[18:07] <sseehh_> there are CPU simulators like flexus and gem5 that do instruction / timing-level simulation of an entire OS + applications
[18:08] <sseehh_> so you provide the image including kernel and packages and user-level applications and run it
[18:08] <sseehh_> and it generates lots of statistics
[18:08] <sseehh_> oh you also provide the architecture parameters
[18:08] <sseehh_> like ISA (x86, ARM, etc...)
[18:08] <sseehh_> and cache parameters, like L1/L2 size and microcode policy
[18:08] <sseehh_> and # of cores
[18:08] <sseehh_> make sense so far?
[18:08] <sseehh_> its kind of like VMWARE but more for instrumentation
[18:08] <sseehh_> not for speed
[18:09] <sseehh_> it will tell for example, cache hit/miss ratios
[18:09] <sseehh_> on memory access
[18:09] <sseehh_> because it simulates the cache and memory too
[18:09] <sseehh_> so that affects chipset speed & also energy efficiency
[18:10] <sseehh_> the overall goal here is to make the process of finding the maximum speed/energy results for given architecture parameters (an n-dimensional space)
[18:10] <sseehh_> to make it automatic
[18:10] <sseehh_> to find things that humans otherwise might not find manually experimenting
[18:10] <sseehh_> because the design space is so huge

(06:49:51 PM) **S H:** basically the search algorithms generate 'hypotheses' which are possible simulations that can be run and add to the knowledgebase. the search uses the known experiments and result data to generate those hypotheses for future runs
(06:50:08 PM) **S H:** then it's all put into a task scheduler that prepares and deploys the simulation to a compute cluster, or just the local machine
(06:50:19 PM) **S H:** the results can be stored in a DB like mongodb
(06:50:22 PM) **Mariana Soffer:** of how to look for the goal
(06:50:26 PM) **S H:** which will be helpful for JSON flexibility
(06:50:48 PM) **S H:** then the results can be browsed individually, or multiple of them can be combined into a report/analysis
(06:51:02 PM) **S H:** and manually-specified experiments will become the "seeds" for new discovery
(06:51:19 PM) **S H:** so its just a mapping of all the simulation tool parameters and results
(06:51:22 PM) **S H:** to a high level interface
(06:51:25 PM) **S H:** pretty simple actually
(06:51:36 PM) **S H:** if you want to design the gradient ascent part, that would help a lot

(06:52:00 PM) **S H:** just assume you have a list of parameters w/ values for the experiment (inputs) and the list of parameters w/ values for the experiment results (outputs)
(06:52:03 PM) **S H:** and a list of those pairs
(06:52:14 PM) **S H:** [(simulation parameters, simulation results)]
(06:52:38 PM) **S H:** so the algorithm could take the N-best simulations and produce N*M hypotheses
(06:52:54 PM) **S H:** where M is the number of tweaked alternates that it generates
(06:53:13 PM) **S H:** it can use multiple hypotheses that share a common variable to formulate curves
(06:53:14 PM) **Mariana Soffer:** i can help
(06:53:16 PM) **Mariana Soffer:** with algorithms
(06:53:23 PM) **Mariana Soffer:** and strategies
(06:53:26 PM) **Mariana Soffer:** big picture
(06:53:28 PM) **S H:** and these can guide the search vectors (distance and direction)
(06:53:31 PM) **S H:** i know
(06:53:45 PM) **S H:** i have asked jeff to develop the genetic algorithm one
(06:53:54 PM) **S H:** so multiple algorithms can be applied in parallel
(06:54:03 PM) **S H:** they are just generating hypothetical experiments scheduled for a later time
(06:55:10 PM) **Mariana Soffer:** nice
(06:55:13 PM) **Mariana Soffer:** you can store all steps
(06:55:15 PM) **Mariana Soffer:** and tips
(06:55:21 PM) **Mariana Soffer:** and analyze them
(06:55:30 PM) **S H:** yes
(06:55:37 PM) **S H:** the simulators will produce lots of results
(06:55:50 PM) **S H:** cache hit/miss ratio, instruction pipeline density, etc

http://people.cs.pitt.edu/~childers/

CloudSuite [2.0]

http://parsa.epfl.ch/cloudsuite/isca13-tutorial.html
http://parsa.epfl.ch/cloudsuite/downloads.html

Flexus

http://parsa.epfl.ch/simflex/

- Built on Simics targeting the SPARC ISA
  - Timing-accurate processor, memory, and interconnect simulation
  - Component-based design leveraging C++ features and libraries
  - Easy composition of in-order and out-of-order uniprocessor and multiprocessor models
- Built-in statistics management and simulation state checkpointing
  - Leverages SMARTS and live-points acceleration techniques
- Current models include a uniprocessor, and a CMP and DSM derived from Piranha

  6/26/2012 - CloudSuite on Flexus tutorial presented on June 9th, 2012 at ISCA. Here are the tutorial slides and a guideline for using the released Simics images.

ProtoFlex FPGA-accelerated simulator
http://www.ece.cmu.edu/~protoflex/doku.php

Simics

WIND RIVER SIMICS DEMO: GETTING STARTED WITH SIMICS
http://www.youtube.com/watch?v=1MJgxFi2I-U&

The Wisconsin Multifacet Project is pleased to present the open-source release of our General Execution-driven Multiprocessor Simulator (GEMS). GEMS is a set of modules for Virtutech Simics that enables detailed simulation of multiprocessor systems, including Chip-Multiprocessors (CMPs).
http://research.cs.wisc.edu/gems/

gem5

The gem5 simulator is a modular platform for computer system architecture research, encompassing system-level architecture as well as processor microarchitecture.

