

Vincent M. Weaver

364 Upson Hall
School of Electrical and Computer Engineering
Cornell University, Ithaca, NY 14853

vince@cs.cornell.edu
<http://www.csl.cornell.edu/~vince>
office 607.255.3678

Education

Ph.D.	Electrical and Computer Engineering, Cornell University	Jan 2010
M.S.	Electrical and Computer Engineering, Cornell University	Jan 2009
B.S.	Electrical Engineering, Univ. of Maryland College Park	Dec 2000

Work Experience

Center for Applied Math, Cornell University	<i>Ithaca, NY</i>
2008-2009	TA: Sysadmin for CAM department Maintain network and 30+ RHEL4 machines
ECE, Cornell University	<i>Ithaca, NY</i>
2006-2008	TA, Head TA: ECE314 (Computer Organization)
CASC, Lawrence Livermore National Lab	<i>Livermore, CA</i>
2005	Summer Intern: Cache Conscious Data Placement
ECE Techshop, Cornell University	<i>Ithaca, NY</i>
2003-2004	TA: Sysadmin of Linux, BSD, and Solaris network
US Army SBCCOM	<i>Aberdeen Proving Ground, MD</i>
2002-2003	ORISE: Web-interfaces for legacy Fortran models
Frontpath Inc.	<i>Billerica, MA</i>
2000-2001	Jr. Software Engineer: Transmeta-based Tablet PC Linux, Gtk+, and x86 BIOS development
US Army SBCCOM	<i>Aberdeen Proving Ground, MD</i>
1995-2000	GWU, ORISE Summer/Winter intern: Web-interfaces for legacy Fortran models

Publications

Refereed Conferences

1. V.M. Weaver, S.A. McKee. “**Code Density Concerns for New Architectures**”, 27th IEEE International Conference on Computer Design (ICCD 2009), Lake Tahoe, California, October 2009 (to appear).
2. M. Bhadauria, V.M. Weaver, S.A. McKee. “**Understanding PARSEC Performance on Contemporary CMPs**”, IEEE International Symposium on Workload Characterization (IISWC 2009), Austin, Texas, October 2009 (to appear).
3. M. Bhadauria, V.M. Weaver, S.A. McKee. “**Accommodating Diversity in CMPs with Heterogeneous Frequencies**”, 4th EC International Conference on High Performance Embedded Architectures and Compilers (HiPEAC’09), Cyprus, January 2009, pp. 248-262.
4. V.M. Weaver, S.A. McKee. “**Can Hardware Performance Counters be Trusted?**”, IEEE International Symposium on Workload Characterization (IISWC 2008), Seattle, Washington, September 2008, pp. 141-150.
5. V.M. Weaver, S.A. McKee. “**Using Dynamic Binary Instrumentation to Generate Multi-Platform Simpoints: Methodology and Accuracy**”, 3rd EC International Conference on High Performance Embedded Architectures and Compilers (HiPEAC’08), Göteborg, Sweden, January 2008, pp. 305-319.

Refereed Workshops

1. V.M. Weaver, S.A. McKee. “**Are Cycle Accurate Simulations a Waste of Time?**”, WDDD: Workshop on Duplicating, Deconstructing and Debunking, Beijing, China, June 2008, pp. 40-53.

Posters

1. M. Bhadauria, V.M. Weaver, S.A. McKee. “**Hardware Profiling of PARSEC for CMP Design**”, International Conference on Supercomputing (ICS’09), New York, June 2009.
2. V.M. Weaver, S.A. McKee. “**Optimizing for Size: Exploring the Limits of Code Density**”, Architectural Support for Programming Languages and Operating Systems (ASPLOS’09), Washington DC, March 2009.
3. V.M. Weaver, M. Schulz. “**A Cache Conflict Analysis Tool**”, Poster UCRL-POST-214300, LLNL, 11 August 2005.

Technical Reports

1. V.M. Weaver, S.A. McKee. “**Can Hardware Performance Counters be Trusted?**” Technical Report CSL-TR-2008-1051, Cornell University, August 2008.
2. V.M. Weaver. “**Web Plotting: Using gnuplot and vis5d in a Web Environment**”, SBCCOM/ORISE End of Year Report, 19 August 2003.
3. V.M. Weaver. “**Hypertext Interfaces for Models: Modeling on the World Wide Web**”, ERDEC SEAP/GWU End of Year Report, 9 August 1996.
4. V.M. Weaver. “**Creating X Window Graphical User Interfaces for Text Programs**”, ERDEC SEAP/GWU End of Year Report, 9 August 1995.

