Report on the Panel: “How Can Computer Architecture Researchers Avoid Becoming the Society for Irreproducible Results?”

The First High-Performance Computer Architecture Symposium, Raleigh NC,
Tuesday 24th January.

Trevor Mudge, The University of Michigan, Ann Arbor

The High-Performance Computer Architecture Conference, although young even among computer conferences (the second is scheduled for February 96), is already recognized as one of the leading forums for computer architecture research. At the first HPCA earlier this year I organized a panel with the above title. It was purposely tongue-in-check, however, the intent was to provoke discussion on a point that distinguishes much of the experimental work in Computer Architecture from that of traditional experimental sciences: many results that are published are difficult or impossible to confirm. As a starting point for the panelists I suggested the following reasons for this irreproducibility:

1. The authors: they give incomplete information about their experimental procedures.
2. Lack of access to the relevant experimental setup.
3. The reward system: their is no kudos for validating experiments (unlike in the bio sciences) — this means no tenure or research funding for confirming experiments.

We were very fortunate to have a distinguished panel, who represented a wide range of experience in experimentation and system building. The format of the panel gave each speakers one overhead transparency and 2 minutes to state his position. The panelists were:

- Tilak Agerwala, IBM
- Tom Conte, North Carolina State University
- Michel Dubois Unversity of Southern California
- Michael Foster, National Science Foundation
- Jim Goodman, University of Wisconsin
- Paul Schneck, Mitre Corporation

The panel was a lively one, with plenty of audience participation. A consensus developed that favored fostering a culture closer to the traditional sciences as far as experimentation is concerned. However, there was by no means unanimity on this point, and several panelists pointed out that the rate of change in computing is so great that by the time results are confirmed they may no longer be of any relevance. A stronger case needs to made if any change in the methodology of computer architecture research is to occur. I asked the panelists to summarize their points in a paragraph or two. They are included below.

Tilak Agerwala, IBM

The goal of architecture research is not to produce innovative system architectures but to provide a bridge from basic technologies to tangible end user value. The onus to demonstrate value is first and foremost on the architect. To accomplish this, architecture researchers often rely heavily on simulation (or experimental test beds) driven by a limited set of applications. Without a scientific approach to workload characterization, such validation is quite insufficient.

My basic position on the panel was that architecture researchers need to adopt a methodology based on first principle reasoning and hypothesis validation. Starting from applications and user environments, architects should clearly describe the problem being solved, state the simplifying assumptions, and provide the technical reasoning behind the proposed architecture. Simulation and prototyping should be used as supporting evidence and not as the primary or sole justification. System architecture research is fundamental to the success of many leading edge products, and the current competitive environment, with its time-to-market pressures, does not allow for too many false starts. To provide significant value in this environment, the architecture community must set high standards for reproducibility. In addition to researchers adopting a "scientific" methodology, referees should not accept papers where reproducibility is clearly questionable, and funding agencies should make reproducibility a criteria for success.

Tom Conte, North Carolina State University

Reproducibility for architecture has two components: de facto standards for experimentation (i.e., conventions), and a forum for publication of reproductions. There already exist conventions for benchmarks, developed by consensus. They are not perfect, but they continue to evolve and improve thanks to public debate. There are also conventions for reporting data (e.g., the miss ratio, IPC or harmonic mean) that also continue to evolve. We do not yet have conventions for performance evaluation. Most modeling today is done in C and protected by the authors. These simulators are often considered "verified" when they no longer core dump! (Similar problems exist for compilation, where hand-compilation and unstable/ benchmark-specific compilers are all too common.) Implementation is one solution, although it is expensive in both time and money. Another possibility is a new method for construction of architectural simulators. The emerging field of reconfigurable logic offers a hybrid between the two. Whatever the convention becomes, the intellectual value should be in the architectural ideas, not in their evaluation. Architects should not feel as though they are "giving something up" by publishing enough details to reproduce their results.

The second component of reproducibility is a public forum. This can begin if a symposium solicits for and accepts reproduction papers. Whistle blowers are often penalized rather than rewarded. Highly-regarded senior architects must take the lead. It will be a rocky first few years, but in time the quality of all published results will increase. Perhaps this forum should instead take the form of a watchdog organization. The sciences have groups that hound "paranormal science," perhaps we need a similar organization to uncover "paranormal architecture.'

Michel Dubois, University of Southern California

Computer architecture deals with extremely complex systems and is more an engineering art than a science. Recommendations based on simulation results can often be argued different ways. For example, if I advocate mechanism X and "simulations show that X is better than Y by 15%", I can support X with the data. But, if I advocate Y, I could also say that "15% is not much and does not justify the higher cost associated with X". Given that only a few test programs were run, that the real costs of X and Y are hard to evaluate without the full hardware design, that all kinds of approximations were made in the simulation, and that data set sizes are usually drastically reduced, it should be clear that 15% does not mean much. In this respect, I find that many papers claim too much in their conclusions based on simulation evidence alone.

