

## Ken Sevcik on Performance Evaluation

### *The Origin of Queueing Network Models*

[Ken Sevcik is Professor of Computer Science at the University of Toronto. He received his B.S. in 1966 from Stanford University and his Ph.D. in 1971 from the University of Chicago. Sevcik joined the faculty at the University of Toronto in 1971, and was Chair of the Department from 1990 to 1992. He also served as Director of the Computer Systems Research Institute (CSRI). His research interests are in the use of analytic models for performance analysis of resource allocation, scheduling and file structures in computer systems, computer networks, and distributed data management systems.]

**UBIQUITY:** Our colleague Peter Denning has called your work in the performance evaluation field "an exemplar of experimental computer science." Tell us about it.

**SEVCIK:** Well, the origins of performance evaluation for computer systems comes from queueing theory, which was established before computer systems were widely used. In the late '60s, we were able to take some of the early single server queueing models and use those to represent the original uni-programmed computer systems where just one user would use the computer at a time, and that was a pretty good model. Then time sharing systems came along, and that required us to move to another type of queueing model called the machine repairman model. It turned out that in two places -- both at MIT and at RAND Corporation -- early time sharing systems were successfully modeled by machine repairman models.

**UBIQUITY:** Can you give the gist of the repairman model?

**SEVCIK:** The machine repairman model was developed in industrial engineering. The idea was that you have a set of, let's say ten machines and one repairman. What you want to know is, if the machines occasionally break down, what's the average number of machines that are going to be in service at any time, given that the repairman keeps fixing broken machines? So at any time, most of the machines may be operating, but one or two or three may be broken down. That situation relates to a time sharing system in the sense that the customers of the time sharing system correspond to the machines, and the processor is trying to serve their requests one at a time. You can assess how many customers are queued up for service, since the analog is how many machines are broken down and awaiting repair. So a model that was developed in industrial engineering for a completely different purpose was used quite successfully to represent the early time sharing systems.

**UBIQUITY:** And how did your own model look differently at the problem?

**SEVCIK:** The next step was that computer systems, about 1970, stopped having a single resource of contention. That is, you couldn't treat them as a uni-programmed system where there was only one user at a time. Instead, you had a processor and you had some disk drives or a drum or a printer. Different customers would queue at different resources. So we needed to move from a single resource model to a multiple resource model. That was the step to "queueing network models," which took place right at the end of the '60s and in the early '70s. Jeff Buzen in his Ph.D. work at Harvard was the first to come up with an efficient algorithm to solve queueing network models so that they could be practically used in the performance evaluation of computer systems.

**UBIQUITY:** So is this a solved problem now?

**SEVCIK:** Well, I'd say it was a huge step forward when he came up with that. What Peter was referring to in that quote, I believe, was that parts of queueing theory have been exploited in very practical applications. There was a well-developed theory for multiple resource queueing models, but the critical step was to find algorithms for solving these models efficiently so that they could be put into use in business and industrial contexts, such as banks and insurance companies. The models allowed us to represent various alternative systems quite faithfully by representing each processor and disk as one resource in this multiple resource model.

**UBIQUITY:** Is there any ambiguity about "systems"?

**SEVCIK:** There's lots of ambiguity there. The key bifurcation is that queueing theory and queueing networks were applied both to computer networks (starting with the Arpanet and the work of Len Kleinrock and people at UCLA in the early days) and also to computer systems. Exactly the same techniques and tools were applied in the two domains and were very helpful in both. I, myself, worked on the computer systems side, where I was looking primarily at mainframe computer systems, largely within a single organization. Other people looked at wide-area networks. We were using the same tools but the problems were slightly different. In mainframe computer systems, one challenge you face is representing complex I/O subsystems. On the network side, there are issues like routing algorithms and layered communication protocols, which were examined quite thoroughly.

**UBIQUITY:** In the 1970s, computer scientists contributed a new way to understand the queueing network models, which they called "Operational Analysis." You had a role in that movement, and you used the Operational Analysis approach in your textbook on performance analysis. What is Operational Analysis and what was your contribution?

**SEVCIK:** Traditional queueing theory arises from the theory of stochastic processes, in which models are based on several mathematical assumptions that are difficult to relate to real computer systems. Operational Analysis, which was developed by Jeff Buzen and Peter Denning and their associates, is an alternative mathematical framework, in which the assumptions are understandable and testable with respect to real computer systems observed over a finite measurement interval. Essentially identical formulas are produced and applied in each approach. Operational Analysis is equally as rigorous and mathematically sound as the approach based on stochastic processes, but it is vastly easier to explain and justify to capacity planners and systems analysts who work with computer systems. Also, Maria Klawe and I were able to identify a number of realistic situations in systems where the stochastic assumptions were blatantly violated, while the Operational Analysis assumptions were satisfied. This helped explain why queueing network models had been successfully used to predict computer system performance even in situations where it was obvious that the stochastic assumptions did not hold.

