The Use and Abuse of SPEC: An ISCA Panel

THE MAJORITY OF PAPERS PUBLISHED BY COMPUTER ARCHITECTURE CONFERENCES USE THE SPEC BENCHMARK SUITE TO MEASURE PERFORMANCE. HOWEVER, MANY RESEARCHERS USE ONLY A SUBSET OF THE SUITE. WHAT'S MORE SURPRISING IS HOW FEW PEOPLE EXPLAIN THEIR REASONS FOR SUBSETTING.

The 30th International Symposium on Computer Architecture featured an evening panel and dinner cosponsored by Intel and IEEE Micro. *To kick off this panel, Daniel Citron presented "MisSPECulation: Partial and Misleading Use of SPEC CPU2000 in Computer Architecture Conference" (*Proc. 30th Ann. Int'l Symposium Computer Architecture*, IEEE CS Press, 2003, pp. 52-59), his position paper from this year's ISCA proceedings.*

The ISCA program committee and IEEE Micro *thank moderator John Hennessy, and panelists Daniel Citron, Dave Patterson, and Guri Sohi for providing a lively and thought-provoking discussion.—Pradip Bose, editor-in-chief*

Hennessy: We wanted to talk about some of the ways in which performance numbers are gathered and published in papers. We're going to start with a presentation by each of the panel members. We allocated different amounts of time, according to how much they have to say that's useful.

First, Daniel Citron will come up; he's from IBM Israel, and he's going to talk about his work, which you've probably seen in the proceedings.

Citron: OK, let's look at the misleading use of SPEC.

Let's start with the scene. Many of you recognize the various places that SPEC use/misuse takes place—recent computer architecture conferences. I started to look at the ISCA, Micro, and HPCA conferences of the past three years. Each published a varying number of papers, and SPEC use was widespread: 209 papers were published and 66 percent of them used SPEC. Research for the earliest conference, ISCA 2001, was probably done by the end of 2000. But that was a year *after* SPEC announced CPU2000, so you had plenty of time to get the new benchmarks and use them.

Next, the victim: SPEC CPU. It's currently in its fourth version. SPEC CPU2000 is CPU intensive, and it measures the performance of the processor, memory, and compiler. And if you look at the paper breakdown [Figure 1], you can see that data path and memory papers are predominant in these proceedings, and they use SPEC; up to 90 percent of the papers that give values use SPEC.

OK, so what's the crime? You know what, I'll plead it down to a misdemeanor: the partial use of CPU2000.

0272-1732/03/$17.00 © 2003 IEEE

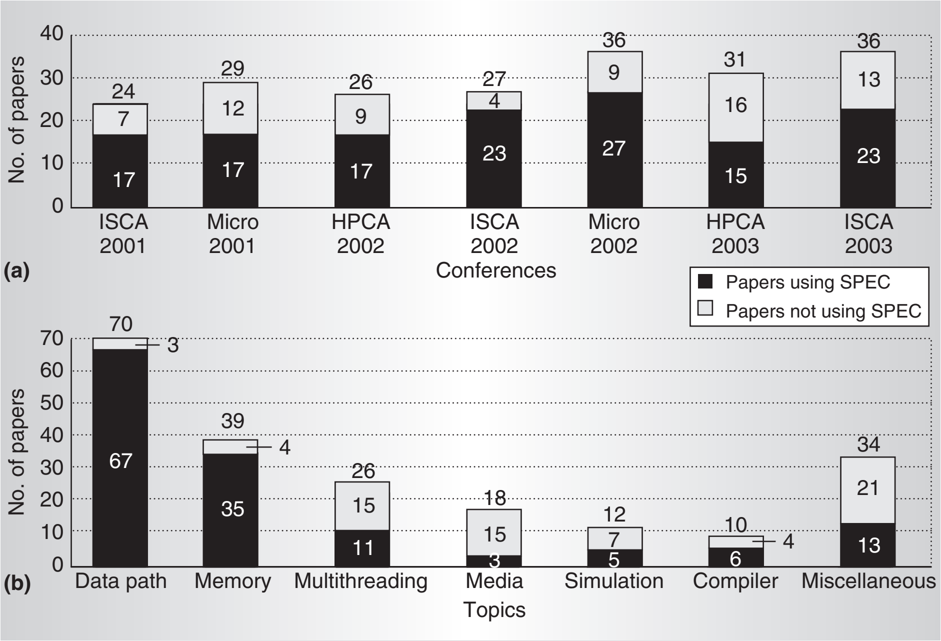

Figure 1. Papers in study categorized by conference (a) and topic (b).

