Simics and Friends – Modeling Tools for CMP Research

Zvika Guz, Isask’har (Zigi) Walter

The Technion – Israel Institute of Technology
Agenda

Official Agenda
- Review the most commonly used tools in CMP μarch research
  - Simulators
  - Benchmarks

Unofficial Agenda
- Convince you to use Simics
  - Because most often than not it is the best option
  - Because we need more (geographically adjacent) people

Not on Our Agenda
- Teaching the tools
Outline

- Choosing a Simulators
- Simics
- And friends
  - GEMS, Garnet & Orion, FeS$_2$, SimFlex
- OPNET - modeling CMP interconnect
- Benchmarks
- Summary
  - Technion goodies
Outline

- **Choosing a Simulators**
- Simics
- And friends
  - GEMS, Garnet & Orion, FeS$_2$, SimFlex
- OPNET - modeling CMP interconnect
- Benchmarks
- Summary
  - Technion goodies
Choosing A Simulator

- What should it model?
  - Processor /Cache/Interconnect/etc.

- What would run on it?
  - Benchmarks type
CMP Research (very partial list)

- System-wide architecture
  - Asymmetric CMP

- Memory hierarchy
  - Caches
  - Coherence protocols

- Reciprocalation between HW and SW
  - Hybrid Transactional memory

- Interactions among the cores
  - Task assignment/migration

- Interconnect
  - Network-On-Chip
Choosing a Simulator for CMP Research

- What will it model?
  - Multiple cores
  - Memory hierarchy (caches, coherence)
  - Interconnect (NoC)
- What will run on it?
  - Multi-threaded benchmarks → Need OS for that
  - Commercial workloads → Really need OS for that

⇒ Full-system simulator, capable of booting (commercial) OS
Outline

- Choosing a Simulators
- **Simics**
- And friends
  - GEMS, Garnet & Orion, FeS$_2$, SimFlex
- OPNET - modeling CMP interconnect
- Benchmarks
- Summary
  - Technion goodies
Meet the Contenders

- SimpleScalar
  - Uniprocessor
- PIN
  - Not a simulator
- Several in-house tools
  - Not relevant
- M5
- Simics
- ?
Why Simics? (the short answer)

- Because *everyone* is using it
  - THE most widely used simulator in our field
  - 1/3 of ISCA’07 papers used Simics
- Huge, active community
- Alive and kicking forum
- Because it is free
  - For academia
    - Up to Simics 4.2 😊
- ..Oh.. and because it is really really really good!
Simics in a Nutshell

- Virtual Hardware
  - Event driven
  - Cycle accurate*

The software can’t tell the difference
Runs binaries from real target

Complete production software

Simulated (virtual) hardware

HW/SW interface

http://www.virtutech.com/
Simics Overview (1/3)

“A software, event-driven simulator

Full-system simulator

- Processor
- Memory hierarchy (DRAM, Disk)
- Network
- Devices (DMA, Interrupt controller, PCI, etc.)

Runs unmodified binaries

- OS, drivers and applications
- Models the entire machine that OS sees
- Application cannot tell the difference

http://www.virtutech.com/
Simics Overview (2/3)

“Simics is a flexible, scalable, and high-performance full-system simulator”

- Fully supported ISAs:
  - SPARC
  - X86
  - Alpha, Itanium, MIPS, ARM, ..

- Scalable:
  - Single processor (uniprocessor /CMP) → MPs → Racks → Clusters → Distributed systems

http://www.virtutech.com/
Simics Overview (3/3)

“Simics is a flexible, scalable, and high-performance full-system simulator”

- Flexible
  - Different degrees of simulation (details)
    - Functionality only
    - Microarchitecture and timing
  - Configurable
    - Hook/unhook modules
    - Control their timing
    - Write your own (in C++)

http://www.virtutech.com/
"Demo"

Red Hat 7.2/Itanium

Solaris/PowerPC

Red Hat 6.2/x86

ITanium

Simics console

Solaris 8/UltraSparc II

Red Hat 7.2/ Pentium III

Simics console

NT/x86

Red Hat 6.2/ Pentium III

XP/x86-64

Sun Solaris 8/UltraSparc II

http://www.virtutech.com/
What Have We Seen?

- User application code
- Middleware and libraries
- Target operating system(s)
- Virtual target hardware
- Simics
- Host operating system
- Host hardware

http://www.virtutech.com/
Simics Provides:

- Checkpoints
  - Save/restore state

- Breakpoints
  - Temporal breakpoints
  - Break on memory/Register/IO
  - Graphics breakpoint

- Magic instructions
  - Signal Simics from within your application

- Access host files from the simulated machine

- So much more..
**Simics Timing Models**

- **Default mode**
  - Every instruction takes exactly 1 clock cycle
    - Including access to disc, access to memory, etc.