**UBIQUITY:** Has this specialty, then, settled down to the extent that there's no particular controversy raging now?

**SEVCIK:** Yes, I think so. There was a golden age of this style of performance evaluation, and that occurred in the 1980s. At that time, mainframe computer

systems cost millions of dollars, so making a purchasing decision often meant betting the ranch. It was critical to know whether the decision was a good one or not. Also, there were a limited number of vendors, IBM, Amdahl and Unisys and a few others, so the enumeration of all the different configuration options at any point in time was fairly short. For that reason, we could take our models and more or less exhaustively represent all the possible choices one might make, and thereby give the purchasers a pretty good indication of what kind of performance they would get for their investment. So the high prices and the limited options made queueing network models and the software packages for solving them quite useful for answering questions about the system performance that could be expected in return for a multimillion-dollar investment.

**UBIQUITY:** How did you start?

**SEVCIK:** In the late 1970's, our group at the University of Toronto (which included Ed Lazowska, John Zahorjan, Satish Tripathi, Allan Levy, Scott Graham, and others) had already developed software packages that solved several variants of queueing network models efficiently. We then amalgamated and unified the techniques in a single package called MAP (Modeling and Analysis Package) with the sponsorship of Amdahl Corporation. Amdahl then distributed MAP commercially for a few years in competition with the clear market share leader BEST/1 from BGS Corporation, the company founded by Jeff Buzen and two partners. Amdahl benefited from the use of queueing network models because the models could quantify the price/performance advantage that Amdahl systems held at that time over their competitors.

**UBIQUITY:** How did you get involved in supercomputing?

**SEVCIK:** Well, in the mid '80s, I spent a sabbatical year working at NASA Ames Research Center and teaching a course at Stanford. I was hired for the NASA position by Peter Denning, who was at the time Director of RIACS, a research center at NASA Ames. My main job there was building a performance model for the Numerical Aerodynamic Simulator system that they were developing to perform computational fluid dynamics calculations to assess spacecraft designs. The NAS system was intended to become the first computer system to achieve a sustainable gigaflop in performance. It consisted of a complex of three Cray parallel computers plus many supporting computer systems to transfer, store and display the data. The task of building a queueing network model of the whole NAS system in order to predict how well it would work got me involved in parallel computing and supercomputing. I then kept working in that area for about five or six years more.

**UBIQUITY:** What impressed you most?

**SEVCIK:** While working in parallel computing, I always found that a good way to improve performance was to just wait a little while. The basic underlying technology was improving quickly and the prices were dropping. So you could work two years to put together a parallel system built on specific processors, and find that you hadn't really gained very much over the uniprocessor performance two years later. So the trend that paid off was the trend to distributed systems and clusters of work stations, where high performance is attained by assembling lots of cheap off-the-shelf components.

**UBIQUITY:** There were skeptics in those days who liked to make fun of parallel processing by comparing it to the idea of harnessing the power of 1,000 chickens to pull a large boulder. Have you ever heard that one?

**SEVCIK:** Not that specific one. I've probably heard some similar ones. Well, the 1,000 chickens probably won the fight in the end, in that the supercomputers that exist today are so specialized and so directed at a single application that they really aren't used in commercial practice. In the ordinary world, networks of workstations are effectively used all over the place now.

**UBIQUITY:** Are they at some point going to make completely obsolete the old single machines, like Cray?

**SEVCIK:** I think there will always be a place for special-purpose machines. IBM has followed up its Deep Blue project, which was the hardware-based chess-playing machine that was the first machine world champion in chess, with a line of machines of different kinds that are based on high degrees of parallelism. They potentially have some specialized hardware but they are aimed at one specific application, and sometimes at a single program. And I think that's where the most powerful computers in the world are today. They aren't really general-purpose any more.

**UBIQUITY:** What was your undergraduate work? Was it in computer science?

**SEVCIK:** They didn't have computer science when I was an undergraduate at Stanford. So I took my degree in mathematics and I took as many computer science courses as I could, and then went to graduate school in the field of Information Science, as it was called at University of Chicago.

**UBIQUITY:** Have you seen the field change much over the years?

**SEVCIK:** Absolutely. I guess the biggest change is the breadth. When I first went to graduate school in the late '60s, I could read all the journals in computer science -- the *Journal of the ACM*, *Communications of the ACM*, and a handful of others -- and keep up with all of them month-to-month and know what was going on and know all the major names in the field. Now, 30 years later, that's impossible. It's hard to believe there are so many sub-fields and so many different journals. It's a struggle to keep up even in my own areas of specialty.

