Performance Evolution

Vijay S. Pai
Department of Electrical and Computer Engineering
Rice University – MS 380
6100 South Main
Houston, TX 77005

Looking back over the past 15 years, the architectural community has seen considerable consolidation in choices of quantitative performance evaluation methods. Previous generations of architecture conferences featured a gamut of evaluation schemes for new architectural proposals, including mean-value analysis and other analytical models, trace-driven simulation of various system components, execution-driven simulation based on application instrumentation, instruction-level emulators, and detailed microarchitectural simulators. Workloads also were highly divergent, often test cases hand-crafted to expose some particular architectural feature. In contrast, the overwhelming majority of papers published in major architectural conferences today use variants on a few widely-distributed detailed microarchitectural simulators (e.g., Simplescalar and, to a lesser extent, RSIM), and workloads focus primarily on SPEC, with some contributions from Olden, SPLASH, and TPC. Even publications on emerging areas of concern such as technology and power have standardized to a small number of simulators.

This paper contends that such consolidation has unwarranted side-effects on the overall health of the community, based on two axioms: that performance evaluation is important to architecture, and that science is primarily an evolutionary process. In particular, we examine the benefits of diversity through an evolutionary paradigm.

Diversity prevents epidemics. A recent example of a pitfall in the current status quo was provided by an algorithmic analysis of the Health benchmark of the Olden suite which showed that orders of magnitude performance improvement were available from minor code changes [3]. This particular finding has serious implications for our community, since many microarchitectural proposals are justified with results showing their greatest benefits on Health. The dearth of evaluation schemes and worloads implies

that the choice to use those schemes can be made without analysis, with possible consequences for the quality of research. By insisting on analysis of all algorithms, software implementations, and hardware choices made, we can help to promote the diversity needed to study diverse architectural proposals.

Diversity promotes cross-pollination. Use of diverse evaluation methods and workloads should also encourage growth as researchers have a greater base with which to analyze their proposals, borrowing bits and piece of infrastructure from various sources and possibly contributing back their changes. Having alternative approaches also enables serious comparisons and reconsiderations of published work, akin to the review process in more traditional scientific fields. Contributions in diverse orthogonal subfields of performance evaluation allow multiplicative growth in the alternatives available.

Diversity helps to avoid vestiges. Our acceptance of a few key benchmarks gives architects an incentive to ignore the system features not exercised by those benchmarks. Good examples are networks for our current widely-accepted benchmarks and disks for SPEC, SPLASH, and Olden. Although TPC uses disk more extensively, more recent papers have ignored this impact since earlier results showed that certain configurations could minimize the resulting performance impact. It is not clear, however, if those configurations are practical or common for deployment of commercial servers.

Diversity is essential for evolution. Natural selection can only take us so far, as the process is limited by both environmental factors and inputs. If the review process continues to accept papers that do not seriously analyze and justify their evaluation choices, the major environmental factors are already determined. If the base of available infrastructure also remains small, there simply may not be enough possible choices or interactions to encourage the right kind of progress, particularly given rapid changes in the overall environment.

Introducing new diversity. If science is primarily evolutionary, then we also arrived at our status quo through evolution. This seems believable, since our workloads and simulators have been selected with the review process as the primary environmental factor. Evolution also implies some degree of consolidation, but depends on successful mutations being propagated to break out of such equilibrium. Although nearly all

research requires some mutation to the evaluation infrastructure, such mutations are rarely propagated. We can thus encourage evolution by promoting broader publication of not only full simulators and workloads, but even patches to existing ones. Additionally, we can look to the demetic subcommunities within our field. We have seen a recent uptick of papers on analytical and statistical simulation in architectural conferences, but their ideas have not yet been widely adopted for evaluation of architectural proposals [1] [2]. Such methods, based on formerly accepted evaluation schemes, are still viable today and actually enable limited evaluation of software and hardware ideas even before fleshing out their implementations. Additionally, we can look to the vast gene pool outside of our field. For example, better consideration of workloads from the operating-system and networking communities may help us find new uses for what we now consider vestiges. Such choices may be essential for allowing continued growth and the emergence of new areas.

References

[1] Mark Oskin, Frederic T. Chong, and Matthew Farrens. "HLS: Combining Statistical and Symbolic Simulation to Guide Microprocessor Designs." In *Proceedings of the 27th Annual International Symposium on Computer Architecture*, June 2000.

[2] Daniel J. Sorin, Vijay S. Pai, Sarita V. Adve, Mary K. Vernon, and David A. Wood. "Analytic Evaluation of Shared-Memory Systems with ILP Processors." In *Proceedings of the 24th Annual International Symposium on Computer Architecture*, June 1997.

[3] Craig Zilles. "Benchmark Health Considered Harmful." *Computer Architecture News*, June 2001.