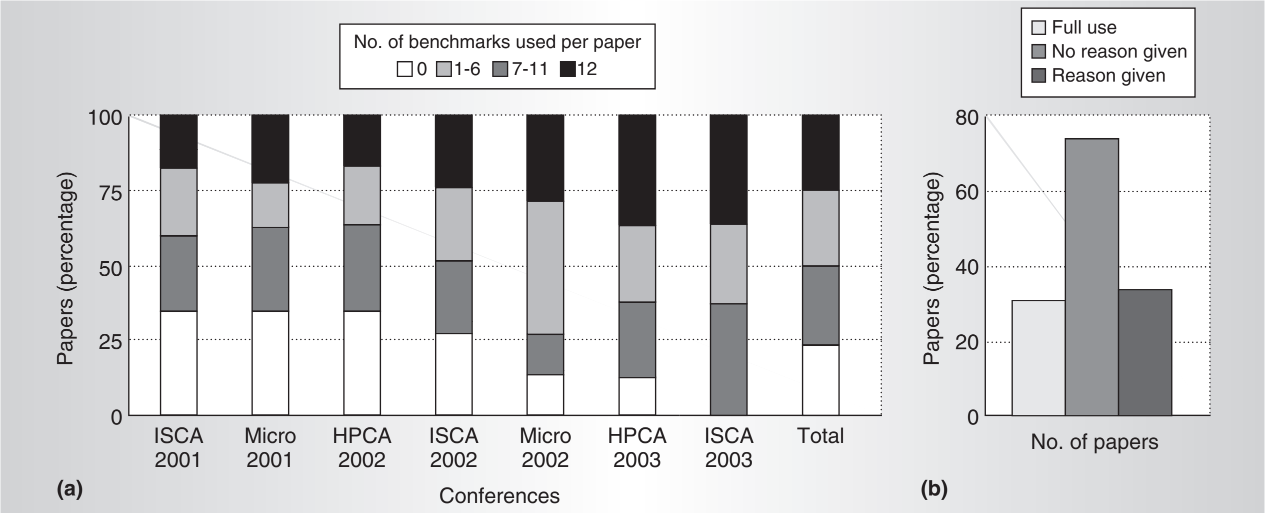

Figure 2. Number of benchmarks used per paper (a) and breakdown of SPEC benchmark suite use (b).

Let's look at the number of benchmarks used per paper [Figure 2]. In earlier conferences, many papers continued using CPU95, even after CPU2000 became available. And I'm happy to state that this is the first conference where there's probably not any use of CPU95. The few papers that did use it, also used CPU2000. But the full use of the suite varies from none to about 30 percent, if I aggregate all the papers. But what's even more disturbing is why—why are we not using the benchmarks? Only 30 papers using the partial suite explained anything; however, over 80 papers didn't give any reason at all. If you ask someone why, they would say "they didn't fit into our displays," "they didn't compile," or whatever. But in the paper, there's no statement; they don't give any explanation.

But some do. And it appears that the most leading explanation is "they weren't relevant to the problem." So we have this crime, and in many cases, there's no motive at all.

And let's look at some of the evidence: 105 papers used CINT2000, an average of eight benchmarks per paper. And I have to say that this ISCA conference raised this average by more than one benchmark per paper.

We also use some benchmarks more than others [Figure 3]. And the arguable reason—uncovered by our forensic lab—basically has to do with compiling, running, and simulating the benchmark.

It is intriguing that the top nine benchmarks used match those benchmarks that have MinneSPEC datasets available.

Let's go back and see if we have any criminal record here. For CINT95, the behavior is very similar: Only 50 percent of the papers used the whole suite. This is a small suite—it had eight benchmarks—and we have 12 now. And SPEC 95 was around for three years before I collected the data, and nobody explained their misuse of it.

Let's jump into the future; I'll project the frequency of full use in all the conferences. Even for this conference, there is a slight degradation, but that's not entirely true because a lot of papers had 10 or 11 benchmarks in their results. But if you project to 2005, you find that we'll be using all of our SPECs—pretty good. However by then, CPU 2004 should be out.

Let's try to fill in the missing results with similar numbers. You can see there that the speedups are close. If we assume a slowdown for several of the benchmarks, performance starts to degrade (of course), and if we assume that the missing benchmarks are invariant to the enhancement, the degradation is worst. The worst will always be assumed about the work of others. Most beautiful of all, the fewer benchmarks used, the higher the speedups.

Many papers that aren't about speeding up performance are about tradeoffs. What do we assume about the missing results? We now have a bunch of slowdowns and a bunch of missing results. So we can go down to 80 percent, 65 percent, or even lower. Of the 18 papers presented today, eight of them traded performance for other metrics.

