>
<td>1.78</td>
<td>1.86</td>
<td>-0.08</td>
<td>2.12</td>
<td>2.46</td>
</tr>
<tr>
<td>483.gcc-ref</td>
<td>1.10</td>
<td>1.02</td>
<td>1.31</td>
<td>1.19</td>
<td>0.08</td>
<td>7.43</td>
<td>13.47</td>
</tr>
<tr>
<td>444.namd-ref</td>
<td>1.62</td>
<td>1.77</td>
<td>1.81</td>
<td>1.97</td>
<td>-0.08</td>
<td>1.86</td>
<td>1.78</td>
</tr>
<tr>
<td>465.tonto-ref</td>
<td>1.80</td>
<td>1.97</td>
<td>2.25</td>
<td>2.30</td>
<td>-0.08</td>
<td>1.18</td>
<td>1.80</td>
</tr>
<tr>
<td>458.sjeng-ref</td>
<td>1.46</td>
<td>1.32</td>
<td>1.60</td>
<td>1.51</td>
<td>0.10</td>
<td>0.79</td>
<td>17.23</td>
</tr>
<tr>
<td>447.dealII-ref</td>
<td>1.52</td>
<td>1.69</td>
<td>1.91</td>
<td>2.12</td>
<td>-0.10</td>
<td>3.09</td>
<td>3.07</td>
</tr>
<tr>
<td>400.perlbench-ref</td>
<td>1.62</td>
<td>1.47</td>
<td>1.87</td>
<td>1.89</td>
<td>0.11</td>
<td>5.87</td>
<td>5.68</td>
</tr>
<tr>
<td>450.soplex-ref</td>
<td>0.58</td>
<td>0.66</td>
<td>0.67</td>
<td>0.74</td>
<td>-0.12</td>
<td>5.20</td>
<td>5.80</td>
</tr>
<tr>
<td>482.sphinx3-ref</td>
<td>1.33</td>
<td>1.52</td>
<td>1.63</td>
<td>1.85</td>
<td>-0.12</td>
<td>3.33</td>
<td>2.99</td>
</tr>
<tr>
<td>435.gromacs-ref</td>
<td>1.12</td>
<td>1.31</td>
<td>1.37</td>
<td>1.59</td>
<td>-0.14</td>
<td>0.82</td>
<td>0.83</td>
</tr>
<tr>
<td>437.leslie3d-ref</td>
<td>0.69</td>
<td>0.69</td>
<td>0.68</td>
<td>0.80</td>
<td>0.15</td>
<td>0.13</td>
<td>0.72</td>
</tr>
<tr>
<td>429.mcf-ref</td>
<td>0.44</td>
<td>0.38</td>
<td>0.50</td>
<td>0.43</td>
<td>0.17</td>
<td>15.23</td>
<td>18.29</td>
</tr>
<tr>
<td>446.gcc-ref</td>
<td>0.86</td>
<td>0.83</td>
<td>1.30</td>
<td>1.50</td>
<td>-0.17</td>
<td>0.82</td>
<td>0.83</td>
</tr>
<tr>
<td>483.xalanbmk-ref</td>
<td>0.86</td>
<td>1.05</td>
<td>0.95</td>
<td>1.14</td>
<td>-0.18</td>
<td>1.08</td>
<td>5.74</td>
</tr>
<tr>
<td>471.omnetpp-ref</td>
<td>0.71</td>
<td>0.94</td>
<td>0.76</td>
<td>1.01</td>
<td>-0.25</td>
<td>3.75</td>
<td>5.14</td>
</tr>
<tr>
<td>478.lbm-ref</td>
<td>0.64</td>
<td>0.67</td>
<td>0.73</td>
<td>0.90</td>
<td>-0.22</td>
<td>0.05</td>
<td>0.04</td>
</tr>
</tbody>
</table>
Average absolute IPC error is 8.5%.

Max error is 26%

In 21 out of the 29 benchmarks, error is less than 10%.
MPKI Errors for different caches

Average Absolute MPKI errors

- L1i - 0.32
- L1d - 1.14
- L2 - 0.59
- L3 - 0.30
Traces

IPC Trace

L3 MPKI Trace
Major sources of error

- Does not model TLB and page table walkers.
- Inaccuracies in the front end model.
  - The modeled 2-level branch predictor with an idealized BTB has significant errors in some cases.
- Most of the errors are observed in benchmarks that have non-negligible TLB misses.
- It is difficult to figure out the exact details of a processor’s architecture.
μop coverage

- ZSim implements decoding for the most frequently used op-codes.
- Only 0.01% of executed instructions have an approximate dataflow decoding.
- Modern compilers only produce a fraction of the x86 ISA.
- Ignores micro-sequenced instructions.
- Uop error = (uop real – uop zsim )/uop real
- Average μop error is 1.3%.
Multithreaded validation

- 22 applications from different benchmark suites
  - 6 from PARSEC, 7 from SPLASH2, 9 from SPEC OMP2001

- Run most workloads at 6 threads
  - Those that need power of 2 threads run with 4 threads

- Measure performance as $1/(\text{time to completion})$ and not IPC.
- Average absolute error is 11.2%.
- 10 out of 23 workloads are within 10% error.
Contention models

- Many simulators fail to accurately model bandwidth contention.
- ZSim can accurately simulate a real hardware system by using detailed contention models.
- We study the scalability of STREAM benchmark on real machine and simulation with several timing models.
- STREAM saturates memory bandwidth, scaling sub-linearly.
Without contention, there is no bandwidth limitation and performance scales linearly.

Approximate Queueing theory model (MD1) is still quite inaccurate.

Using event-driven model or DRAMSim2 closely approximates real machine.
Accuracy vs Speed

- Bound-weave algorithm allows for modeling contention at varying degrees of accuracy.

- Tradeoff between simulation speed and accuracy
  - DRAMSim2 is cycle-accurate — limits ZSim performance to 3 MIPS.
  - Few tens of MIPS with simpler models.
Silvermont validation

- Changed a few parameters to model a silvermont like core.
- Absolute performance error of 20.89%.

- Uop decoding is slightly different.
- Much simpler branch predictor.
- We do not model
  - Differences in backend architecture.
  - Silvermont’s prefetcher.

- Possible to reduce the errors by doing more accurate modelling.
You can trust zsim to be quite accurate, but

‘If you are using zsim with workloads or architectures that are significantly different from ours, you should not blindly trust these results’

Detailed results available at

zsim.csail.mit.edu/validation

Plan to release the complete validation infrastructure in future.
THANK YOU

QUESTIONS?