- **In-order mode**
  - User defines a timing model function which will be called when memory request occurs
  - Function returns the number of cycles to stall

- **Out-of-order mode (MAI mode)**
  - Detailed out-of-order µarch simulation
  - User-defined processor model
    - Full control on how instructions advance
Simics Timing - *default*

- Emulation mode
- Used for fast-forwarding
  - □ Boot OS
  - □ Build workload
  - □ Fast-forward to relevant execution part
- Basically, used for creating a checkpoint
Simics Timing – *in order*

- Timing model is a C program
- You can act on every memory access
- Usually used for modeling:
  - Caches (and cache hierarchies)
  - Coherency protocols (directory)
  - Hardware/Hybrid transactional memory
Simics Timing – *Out Of Order Mode*

- Gives full control over timing
  - User decides when things happen
    - Fetch/decode/execute/commit
  - Simics handle how these things happen
- MAI supports:
  - Out-of-order execution, multi-processor, multi-threading, branch prediction, value prediction
- Used for processor µarch research
  - Models processor internal
- And whenever you need a better notation of time
  - Interconnect study
Simple Example – Adding Cache (1/4)

- Nahalal – A new cache architecture for CMP
- Architectural differentiation of cache lines at runtime
  - According to usage - *Private vs. Shared*
Simple Example – Adding Cache (2/4)

1. Writing a cache timing model

   C- Program

```c
/* Handles read transactions coming to the cache. */
static cycles_t
nuca_handle_read(generic_cache_t *gc, generic_transaction_t *mem_op,
                 conf_object_t *space, map_list_t *map)
{

    SIM_log_info(4, &gc->log, GC_Log_Read_Hit, "&s: Enter real_nuca_handle_read", gc->log.obj.name);

    /* Look for a matching line. */
    line_num = lookup_line(gc, mem_op, &way);

    if (line_num != -1) { /* Yes, we have a hit. */

        SIM_log_info(2, &gc->log, GC_Log_Read_Hit,"Hit");
        penalty++;
        CalcAccessTime(gc, cpu, mem_op, DistanceInNumOfHops(gc,cpu,way), READ);

        if(gc->config.NAKALAL)
            penalty += PromoteBlock(gc, mem_op, cpu, line_num, way, &new_line_num, NULL);
    
    } else { /* No, we have a miss. */
        SIM_log_info(3, &gc->log, GC_Log_Read_Miss,"Miss");

        line_num = gc->config.repl_fun.get_line(gc->config.repl_data, gc, mem_op);
        penalty += empty_line(gc, line_num, mem_op, space, map);
        penalty += fetch_line(gc, line_num, mem_op, space, map);
    }

    return penalty;
}
```
Simple Example – Adding Cache (3/4)

2. Hooking the new cache into Simics
   - Python script

```python
conf += [ 
    #
    # 12 cache: 16Mb Write-back
    #
    OBJECT("l2c", "u-cache")
    cpu0 = [OBJ("cpu0"), OBJ("cpu1"), OBJ("cpu2"), OBJ("cpu3"),
            OBJ("cpu4"), OBJ("cpu5"), OBJ("cpu6"), OBJ("cpu7")],
    config_NAHALAL = 1,
    config_line_number = 262144,
    config_line_size = 64,
    config_assoc = 16,
    config_virtual_index = 0,
    config_virtual_tag = 0,
    config_write_back = 1,
    config_write_allocate = 1,
    configReplacement_policy = "lru",
    penalty_read = 15,
    penalty_write = 15,
    penalty_link = 5,  # each link requires 5 cycles to traverse
    configParallel_search = 1,
    higher_level_caches = cache_olist,
    timing_model = OBJ("staller")),
```
Simple Example – Adding Cache (4/4)

3. Run Simics and collect statistics

surya-0 1298% simics -stall -c config8new
Checking out a license... done: academic license.
Looking for additional Simics modules in /home/grads/yding/simics-2.2.12/v9-so18-64/lib

+-----------------------+ Copyright 1998-2005 by Virtutech, All Rights Reserved
| Virtutech  | Version: simics-2.2.12
| Simics     | Compiled: Fri Apr 22 07:08:11 MEST 2005
+-----------------------+
www.simics.com  "Virtutech" and "Simics" are trademarks of Virtutech AB