> The gem5 simulation infrastructure is the merger of the best aspects of the M5 [4] and GEMS [9] simulators. M5 provides a highly configurable simulation framework, multiple ISAs, and diverse CPU models. GEMS complements these features with a detailed and exible memory system, including support for multiple cache coherence protocols and interconnect models. Currently, gem5 supports most commercial ISAs (ARM, ALPHA, MIPS, Power, SPARC, and x86), including booting Linux on three of them (ARM, ALPHA, and x86).
>
> The project is the result of the combined efforts of many academic and industrial institutions, including AMD, ARM, HP, MIPS, Princeton, MIT, and the Universities of Michigan, Texas, and Wisconsin. Over the past ten years, M5 and GEMS have been used in hundreds of publications and have been downloaded tens of thousands of times. The high level of collaboration on the gem5 project, combined with the previous success of the component parts and a liberal BSD-like license, make gem5 a valuable full-system simulation tool.

http://gem5.org/Status_Matrix
http://gem5.org/Architecture_Support

Benchmarks

- For information about running Android on gem5 and using the web browser benchmark, see BBench-gem5.
- For information about using the DaCapo benchmarks on gem5 see the DaCapo benchmarks page for more information.
- SPLASH benchmarks -- See the Splash benchmarks page for more information.
- ***CloudSuite 2.0 ???***

http://gem5.org/Statistics
http://gem5.org/Ruby (not the programming language)

> implements a detailed simulation model for the memory subsystem. It models inclusive/exclusive cache hierarchies with various replacement policies, coherence protocol implementations, interconnection networks, DMA and memory controllers, various sequencers that initiate memory requests and handle responses. The models are modular, flexible and highly configurable. Three key aspects of these models are:

http://gem5.org/Main_Page
http://gem5.org/Tutorials

This tutorial was held in Gothenburg, Sweden in April 2012. It covers gem5 although for information about Ruby you should look at the ISCA 38 tutorial. We recorded video of the tutorial which is available below.

- Slides
- Overview
- Introduction
- Basics
- Running Experiments
- Debugging
- Memory
- CPU Models
- Common Tasks
- Configuration
- Conclusion

https://www-auth.cs.wisc.edu/lists/gems-users/2011-June/msg00017.shtml

> *There are a lot of differences (and similarities). Simics does not provide cycle-accurate simulation but does provide full system functionality. GEMS was designed to explore multiprocessor (multicore) architecture features and add cycle accuracy and a detailed memory hierarchy to Simics. M5 supports cycle accurate full system simulation and system call emulation (a la SimpleScalar) for multicore architectures on a couple of architecture families. With the inclusion of Ruby in M5, the GEM5 project has both the detailed memory model of GEMS and the rich processor support of M5.*
>
> *gem5 is FOSS and you can download and run it immediately (after some setup). Simics is available for academic licensing through Wind River Systems, whom you would contact to request a site license. They grant one per institution typically. GEMS is installed as an extension to Simics and runs as a plugin module.*

Since all of the source code is readily available, gem5 is easier to code and debug. I have written extensions to both simulators and found the gem5 approach to be more mature and usable.

gem5 supports more architecture families, although its ALPHA support is most mature, with x86 and ARM catching up. gems works only on the SPARC v9 instruction set.

With gem5, Linux is the OS of choice. With gems, it is difficult to use any OS other than Solaris, although some groups have had success.

gem5 is still maintained, whereas most of the gems authors have moved on.

My opinion is that if you are just starting out, you will do better with gem5. The reason to use simics/gems would be if you already are familiar with it and have it set up to work in your lab.

A Computer Scientist's Quest: Architectural simulators gedare-csphd.blogspot.com/2010/12/architectural-simulators.html

comments.gmane.org/gmane.comp.emulators.m5.users/9193

Well, as far as I understand GEM5 removes simics and replaces it with M5, so I think the answer to your question is no, GEM5 does not depend on simics, but keep in mind that GEMS and GEM5 are different things. "GEMS is no longer under active development. The effort has shifted to the gem5 simulator system, a fully open-source software."

Discussion for users of the **gem5** simulator () - Gmane

Feb 20, 2012 - Well, as far as I understand GEM5 removes simics and replaces it with M5, so I think the answer to your question is

no, GEM5 does not depend ...

Discussion for users of the **gem5** simulator () - Gmane

comments.gmane.org/gmane.comp.emulators.m5.users/11653

Aug 22, 2012 - You should check with the GEMS and/or Simics people. This list is for gem5 only, which is something completely different. -Tony. On Wed, Aug ...

[PPT]

Slides

ppi.fudan.edu.cn/art/_media/publications%3Btransformer.dac.pptx

Gem5: GEMS + M5; MARSS: PTLsim + QEMU ... E.g., Simics + GEMS, Xen/QEMU + PTLsim; Each cycle, FM/TM interact with each other; To direct the execution ...

[PDF]

Simulating Systems not Benchmarks - **gem5**

gem5.org/dist/tutorials/hipeac2012/**gem5**_hipeac.pdf

Borrowing material from previous gem5 tutorials .... Using saved variables file /work/gem5/build/variables/ARM ..... e.g. SimpleScalar, Simics, Asim etc.

simulator - PARsE | Research Log

parse.ele.tue.nl/weblog?tag=simulator

Jul 14, 2011 - The GEM5 merges the M5sim simulator and the GEMS simulator. It seems the GEM5 no longer depends on the Virtutech Simics.

Multi2Sim

Multi2Sim has recently released a GPU simulator for AMD GPUs, capable of simulating OpenCL programs. It will hold a tutorial in PACT 2011.

http://gem5.org/Splash_benchmarks
http://gem5.org/SPEC2006_benchmarks