Let's go to the closing arguments. SPEC CINT2000 is widely used in research, and we saw that only 20 percent of the published papers use the full-on suite. The projected adoption rate would be too late, basically, and the explanation for omitting benchmarks is not always given. Applying Amdahl's law to missing benchmarks steals the thunder out of published results.

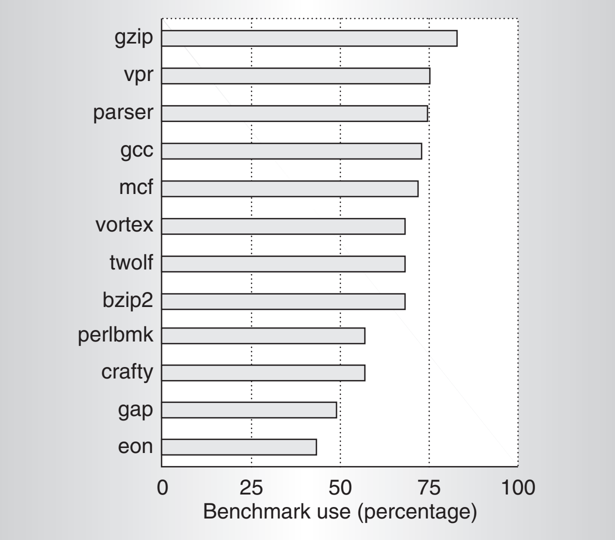

Figure 3. Benchmark use across papers in study.

Hennessy: Great; thank you. Daniel originally wanted to give the names of the authors of these papers; you can pay me not to disclose them. Dave is going to come up and tell us whether or not these numbers are purely misleading or whether or not there is some intentional deception at work.

Patterson: Although we're at a buffet, I don't think SPEC2000's a buffet. People work on these benchmarks for years. They have big discussions and tradeoffs about the number of programs that count. It is intended to be a workload. Certainly, if any company subsetted them, everybody would get upset.

Why is this going on? I thought there might be some technical reason. For example, the benchmark is written in a funny language that researchers can't compile. But that apparently isn't the case.

Why would people be afraid to run the whole suite? At this conference, 80 percent of all papers were rejected. I was on the program committee, and they have to know how the system works. Well, 70 percent of *their* papers were rejected. It's possible that people are subsetting because, as some people have said to me, "the results don't look too good if I include all the results." This is very disturbing if it's true.

Maybe the great solution isn't more extensive use of SPEC. Maybe the right direction is to start working on some other problems. It seems to me as an old-timer that some of the results that are positive now—low double-digit performance improvements—in the past were considered negative results. Maybe it's time for us to start working on other problems, if that's the way it's going.

It's possible that people are subsetting because, as some people have said to me, "the results don't look too good if I include all the results."

Hennessy: I've invited a famous mid-Western attorney to come up and defend the computer architecture community—none other than that famous scholar of legal mechanisms, that magician with words, Guri Sohi.

Sohi: I don't know what I'm going to defend or not defend, but Patterson asked me to participate on this, so it's great that we are telling Donny [Citron] that I trashed his paper for the ISCA review.

Let's see where we came from, what is SPEC's background. SPEC was developed by industry to compare the performance of different machines. And it was intentionally meant to cover a variety of different scenarios: For all the possible things that they would want to test their machines on, there is a different benchmark for each one of those. It so happens that SPEC ended up being a convenient set of benchmarks for academic use. It was never intended for academic use; it was not developed by academics for research.

Furthermore, as architects, we seem to have some very misguided expectations. We keep looking for the magic bullet, we want to come up with a technique that *on average* does well for a variety of scenarios. We want our technique to work very well for all benchmarks on average. One thing we forget is there is no single idea in computer architecture that works equally well across the board. And yet that's what we are asking; we are saying "Hey, give us this magic bullet that works equally well across the board, and try all these varieties of different programs that somebody else has defined."

So, for example, you [Citron] are expecting to get SPECfp results, so let us talk about this first. The paper [Citron's position paper], I believe, mentions that nobody uses SPECfp. Well, yeah, if you're doing a branch prediction, of course you don't use SPECfp—it just clutters up the paper. And it does, you know. A very good reason for not presenting data for all the benchmarks is because it clutters up the graphics.

So it's like asking a pharmaceutical company that comes up with this wonderful drug for cholesterol, "What is it going to do for this other disease?" I want you to do studies for all possible illnesses and tell me how your drug does, before I figure out whether it is any good.