Nevertheless, simulation results are important in a paper because they demonstrate that the authors have looked carefully into the details of their proposed design. Many times simulations yield new insights about the design, uncover oversights and lead to new ideas. However, I do not believe that a practicing engineer in his right mind would trust the simulation numbers in a paper without first doing his own evaluations.

Finally the idea of having a "watchdog" mechanism to repeat results published in architecture papers does not seem appealing to me. It would create a huge overhead and slow down the propagation of ideas and results. For one, most people don't really care that much about most papers' results. And if one really cares, I would suggest that he should contact the authors directly, as is currently done. Authors who refuse to let someone reproduce their results should not be trusted.

Overall, the problem may lie with readers, who should be more critical of the conclusions and more attentive to the details of every paper they read.

Michael Foster, National Science Foundation

If results in computer architecture are to be routinely reproduced or validated, we need to make some cultural changes. These changes cannot be imposed from outside, but must arise from discussion and agreement within our community. The panel meeting at HPCA, along with these summary write-ups, may be the first step in that discussion.

The difficulty with validating results is the myriad of details in a simulation or experiment that may affect the measurement. Reproducing a result means determining which details are important and which are inessential, then reproducing all of the important details. This already difficult job may be made more difficult if a culture of secrecy becomes entrenched.

To make validation possible, both originators of results and validators must cooperate. Originators must make the unique simulators, traces, and other experimental apparatus used in their measurements available to potential validators. This may extend to allowing net access to special hardware used in an experiment. Making these artifacts available is the best way to ensure that all essential elements of a measurement can be reproduced.

Validators also have some duties; the burden does not fall completely on the originators. Most obviously, the validators must be willing to invest time and effort to learn how to use whatever tools the originators make available. This task is tailor-made for junior researchers, since it provides a good introduction to the field, the methods, and the active researchers. The validators must also find some way to provide credit to the originators of the research. If someone tries to reproduce your research, you must take time to provide tools, and run the risk that some faults will be found with your methods. Clearly, some motivation beyond "the good of computer architecture" is needed to encourage you to help in the task.

Many fields, including physics, biology, and astronomy, have experimental results that require help from the originators to reproduce. The help needed ranges from access to unique instruments to preparation of samples and standards. Techniques for allowing the validators to credit the originators of a result vary among these fields. Co-authorship and monetary payments are common in some fields; in others, independent allocation committees assign time on unique instruments to researchers who may not have participated in the construction of the instruments. A discussion in the computer architecture community may unearth mechanisms that will work in our own field.

Jim Goodman, University of Wisconsin

Computer architecture has come under increased pressure to produce quantitative results. Though quantitative results are valuable, it is important to establish the scope to which such results may be applied. Particularly in the use of simulation, which is very easily performed today, but very hard to specify sufficiently to permit reproducibility, it is easy to generalize results to situations where they don't apply, or to overstate the accuracy of the results.

Papers reporting results of simulation should be held to a high standard. If the referee is not convinced that the results could be reproduced based on information in the paper, including references to more detailed material, the paper should be rejected.

Paul Schneck, Mitre Corporation

The issue at hand (how to avoid...) has at its origins the fact that many (most?) students of computer science are not educated as scientists. They are trained as programmers. This results in a situation, reflected in our literature, where many practitioners form unstructured phenomenological inferences instead of creating models, forming hypotheses, and performing experiments to validate (or invalidate) the hypotheses and models. In short, the advice is to study single variables when possible. When not possible, design a multivariate experiment to isolate the effects of individual variables. Finally, provide sufficient data (and perhaps access to the experimental equipment) to enable others to reproduce the results. ESCHEW PHENOMENOLOGY!

It is important to keep in mind that (non-theoretical) computer science is an exercise in engineering economics. As Turing proved, all computers are equivalent (except for issues of speed). That is, a problem that can be solved by one computer can be solved by any other. Differences in speed mean that some machines cannot solve a particular problem in a timely fashion, while others can. Differences in cost mean that some machines are more cost-effective than others. It is the computer scientist's challenge to create a) new mechanisms for delivering increased speed, while improving cost effectiveness; or b) new mechanisms for improving cost effectiveness at a particular level of speed. We see immediately that industrial funders are likely to be relatively uninterested in pursuing results that do not attain these goals, or even of providing a roadmap (for their competitors) of those results that do attain these goals. Generally, government funders are focused on the intellectual content and less on the specifics of implementation and cost-effectiveness. Consequently, there is unlikely to be significant incentive for documenting experiments that highlight the (resource aspects of) performance and trade-offs.