



*Guest Editors' Introduction*

# COMPUTER ARCHITECTURE SIMULATION AND MODELING

**Timothy Sherwood**  
University of California,  
Santa Barbara

**Joshua J. Yi**  
Freescale Semiconductor

..... In an era in which designers navigate the entangled problems of power consumption, performance, parallelism, thermal effects, and reliability, we find ourselves increasingly dependent on models and simulation to ensure that our designs will meet expectations. Suboptimal decisions made in the early stages of design are expensive or impossible to remedy later, and detailed simulation is really the only option for exploring cycle-level effects. With each new processor generation, designs are growing larger and more complex, and while we find ourselves increasingly dependent on simulation, it is a constant challenge to ensure that our simulations remain both fast and accurate.

As the guest editors of this special issue, we are pleased to introduce a collection of articles that highlight some of most interesting aspects of simulation and modeling as they apply to computer architecture and system design. While quantitative analysis is one of the most fundamental aspects of computer system design, finding a proper simulation methodology is all too often either overlooked or relegated to secondary consideration.

Detailed simulation is very expensive, requiring carefully crafted simulators, realistic workload scenarios, and countless machine cycles. The required level of detail comes at the cost of speed; even on the fastest simulators, modeling the full execution of a single benchmark can take weeks or months to complete.

The industry-standard SPEC (Standard Performance Evaluation Corp.) CPU2000 benchmarks call for the execution of a suite of 26 different programs, with a combined total of approximately eight trillion instructions. Exacerbating this problem is the need to simulate each benchmark for a variety of different architectural configurations and design options. Two of the articles we have selected, "SimFlex: Statistical Sampling of Computer System Simulation" and "Efficient Sampling Startup for SimPoint," attempt to address this problem through clever techniques that allow execution of only a small subset of each program.

In addition to tackling the problem of raw simulation performance, future modeling and simulation infrastructures must help us navigate the increasingly complex interactions found in modern systems. Many factors determine the real performance of a modern computer: ambient temperature, workload, multiprocessor effects, power density, network interactions, interactivity, overhead from the operating system and managed runtimes, I/O, soft error rates, battery life, and a host of other parameters outside the traditional scope of computer architecture simulation. Unfortunately, many of these parameters interact in strange and complex ways, which expands the design space. For example, an increase in temperature induces increases in leakage power, which could create a power density problem and reduce reliability. This is just one exam-

.....

**Only when the ideas in these articles make the leap from academic enterprise to engineering solution will our field have succeeded.**

.....

ple of such an interaction; designers could explore many others if sufficient simulation techniques could be developed. To this end, three articles in this special issue, “IPC Considered Harmful for Multiprocessor Workloads,” “Using STEAM for Thermal Simulation of Storage Systems,” and “The M5 Simulator: Modeling Networked Systems,” address different aspects of system-level simulation methodology.

Each of the articles selected for this special issue focuses on a new simulation or modeling method that is either easier to use, faster, or more accurate than its predecessors. A variety of venues specialize specifically in this line of research—most notably the Workshop on Modeling, Benchmarking, and Simulation (MoBS) held with the International Symposium on Computer Architecture (ISCA); the IEEE International Symposium on Workload Characterization (IISWC); and the IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS). The articles in this special issue, however, are the result of open submission. Although they represent only a sampling of the work in the field, we think that this special issue highlights some of the most interesting and practical schemes. Ultimately you, the *IEEE Micro* readership, will determine the utility of these methods. Only when the ideas in these articles make the leap from academic enterprise to engineering solution will our field have succeeded.

We initially received a total of 17 submissions, each of which received at least three reviews from our panel of anonymous reviewers. The anonymous reviewers represent some of the most respected experts in this area, and without their timely and thoughtful opinions, this special issue would not have been possi-

ble. From these reviews and their scores, we, the guest editors, made our recommendations to Pradip Bose, *IEEE Micro*'s editor-in-chief, who then made final decisions.

In the first article, “IPC Considered Harmful for Multiprocessor Workloads,” Alameldeen and Wood take a careful look at the practice of using instructions per cycle (IPC) as a performance metric for multiprocessor workloads. The authors lay out a series of situations where the use of IPC, to the exclusion of work-related metrics, can lead to poor design choices. Although the use of IPC has served our community well for many years because of its simplicity and intuitiveness, as we move to a world increasingly dominated by multiprocessor systems the relationship between instructions executed and wall-clock time begins to blur. In addition to presenting strong quantitative evidence supporting their thesis, the authors take care to illuminate many of the pitfalls lurking in multiprocessor performance evaluation.