Type 'copyright' for details on copyright.
Type 'license' for details on warranty, copying, etc.
Type 'readme' for further information about this version.
Type 'help help' for info on the on-line documentation.

simics>

simics>
Simics in Research

- “Memory Mapped ECC: Low-Cost Error Protection for Last Level Caches”, D. H. Yoon and M. Erez, ISCA 2009
- “Predicting the Performance of Reconfigurable Optical Interconnects in Distributed Shared-Memory Systems”, W. Heirman, J. Dambre, I. Artundo, C. Debaes, H. Thienpont, D. Stroobandt, J. Van Campenhout, Photonic Network Communications ’08
Outline

- Choosing a Simulators
- Simics
- And friends
  - GEMS, Garnet & Orion, FeS2, SimFlex
- OPNET - modeling CMP interconnect
- Benchmarks
- Summary
  - Technion goodies
Add-ons for Simics

- Open-source add-ons enlarge Simics capabilities
- Some as popular as Simics itself

- GEMS
- Garnet & Orion
- SimFlex
- FeS$_2$
Multifacet GEMS

“GEMS is a set of modules for Virtutech Simics that enables detailed simulation of multiprocessor systems, including CMP.”

- The most mature Simics add-on
  - Most of ISCA’s Simics papers actually use GEMS
  - Alive and active forum

- Two main components
  - Ruby – Memory system timing simulator
  - Opal – Timing model for OOO processor

- Flexible
  - Can be configured/ altered/ hacked
  - Add your own models

http://www.cs.wisc.edu/gems/
GEMS Ruby

- Cache hierarchy
  - L1, L2 (private/shared), SNUCA/DNUCA, Simple DRAM
- Different coherence protocols
  - Snoop, Directory, Token coherence
  - Write your own
- HW transaction memory
  - Log-TM
  - Sun’s Rock
- Interconnect
  - Simple
  - Garnet - detailed NoC interconnect

http://www.cs.wisc.edu/gems/
Outline

- Choosing a Simulators
- Simics
- And friends
  - GEMS, Garnet & Orion, FeS2, SimFlex
- **OPNET - modeling CMP interconnect**
- Benchmarks
- Summary
  - Technion goodies
We need to model the interconnect too

- Might have a paramount effect on performance and power

Sometime, this is all we need!
Simulate the Interconnect? Why Bother?

- Important part of the **system**!
- Static modeling can account for static attributes
  - Topology, routing, link bandwidth, packet size, etc.
- Run-time effects are much harder to (statically) model
  - Shared resource arbitration, finite buffer sizes, channel multiplexing, flow control, …
  - Might be dominating factors
    - Driving home during rush hours
Network vs. Full System Simulator

- NoC is a network!
  - Use a network oriented tool with built in support for traffic modeling
  - Eliminate complex system simulator if not really needed
- Perfect tool for optimizing the interconnect
  - Architecture, topology, protocols, parameter tuning, etc.
- Easy programming and debugging
- Fast!
  - “Fastest discrete event simulation engine among leading industry solutions”
OPNET Modeler Features

- Object-oriented modeling
- Hierarchical modeling environment
- GUI-based debugging and analysis
- Event-driven simulation engine
- Coding C/C++ & auxiliary functions
- Open interface for integrating external object files, libraries, and other simulators
- Asynchronous/synchronous modeling
OPNET in CMP Research


- "Routing Table Minimization for Irregular Mesh NoCs“, E. Bolotin, I. Cidon, R. Ginosar, A. Kolodny, DATE 2007


- "Best of Both Worlds: A Bus Enhanced NoC (BENoC)“, R. Manevich, I. Walter, I. Cidon, and A. Kolodny, the ACM/IEEE Int. Symp. on Networks-on-Chip (NOCS), 2009
Bus-Enhanced Network on-Chip

- A new interconnect architecture, utilizing “the best of both worlds”
  - Use NoC for data delivery
  - Use bus for lightweight, latency critical meta-data
    - Coherency
- Evaluated used OPNET and Simics
Bus-Enhanced Network on-Chip
Gluing OPNET to Simics

- Run OPNET as a trace-driven simulator
  - L2 access logs generated by Simics

- Advantages
  - Fast
  - Simple

- Disadvantage
  - Dependencies are lost
  - Does not account for latency hiding techniques (e.g. OOO)

- But..
  - OPNET can be glued to Simics using Ruby
Outline

- Choosing a Simulators
- Simics
- And friends
  - GEMS, Garnet & Orion, FeS2, SimFlex
- OPNET - modeling CMP interconnect
- **Benchmarks**
- Summary
  - Technion goodies
Meet the Contenders

- CPU2006, CPU2000
- OMP2001
- JBB2005, JBB2000
- SPLASH-2
- PARSEC

- Commercial workloads
  - Apache
  - Databases
  - ?

