An Integrated Simulation Tool for Computer Architecture and Cyber-Physical Systems

Hokeun Kim\textsuperscript{1,2}, Armin Wasicek\textsuperscript{3}, and Edward A. Lee\textsuperscript{1}

\textsuperscript{1}University of California, Berkeley
\textsuperscript{2}LinkedIn Corp. \textsuperscript{3}Technical University Vienna

CyPhy'17, Seoul, Korea
Introduction

• Many tools used for CPS modeling and simulation employs a simplified timing model for “cyber” part of CPS
  – Example tools: OpenModelica, Ptolemy II
  – E.g., computation time, communication delay

• These tools are useful
  – Faster than simulating or emulating cyber part
  – Enough for CPS simulation in many cases

• But, sometimes we need more than just simplified computation & communication models
Motivation 1 – Side Channels

- Side channel attacks
  - Gaining information by leveraging physical implementation of computer systems
  - E.g., power analysis

(a) Hamming distance from data to ref. state
(B) Power consumption

Timing delays are not enough!

Motivation 1 – Side Channels

• Cold boot attack on DRAMs

Freeze the DRAM memory of the running system to prevent the data from decaying

Read out data and look for high entropy in data (cryptographic key)

Shamir and Someren, "Playing Hide and Seek with Stored Keys", FC 99 (Conference on Financial Cryptography)

Motivation 2 – MOOC for CPS

• CPS classes
  – Involve a lot of hands-on experiments

  e.g. EECS149.1x, Cyber-Physical Systems at UC Berkeley
  https://www.edx.org/course/cyber-physical-systems-uc-berkeleyx-eecs149-1x

• MOOC for CPS classes?
  – Not like other CS classes
  – Accurate model for CPS would help
Goals

• Building a CPS simulator supporting accurate computer architecture model
• Demonstration of an open-source integrated simulation tool for CPS and computer architecture
• Case study using DRAM power and thermal modeling
• The gem5 architecture simulator (from UMich)
  – Open-source powerful, modular, flexible and widely used both in academia and industry
• Characteristics
  – Object-oriented, discrete-event
  – Modular components (CPUs, Memories, Buses, Interconnects), easily interchangeable
  – Simulated system = collection of objects
• **Ptolemy II**
  – An open-source software for research on cyber-physical systems
  – Developed at UC Berkeley since 1996
  – Supports modeling of both the cyber part (computation, communication) & physical process (continuous dynamics)
  – Quite stable, easy to learn and use (supports GUI, one can build a model by drawing components)
  – Based on **actor-oriented design**
  – More information on [http://ptolemy.org](http://ptolemy.org)
• **Actor-Oriented Design in Ptolemy II**
  
  – **Actors**
  
  • Concurrently executed components
  • Interact with other actors through input/output ports
  • Model computation, communication, physical processes, etc.
  
  – **Directors**
  
  • Implement Models of Computation (MoCs)
  • Orchestrate behavior of actors, for example, when each actor should be executed (=fired)
  
  – **Actor hierarchy**
  
  • An actor can have sub-actors
Background – Tools

• Model of Computation (MoC)
  – A set of rules orchestrating behavior of actors (e.g., when to execute actors, how actors react to inputs)

Multiple MoCs in a single Ptolemy II model

- Continuous Time
  • Sampling-based simulation, ODE solvers
  • For modeling physical processes (e.g., thermal transfer)

- Discrete Event
  • Time-stamped events (e.g. timer event, arrival of messages)
  • For modeling computation or communication
Background – DRAM Model

- DRAM thermal model by Lin et al. (ISCA `07)
  - Power is proportional to throughput (GB/s)
  - Factors that affect DRAM temperature

Lin et al., “Thermal modeling and management of DRAM memory systems” ISCA ‘07
Approach
- Integrated Tool Overview

- TerraSwarm Research Center
Approach
- gem5 as Cyber Part of CPS
Approach
– Configuring gem5 Simulator

• Implementation of gem5
  – Python – high-level object configuration & simulation
  – C++ – low-level object implementation (for performance)

• The gem5 Simulator python scripts
  – Modify execution scripts for periodic execution
  – gem5 runs for given cycles and stops
  – Resume after Ptolemy II model runs

• DRAM component
  – Add DPRINTF functions to DRAM component
  – Print out command and cycle information
Approach
– Communication between gem5 & Ptolemy II
Approach – Communication between gem5 & Ptolemy II

Gem5 Simulator

- CPU
- L1 I Cache
- L1 D Cache
- DRAM

Ptolemy II Model

- DiscreteClock
- DE Director
- Gem5Wrapper
- DRAMModel
- PowerTemperatureModel

“Fire”
(Run simulation for N cycles)

“Notify”
(Simulation finished & results ready)

Named pipe 1
Named pipe 2

Shared File
Memory trace:
<time, access type, addr>
<time, access type, addr>
<time, access type, addr>
...