Whether IPC is your metric of performance or not, slow simulation is a problem that everyone faces: Wenisch et al. address this issue in the second article, titled “SimFlex: Statistical Sampling of Computer System Simulation.” SimFlex is a simulation infrastructure that applies rigorous statistical sampling theory to the simulation problem, with the goal of having both high accuracy and confidence in the estimates. Statistical sampling on a computer system is no easy task. Unlike samples from a simple population, a performance estimate based on a sample of execution time is highly dependent on a great deal of state. For instance, the performance estimate from a sample of 100 executed instructions would be meaningless if the cache and branch predictor were full of invalid entries. Wenisch et al. provide an overview of their SimFlex toolset and describe how it addresses these and other problems related to statistical sampling.

Rather than statistical random sampling, the goal of the SimPoint project is to apply machine-learning techniques to pick the most representative samples from a program. However, whether samples are chosen carefully or randomly, the problem of maintaining sample state remains more or less the same. In “Efficient Sampling Startup for SimPoint,” Van

Biesbrouck, Calder, and Eeckhout examine several state-saving techniques, including touched memory image and memory hierarchy state, in an attempt to drastically cut simulation time.

Changing gears completely, the fourth article, “Using STEAM for Thermal Simulation of Storage Systems,” presents a simulation system that takes into consideration the thermal and physical characteristics of hard drives. Modern storage systems, much like modern computing devices, are often thermally limited. As storage systems play a huge part in the performance of many real systems, Gurumurthi, Kim, and Sivasubramaniam have developed a simulator that captures both the thermal and physical realities of the media to enable design exploration at the level of machine organization and above.

As modern machines are deployed in an environment saturated with devices for communication and storage, it is very likely that system-level effects, such as the storage system trade-offs discussed in the article on STEAM, will continue to grow in importance over the coming years. With that growth will come a rising need for full-system simulation to place individual effects (operating system, storage, network, and so on) within the broader context of a complete design. The M5 simulator system is an impressive toolkit that can boot an operating system, supports multiprocessing in a modular way, and uses a fully event-driven memory system. In their article, “The M5 Simulator: Modeling Networked Systems,” Binkert et al. describe one capability of their simulation infrastructure, the modeling of fully detailed machines in a network. Their infrastructure has already aided in the evaluation of several interesting optimizations, and due to their liberal licensing policy, M5 is likely to be of interest to both academics and industrial researchers.

Although the desire for faster and more accurate simulation is by no means satiated, the articles in this special issue demonstrate that the research community has started to turn its focus to this critical problem. Already, several of the top conferences and journals have seen a flurry of activity aimed at easing the burden of simulation, and techniques for simulation are receiving a level of interest that makes them a research area in its own right. Several techniques are now mature enough to

be considered seriously by practitioners. As the editors of this special issue, we are encouraged to see the scope of *IEEE Micro* expand to encompass these important areas.

We thank Pradip Bose, Lizy Kurian John, and the *IEEE Micro* staff for their guidance in assembling this special issue, and the anonymous reviewers for their insight and hard work. We are grateful to all the authors who took the time to submit their manuscripts to *IEEE Micro*, and we hope you enjoy this special issue on simulation and modeling.

**Timothy Sherwood** is an assistant professor in the Computer Science Department at University of California, Santa Barbara. His primary research interests are in architectures and techniques that allow the continuous streaming analysis of systems, including network processing, security accelerators, and methods for program phase analysis and introspection. He has a BS from University of California, Davis, and an MS and PhD in computer science from University of California, San Diego. He is a member of the ACM and the IEEE.

**Joshua J. Yi** is a performance analyst at Freescale Semiconductor in Austin, Texas. His research interests include high-performance computer architecture, simulation, benchmarking, low-power design, and reliable computing. He has a BS, an MS, and a PhD—all in electrical engineering—from the University of Minnesota in Minneapolis. He is a member of the IEEE and the IEEE Computer Society.

For further information on this or any other computing topic, visit our Digital Library at <http://www.computer.org/publications/dlib>.