Simics: A Full System Simulation Platform

Peter S. Magnusson et al
Virtutech

Presented by Clif Kerr
Overview

- Accurately simulate function of a given architecture
- Facilitate timing simulation with hooks to third-party timing simulators
- Operate fast enough to run real workloads
About Simics

- Functional simulation of UltraSparc, Alpha, x86 (32 bit and 64 bit extended), PowerPC, Itanium, MIPS, and ARM processors
  - Simulations run unmodified operating systems and code
Simics Applications

- Processor Design
- Multiprocessor Architecture
- Operating System Development and Emulation
- Debugging
Implimentation

- Simics Central – coordinates communication between Simics modules in a simulated network
- Memory simulation through simulated transaction cache
- Interpreter generated from high-level architecture specification
Extensibility

- New device modules “plug in” to Simics framework
- Simics API provides numerous functions, data types, and interfaces defined for user-defined extensions
Simics Architecture

Target machine
- Applications
- Operating system
- Firmware

Simics
- Configuration
- Event handling
- Memory
- Interpreter

Programming interface (Simics API)
- Memory management unit
- Command-line interface
- Scripting
- Tracing
- Debugger
- Cache models

Memory bus
- Devices
- Graphics
- Ethernet
- SCSI

Simics Central
- Local disk

VHDL simulator

Other Simics process

Real network
Performance

- Simics can simulate real workloads at reasonable speeds – on the order of a million instructions per second

<table>
<thead>
<tr>
<th>Target</th>
<th>Boot workload</th>
<th>Instructions</th>
<th>Time (sec)</th>
<th>MIPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alpha-ev5</td>
<td>Tru64</td>
<td>2,112,119,247</td>
<td>354</td>
<td>5.9</td>
</tr>
<tr>
<td>Alpha-ev5</td>
<td>Linux</td>
<td>1,201,600,120</td>
<td>164</td>
<td>7.3</td>
</tr>
<tr>
<td>Sparc-u2</td>
<td>Solaris 8(^1)</td>
<td>1,597,537,438</td>
<td>284</td>
<td>5.6</td>
</tr>
<tr>
<td>Sparc-u3</td>
<td>Solaris 8(^1)</td>
<td>6,155,835,717</td>
<td>987</td>
<td>6.2</td>
</tr>
<tr>
<td>x86-p2</td>
<td>Linux(^2)</td>
<td>1,299,639,608</td>
<td>227</td>
<td>5.7</td>
</tr>
<tr>
<td>x86-p2</td>
<td>Windows XP</td>
<td>3,129,351,000</td>
<td>1,518</td>
<td>2.1</td>
</tr>
<tr>
<td>x86-64</td>
<td>Linux(^2)</td>
<td>1,299,639,608</td>
<td>285</td>
<td>4.5</td>
</tr>
<tr>
<td>Itanium</td>
<td>Linux</td>
<td>4,644,372,142</td>
<td>1,470</td>
<td>3.2</td>
</tr>
<tr>
<td>PPC-750</td>
<td>VxWorks</td>
<td>1,179,516,468</td>
<td>136</td>
<td>8.7</td>
</tr>
<tr>
<td>PPC-750</td>
<td>Linux(^2)</td>
<td>498,836,969</td>
<td>53</td>
<td>9.3</td>
</tr>
</tbody>
</table>
Questions

- Is the reported performance a result of optimization or simulator simplicity?
- What are some of the potential drawbacks of making a completely general simulation environment?