Subsetting is also very useful, because it focuses discussion and provides insight. In fact, most papers in microarchitecture don't target floating-point, so why bother presenting data for floating-point? It's not needed.

Now I have some questions for the audience—and let's be honest. How many of you use the SPEC benchmarks? Everybody, come on, everybody does. How many of you have a license for the SPEC benchmarks? How many of you look at the source code? Come on; how many of you simply use the binaries that came with SimpleScalar? How many of you simply run those binaries?

I'll let you on a little secret on those binaries. Most papers use SimpleScalar Alpha binaries that have 20 percent no-op instructions for branch alignment. The common instruction for those no-ops is load offset from the stack, zero offset from the top of the stack. Lots of cache papers have been written; they probably use the entire SPEC benchmark suite. Is that meaningful? Or is a paper that talks about three benchmarks with a lot more insight, more meaningful?

There are innocuous reasons for subsetting—some binary not working in some case, such as when using some system call, using some simulation system, or some system for launching simulation. These really are innocuous reasons. Furthermore, when you are trying to come up with something really new, you're trying to do a different compiler, you're trying to do something different, you can't expect your tool chain to work for all the benchmarks. How many people's compilers compile all the benchmarks? So what we are saying is that we are trying to do this radical work with new compilers and new tools, but we expect them to run all the SPEC benchmarks, because I want to know how it does for *all* the SPEC benchmarks.

On the other hand, there's the obvious question that people hesitate to bring up, "Are people cooking data?" That's really the bottom of it, but everybody's unwilling to ask this question. I think for the most part, not; 95 percent of people are not cooking data.

When there is cooked data, I believe we deserve it. Why do we deserve it? We have a blind faith in numbers, and we ask for blind reviews. We let people who want to fudge their data hide their identities. And then, after the paper has been submitted and you've conned the program committee members, then you get a chance to revise it.

So what are my suggestions? Less reliance on numbers, of course, but I don't know if the community is willing to accept that. I believe we need to have less reliance on SPEC for new ideas. We need to go back to microbenchmarks, and perhaps, benchmarks developed by researchers, not by industry people that run 500 billion instructions, and do nothing useful. Livermore loops—people really trashed them but each one assessed different computing scenarios. We need to go back to that way of thinking. And what we also need to do, we need to have NSF-style certification for the contents of papers. We should hold the author responsible for what is in the paper.

By the way, there are people who cook data, and most of us in this room know who. But when you use a blind review process, you let people get away with it. And if you don't like these suggestions, I have an alternate suggestion that we can discuss at the SIGARCH business meeting.

What we charge SIGARCH to do is prepare paper-preparing scripts. In those scripts, first the PC members should check off what they expect. Then the scripts—every author is going to have them—should take these expectations and launch the relevant simulations and prepare the data tables. So nobody needs to know whether we've subsetted or anything like that—we've done everything the same as everyone else. We could even have some default text for linking: "As you can see from figure X, this is what happens." This will work great; it helps standardize the research and greatly simplifies the program committee's job. Is that what we want?

Hennessy: Well, Guri, that was very entertaining, but I think your defendant is still in jail. I did want to point out one thing with respect to your drug analogy: The largest selling drug of all time is Viagra. It was originally developed for heart disease, but it has a much more effective use right now. MICRO

About the participants

Daniel Citron is a computer scientist at the IBM Haifa Labs in Israel. His inability to obtain answers for missing results fueled the writing of his position paper, "MisSPECulation: Partial and Misleading Use of SPEC CPU2000 in Computer Architecture Conferences." Citron's research interests include processor microarchitecture, simulation environments, and parallel processing. He is a member of ACM SIGARCH and SIGMICRO.

John Hennessy, our moderator, is president of Stanford University. He initiated the MIPS project at Stanford. MIPS was one of the first three experimental RISC architectures. Hennessy is also well known as the coauthor (with panelist David Patterson) of two widely used textbooks: *Computer Architecture: A Quantitative Approach*, and *Computer Organization and Design: The Hardware/Software Interface*.

David Patterson is a professor of computer science at the University of California, Berkeley. He led the design and implementation of RISC I, likely the first VLSI reduced instruction set computer; it became the foundation for the Sparc architecture. Patterson is a member of the National Academy of Engineering and a fellow of ACM and IEEE.

Gurindar (Guri) Sohi is a professor in the Department of Computer Science at the University of Wisconsin at Madison. In the 1980s, he articulated a model for a dynamically scheduled processor that supported precise exceptions, a model widely adopted by several microprocessor manufacturers. His research group also developed the SimpleScalar simulator, a simulation toolset widely used for research and instruction. Sohi is a member of ACM and IEEE.