Store simulation results

Load simulation results

Simulation information transferred

Java custom actor

gem5 blocks on read
Wrapper fire() blocks on read
Approach – Communication between gem5 & Ptolemy II

(1) Gem5Wrapper initialize() triggers gem5

“Fire” (Run simulation for N cycles)

“Notify” (Simulation finished & results ready)

Shared File
Memory trace:
<time, access type, addr>
<time, access type, addr>
<time, access type, addr>

Load simulation results

Store simulation results

TerraSwarm Research Center
Approach
– Communication between gem5 & Ptolemy II

TerraSwarm Research Center

gem5 Simulator

CPU
L1 I Cache
L1 D Cache
DRAM

Ptolemy II Model

“Fire”
(Run simulation for N cycles)

“Notify”
(Simulation finished & results ready)

Named pipe 1
Named pipe 2

Gem5 Wrapper Actor

Shared File
Memory trace:
<time, access type, addr>
<time, access type, addr>
<time, access type, addr>
...

(2) gem5 runs

Store simulation results

Load simulation results

(2) gem5 runs
Approach – Communication between gem5 & Ptolemy II

- **gem5 Simulator**
  - CPU
  - L1 I Cache
  - L1 D Cache
  - DRAM

- **Ptolemy II Model**
  - DiscreteClock
  - DE Director
  - Gem5Wrapper
  - DRAMModel
  - PowerTemperatureModel

- **Communication**
  - **“Fire”** (Run simulation for N cycles)
  - **Named pipe 1**
  - **Named pipe 2**
  - **“Notify”** (Simulation finished & results ready)

- **Shared File**
  - Memory trace:
    - <time, access type, addr>
    - <time, access type, addr>
    - ...

- **Actions**
  - (3) gem5 finishes and stops
  - Store simulation results
  - Load simulation results
**Approach**

– Communication between gem5 & Ptolemy II

---

**Diagram:**

**gem5 Simulator**
- CPU
- L1 I Cache
- L1 D Cache
- DRAM

**Ptolemy II Model**
- DiscreteClock
- DE Director
- DRAMModel
- PowerTemperatureModel

**Actors & Pipes:**
- **“Fire”** (Run simulation for N cycles)
  - Named pipe 1
  - Named pipe 2
- **“Notify”** (Simulation finished & results ready)

**Shared File**
Memory trace:
- \(<\text{time}, \text{access type}, \text{addr}\>
- \(<\text{time}, \text{access type}, \text{addr}\>
- \(<\text{time}, \text{access type}, \text{addr}\>
- ...  

**Steps:**
1. Store simulation results
2. Load simulation results
3. Gem5Wrapper fire() returns
Approach
– Communication between gem5 & Ptolemy II

(5) Gem5Wrapper postfire() triggers gem5 again

Store simulation results

Load simulation results

Shared File
Memory trace:
<time, access type, addr>
<time, access type, addr>
<time, access type, addr>

“Fire”
(Run simulation for N cycles)

“Notify”
(Simulation finished & results ready)
Approach – A DRAM behavioral model in Ptolemy II

TerraSwarm Research Center
Approach
– A DRAM Behavioral Model in Ptolemy II

An array of records

{{bank = 5, cmd = "READ", channel = 0, service_time = 107988},
{bank = 5, cmd = "READ", channel = 0, service_time = 108192},
{bank = 5, cmd = "READ", channel = 0, service_time = 108418},
{bank = 5, cmd = "READ", channel = 1, service_time = 109030},
{bank = 6, cmd = "WRITE", channel = 0, service_time = 109078}}

Process commands using service_time field

Calculate throughput over a moving time window
Approach
– DRAM Power & Thermal Model in Ptolemy II

TerraSwarm Research Center
Approach
– DRAM Power & Thermal Model in Ptolemy II

• CMOS Device power = Static power + Dynamic power

\[ P_{device} = P_{DRAM\_static} + P_{DRAM\_dynamic} \]

• DRAM dynamic power \( \propto \) Throughput

\[ P_{DRAM} = P_{DRAM\_static} + \alpha_1 \times \text{Throughput}_{read} + \alpha_2 \times \text{Throughput}_{write} \]

\[ P_{AMB} = P_{AMB\_idle} + \beta \times \text{Throughput}_{Bypass} + \gamma \times \text{Throughput}_{Local} \]