?
# Benchmark Comparison

<table>
<thead>
<tr>
<th></th>
<th>CPU 2006</th>
<th>OMP 2001</th>
<th>SPLASH-2</th>
<th>PARSEC</th>
<th>Commercial</th>
</tr>
</thead>
<tbody>
<tr>
<td>Programs</td>
<td>29</td>
<td>11</td>
<td>14</td>
<td>13</td>
<td>1</td>
</tr>
<tr>
<td>Multi-Threaded</td>
<td>❌❌❌</td>
<td>☺</td>
<td>☺</td>
<td>☺</td>
<td>☺</td>
</tr>
<tr>
<td>Diverse</td>
<td>❌</td>
<td>❌</td>
<td>☺</td>
<td></td>
<td>☺</td>
</tr>
<tr>
<td>Updated</td>
<td>❌</td>
<td>❌</td>
<td></td>
<td></td>
<td>☺</td>
</tr>
<tr>
<td>Emerging apps</td>
<td></td>
<td>❌</td>
<td></td>
<td></td>
<td>☺</td>
</tr>
<tr>
<td>Installation ease</td>
<td>☺</td>
<td>☺</td>
<td>☺</td>
<td></td>
<td>❌ ❌ ❌</td>
</tr>
<tr>
<td>Simulation friendly</td>
<td>❌ ❌ ❌</td>
<td>❌</td>
<td>☻ ☻ ☻</td>
<td></td>
<td>❌ ❌ ❌</td>
</tr>
</tbody>
</table>
The PARSC Benchmark Suite

<table>
<thead>
<tr>
<th>Program</th>
<th>Application Domain</th>
<th>Parallelization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blacksholes</td>
<td>Financial Analysis</td>
<td>Data-parallel</td>
</tr>
<tr>
<td>Bodytrack</td>
<td>Computer Vision</td>
<td>Pipeline</td>
</tr>
<tr>
<td>Canneal</td>
<td>Engineering</td>
<td>Data-parallel</td>
</tr>
<tr>
<td>Dedup</td>
<td>Enterprise Storage</td>
<td>Pipeline</td>
</tr>
<tr>
<td>Facesim</td>
<td>Animation</td>
<td>Data-parallel</td>
</tr>
<tr>
<td>Ferret</td>
<td>Similarity Search</td>
<td>Pipeline</td>
</tr>
<tr>
<td>Fluidanimate</td>
<td>Animation</td>
<td>Data-parallel</td>
</tr>
<tr>
<td>Freqmine</td>
<td>Data Mining</td>
<td>Data-parallel</td>
</tr>
<tr>
<td>Raytrace</td>
<td>Visualization</td>
<td>Data-parallel</td>
</tr>
<tr>
<td>Streamcluster</td>
<td>Data Mining</td>
<td>Data-parallel</td>
</tr>
<tr>
<td>Swaptions</td>
<td>Financial Analysis</td>
<td>Data-parallel</td>
</tr>
<tr>
<td>Vips</td>
<td>Media Processing</td>
<td>Data-parallel</td>
</tr>
<tr>
<td>X264</td>
<td>Media Processing</td>
<td>Pipeline</td>
</tr>
</tbody>
</table>

- Over 1000 downloads since release
- This is what everyone will be using
Outline

- Choosing a Simulators
- Simics
- And friends
  - GEMS, Garnet & Orion, FeS2, SimFlex
- OPNET - modeling CMP interconnect
- Benchmarks
- Summary
  - Technion goodies
Technion Goodies

- [http://www.ee.technion.ac.il/matrics/software.html](http://www.ee.technion.ac.il/matrics/software.html)
- Simics workload kits
  - Ease up installation of simics workloads
    - Wisconsin GEMS provide few other too
  - Constantly adding more workloads to the pool
    - Can you help?
- OPNET models for NoC
  - Our entire QNoC model for OPNET
    - Cores, router and links, SNUCA/DNUCA L2 caches
    - Routing schemes, arbitration policies, resource contention
    - Synthetic/trace driven simulation
- Transactified version of Apache
Summary

- A swift overview of simulation tools for CMP
  - Simics
  - GEMS
  - OPNET
  - Benchmarks
- Technion’s two cents

Questions?

zguz@tx.technion.ac.il