Computer Architecture Software

1. http://www.csl.cornell.edu/~vince/projects/bbv_research/
Extensions to Qemu and Valgrind that enable collection of Basic Block Vectors for a variety of architectures.
2. <http://www.csl.cornell.edu/~vince/projects/qemu-trace/>
Extension to Qemu that enables generating memory traces for use in cache and memory hierarchy simulations.

Honors and Awards

- Most Beauteous Visuals, 2005 International Obfuscated C-Code Competition (IOCCC’05)
- Honorable Mention, 2009 Cornell University Library Book Collection Contest

Talks

- “Video Game Programming under Linux”
UM Linux Users Group, University of Maryland December 2000

Research Projects

Multi-Platform BBV Generation *2007-present*

The SimPoint reduced-execution methodology can speed up simulation times with only a minor degradation in accuracy. However, generating the Basic Block Vector (BBV) files needed by the SimPoint tool can be difficult. Previously only a limited number of tools on a limited number of architectures could generate these files. I extended Valgrind and Qemu to be able to generate these files, making BBV generation much easier. I also validated the results against the existing PinPoints implementation.

Fast Memory Hierarchy Simulation *2007-present*

When attempting to speed up simulations, shortcuts are often taken. I investigate which methodologies have what amount of error, and attempt to validate against real hardware. After investigating single-core machines, the work is extended to look at multi-core. The end goal is to find the fastest most accurate method of simulating memory hierarchies.

Performance Counter Validation *2007-present*

In conjunction with my simulator validation and BBV generation work, I have investigated the accuracy and causes of divergences in hardware performance counters. This has involved gathering performance counter data for SPEC CPU2000 and CPU2006 on a wide variety (more than 10 types) of x86 machines, as well as several RISC machines. Improvements to the `perfmon2` Linux performance monitoring project have been merged upstream.

SimSnap *2004-2006*

The SimSnap methodology uses user-level checkpointing to speed simulations, by enabling instant-loading of interesting portions of a program without having to fast-forward. My research was adding file I/O save/restore to the checkpointing method, so that benchmarks utilizing I/O could be handled.

Invalid Heap Access Detection *2004-2005*

I worked on a project to use the OWL hardware monitoring infrastructure to catch invalid heap accesses. A library intercepts memory allocation calls, and informs the hardware monitor of valid ranges. A small set of ranges is cached in hardware and checked at each memory access. If a range is not cached, a trap into the OS acts as a fall-back mechanism. Initial results show cache misses are infrequent enough to allow invalid access detection with much lower overhead than a pure software implementation.

Cache-Conscious Data Placement *2004-2005*

I investigated the feasibility of using DBI tools to aid in cache-conscious data placement. The tool related conflicting addresses back to the corresponding data structures, including dynamically allocated structures, that traditionally are hard to track. The resulting conflicts were graphed and gave insight into which structures were causing poor cache behavior.

Research Advising

Former Students

1. I-Chun Li, M.Eng ECE Cornell University, 2006, thesis: “**Comparing Two Implementations of a Memory Reference Analysis Tool.**”

Teaching Experience

ECE/COMS 314 Computer Organization (Cornell University)

Spring 2008 (185 students): **Head TA**

Taught weekly 75-minute section (45 students), maintained course simulation tools, organized 6 other TAs, gave one full-class lecture

Spring 2007 (155 students): **TA**

Taught weekly 75-minute section (25 students), maintained course simulation tools, organized review sessions

Spring 2006 (172 students): **TA**

Taught weekly 75-minute section (25 students), maintained course simulation tools, organized review sessions

Computer Skills

Setup and maintained 4 large compute clusters at Cornell (16, 40, 40 and 44 machines respectively). Maintained a network of 10 different architectures including 169 machines totalling over 468 cores.

System Administration experience: Linux (13 years) and Unix (IRIX and Solaris) (14 years).

Wrote and maintain various open-sourced tools for Computer Architecture research.

Fluent in C and Assembly Language (20 kinds). Previous experience with Fortran, Pascal and BASIC.

Video game programming experience, both 2D and 3D.

Follow Linux kernel development; have had patches accepted into the mainstream Linux kernel (performance counter, parallel port, and minor fixes)

Author and maintainer of several freely available Linux utilities including linux_logo, fontprint, ansi2gif.

Contribute to various open-source projects, including (but not limited to) valgrind, Qemu, gphoto2, and perfmon2