Equations & coefficients are from Lin et al., ISCA’07
Approach
– DRAM Power & Thermal Model in Ptolemy II

• DRAM stable temperature from DRAM power

\[ T_{AMB} = T_A + P_{AMB} \times \Psi_{AMB} + P_{DRAM} \times \Psi_{DRAM_{AMB}} \]

\[ T_{DRAM} = T_A + P_{AMB} \times \Psi_{AMB_{DRAM}} + P_{DRAM} \times \Psi_{DRAM} \]

• Current DRAM temperature

\[ T(t + \Delta t) - T(t) = (T_{stable} - T(t))(1 - e^{-\frac{\Delta t}{\tau}}) \]

Equations & coefficients are from Lin et al., ISCA‘07
Approach
– DRAM Power & Thermal Model in Ptolemy II

MoC

Inputs

Stable temperature

Current temperature

(a)
• AMB/DRAM power tendency example

Converges to stable temperature ($P_{\text{static}} + P_{\text{dynamic}}$)

DRAM becomes idle

Converges to stable temperature ($P_{\text{static}}$)

DRAM access starts from $T_{\text{ambient}}$
Experiments and Results – Experimental Setup

• Experimented on
  – Different cache configurations
  – Different software workloads

• To measure
  – Average DRAM/AMB power
  – Peak DRAM/AMB temperature reached during simulation (0.1 sec in simulated time)
Experiments and Results  
– Experimental Setup

- gem5 configurations (except caches)
  - ISA – ARM
  - CPU Type – TimingSimpleCPU: Stalls on every load memory access.
  - Clock rate – CPU: 1GHz / System: 1GHz
  - Off-chip DRAM memory: DDR3 SDRAM with a data rate of 1600MHz and a bus width of 16 bits.
  - Cache block size – 64 bytes
Experiments and Results
– Power and Temperature Results

• Benchmark
  – Top 5 memory-intensive programs from MiBench
    • where memory-intensity is defined as
      \[
      \# \text{memory accesses (read+write)} / \text{instruction}
      \]

<table>
<thead>
<tr>
<th>MiBench programs</th>
<th>writes</th>
<th>reads</th>
<th>total instructions executed</th>
<th>memory intensity (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>cjpeg_large</td>
<td>6,183</td>
<td>74,966</td>
<td>1,000,000</td>
<td>8.11</td>
</tr>
<tr>
<td>rijndael_large</td>
<td>2,558</td>
<td>68,458</td>
<td>1,000,000</td>
<td>7.1</td>
</tr>
<tr>
<td>typeset_small</td>
<td>12,843</td>
<td>55,963</td>
<td>1,000,000</td>
<td>6.88</td>
</tr>
<tr>
<td>dijkstra_large</td>
<td>4,942</td>
<td>59,198</td>
<td>1,000,000</td>
<td>6.41</td>
</tr>
<tr>
<td>patricia_large</td>
<td>4,255</td>
<td>49,198</td>
<td>1,000,000</td>
<td>5.35</td>
</tr>
</tbody>
</table>
Experiments and Results
– Power and Temperature Results

• Power and temperature results for different cache configurations
  – (workload: cjpeg_large)

<table>
<thead>
<tr>
<th>Cache size options (KB)</th>
<th>Average power (mW)</th>
<th>Maximum temperature increase (10^6 °C)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DRAM</td>
<td>AMB</td>
</tr>
<tr>
<td>L1 (I/D)</td>
<td>L2</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>N/A</td>
<td>1,057</td>
</tr>
<tr>
<td>32</td>
<td>N/A</td>
<td>1,023</td>
</tr>
<tr>
<td>64</td>
<td>N/A</td>
<td>1,000</td>
</tr>
<tr>
<td>32</td>
<td>128</td>
<td>996</td>
</tr>
<tr>
<td>32</td>
<td>256</td>
<td>995</td>
</tr>
</tbody>
</table>
Experiments and Results
– Power and Temperature Results

• Temperature results for different workloads
  – (Cache configuration: L1: 16kB, L2: N/A)
Tool Demonstration

• Gem5 tuned for tool integration
  – https://github.com/gem5-ptolemy/

• Ptolemy II
  – http://ptolemy.org
    • Version 11.0 – Development version
  – Case study example model:
    • ptolemy/actor/lib/gem5/demo/DramThermalModel.xml
Conclusions

• Summary
  – The gem5 architecture simulator is integrated into Ptolemy II as an computer architectural aspects with higher accuracy
  – Experiments show usefulness of the approach

• Future work
  – More architectural information from gem5
  – More applications for the proposed approach

• For more information
  – https://github.com/gem5-ptolemy/
  – http://ptolemy.org