Performance Evaluation and Its Impact on Design

LIEVEN EECKHOUT
Ghent University

In my editorial column in the Sept./Oct. 2015 issue, I talked about the structure of computer system research and development. Let me briefly recap how a system researcher or designer—the architect, for short—typically operates. The architect takes a hypothesis about a novel design—typically based on intuition, insight, and/or experience—and subsequently designs an experiment to validate this hypothesis. This means that the architect picks a baseline and a number of workloads to evaluate the proposed design. The experiment is run and measurements are obtained using an analytical model, or through simulation at some level of detail, or by doing a real hardware experiment. The experiment may or may not support the architect's hypothesis. If it does, the experimental design is improved and additional experimentation is done—that is, the design is incrementally refined. If the experiment does not support the hypothesis, the architect must reexamine the hypothesis, which may mean changing and improving the design.

Performance evaluation clearly is at the crux of computer architecture research and development. Rigorous performance evaluation is absolutely crucial to make correct design decisions and drive research in a fruitful direction. Clearly understanding how the experimental design is set up and how this can affect the results is absolutely necessary to make meaningful and valid conclusions. There are many aspects to performance evaluation, including picking workloads, a baseline design, and an appropriate modeling approach; running the model; and interpreting the results correctly. And each of these aspects is equally important—a performance evaluation methodology that lacks rigor in one of these crucial aspects could fall short. For example, picking a representative set of benchmarks, picking the appropriate modeling approach, and running the experiments in a rigorous way may still be misleading if inappropriate performance metrics are used to quantify the benefit of one design point relative to another. Similarly, inadequate modeling—for example, a simulator that models the architecture at an inappropriate abstraction level—could either underestimate or overestimate the performance impact of a design decision, even though appropriate performance metrics and benchmarks are being used.

Architects generally are well aware of the importance of adequate performance evaluation, and therefore pay detailed attention to the experimental setup when evaluating research ideas and design innovations. Arguably, the most difficult part to get right is to realize that experiment design in systems research and development involves a subjective human aspect—the step of picking a baseline design and workloads in systems research is typically based on the experimenter's judgment and experience. Also, the choice of performance evaluation method, be it a first-order model or a detailed simulation model, impacts the conclusions that can be drawn from the experiment. The architect must be conscious of this human aspect when interpreting and analyzing the results—trusting the results produced through the experiment without a clear understanding of its design and the assumptions made along the way could lead to misleading or incorrect conclusions.

This issue features an article called "Architectural Simulators Considered Harmful" by Tony Nowatzki et al. from the University of Wisconsin–Madison (p. 4). The article emphasizes the importance of an appropriate evaluation methodology. In doing so, it focuses on a particular performance evaluation method, namely architectural simulation that provides cycle-level performance and power estimates without relying on an RTL-level specification of a particular machine. The authors identify three pitfalls in architectural simulators and suggest how to avoid these pitfalls. Certainly, this article will not be the last word on the topic. Instead, it is likely to stimulate a healthy discussion on the appropriate use of simulation technology in research and development.

The issue also features three more general-interest articles on a variety of topics. In "Comprehensive Circuit Failure

Prediction for Logic and SRAM Using Virtual Aging," Amir Yazdanbakhsh et al. propose a technique to predict failures to address wear-out and extend processor lifetime for logic and SRAM (p. 24). "The Combined Input-Output Queued Crossbar Architecture for High-Radix On-Chip Switches" by Giorgos Passas, Manolis Katevenis, and Dionisios Pnevmatikatos proposes a novel high-radix switch architecture with improved bandwidth (p. 38). And "Design and Optimization of a Horizontally Partitioned, High-Speed, 3D Tree-Based FPGA" by Vinod Pangracious, Zied Marrakchi, and Habib Mehrez makes the case for a 3D tree-based FPGA architecture (p. 48).

Finally, this issue features two articles from Cool Chips XVIII, the IEEE Symposium on Low-Power and High-Speed Chips, held in Yokohama, Japan, in April 2015. "A Silicon-on-Thin-Buried-Oxide CMOS Microcontroller with Embedded Atom-Switch ROM" by Toshitsugu Sakamoto et al. describes an ultra-low-power microcontroller unit for the Internet-of-Things (IoT) era (p. 13). "An Open Approach to Auton-omous Vehicles" by Shinpei Kato et al. pro-poses an open platform for researchers and developers to study and explore autonomous vehicles (p. 60). I'd like to wholeheartedly thank professors Fumio Arakawa (Nagoya University) and Makoto Ikeda (University of Tokyo) for inviting and monitoring the review process for these two very interesting articles.

With that, I conclude, and wish you happy reading, as always!

Lieven Eeckhout
Editor in Chief
IEEE Micro

Lieven Eeckhout is a professor in the Department of Electronics and Information Systems at Ghent University. Contact him at lieven. eeckhout@ugent.be.

---

cn *Selected CS articles and columns are also available for free at http://ComputingNow. computer.org.*