**UBIQUITY:** So how do you actually do that?

**SEVCIK:** With the help of my graduate students. I make sure to point them in the right directions, and see if they can sift out what the important things are, and when they find something interesting, that gives me additional motivation to read it beyond what I would have had otherwise.

**UBIQUITY:** And what about with other fields? Information overload being what it is today, is it still possible to do much authentic interdisciplinary collaboration?

**SEVCIK:** It's difficult. I've always had some interest in the area of medical applications, both just applications of record-keeping in hospitals and also some of the data mining that arises in the bioinformatics area. But it's very hard to find people who are trained well in both of the two disciplines. So our department is now trying to hire somebody in bioinformatics, and we look at some computer scientists,

who've had some exposure to bioinformatics in the biology realm; we also look at some that come from the bioinformatics area, but haven't had the thorough training in computer science. But it's usually picking among people like that, because there's just too much to know for somebody to have a thorough training in both fields.

**UBIQUITY:** At the University of Toronto, how do you recruit students for graduate computer science work?

**SEVCIK:** It's a very competitive enterprise to try to get the top students, so when the applications come in we try to sift out the ones that we really want to try to appeal to in competing against the top universities throughout North American and the world. We then have telephone interviews with them and once we're confident that we really want to have them as our students, we invite them to Toronto for a weekend to show them around, see the school, and meet the other students who might be their future co-graduate students. That effort seems to be paying off.

**UBIQUITY:** How big is the department?

**SEVCIK:** We have approximately 250 graduate students in all, and that counts both people in the Masters program and the Ph.D. program.

**UBIQUITY:** That's pretty big, isn't it?

**SEVCIK:** It's quite big. About seven years ago, the Ontario government decided that we didn't have enough computer scientists and computer engineers. So they offered all the CS and ECE departments in Ontario a double-or-nothing proposition where if we doubled the output of undergraduate students they would effectively double our budget and double our space and double our faculty, and so forth. It was one of those offers that we couldn't refuse, because they wanted higher productivity from the university, and if we had said no then I think we'd have seen no resources for a long time. So we accepted the challenge and over the six years or so we've gone from 33 full-time equivalent faculty members to something in the vicinity of 60, essentially a doubling. And, likewise, our undergraduate enrolments have gone up and our graduate enrolments in proportion.

**UBIQUITY:** Among the various specialties, which ones seem to predominate?

**SEVCIK:** At Toronto, it's probably the general area that used to be called artificial intelligence, and that now includes machine learning, knowledge representation, natural language understanding, neural networks and several other sub-areas. We have a large number of really outstanding faculty members there. Also, our theory area has always been very strong.

**UBIQUITY:** What kinds of debate are going on now within the department about what ought to be done?

**SEVCIK:** Well, it's always a competition whenever the opportunity to hire comes along. All the different groups propose the best candidates they can, and the recruiting committee often has very hard choices in putting priorities on different areas. Recently, the university has asked us to do planning in a much more systematic way than we used to. Now we have to commit, in advance, not only to how many faculty we want to hire in the next five years, but we must also identify what their particular areas of specialty are to be. And to some extent, the university

will hold us to those commitments. In the past, if somebody really sterling came along in some specialty area that you hadn't anticipated, we were able to negotiate so that we could take advantage of such opportunities. That will still be possible, but the planning is stricter than it used to be.

**UBIQUITY:** What prompted the expansion and the new approach?

**SEVCIK:** This all happened just before the dot-com collapse, and Nortel Networks, a huge industrial force in Ontario back at that time, was the primary company that wanted it; Nortel claimed they could hire all the graduates from all the departments in Ontario for the unlimited future. And then, of course, with the dot-com collapse and the downfall of Nortel Networks, the demand for our computer science and computer engineering graduates was much more in question. It turns out that our graduates are all still getting jobs, but the doubling that the province felt was necessary probably had pretty bad timing in the sense that it came just before the technology collapse in a lot of areas.

**UBIQUITY:** But even with that collapse having been noted, do you feel that computer science has effectively become the new queen of the sciences?

**SEVCIK:** Well, there are certainly lots of measures on how we might say that. We have, by far, the largest number of graduate students in our entire Faculty of Arts and Sciences, and I believe that means in any department in the University of Toronto. As you know, in many schools the computing discipline has become a School of Computer Science or a Faculty on its own, rather than a Department. That possibility has come under discussion at our University, but we've never pushed very far with it.

**UBIQUITY:** Do you have much to do with undergraduates?

**SEVCIK:** Half my courses are graduate courses, half are undergraduate.

**UBIQUITY:** Talk about the undergraduate students.

**SEVCIK:** OK.

**UBIQUITY:** We'll note that you began your answer with a laugh.

**SEVCIK:** Yeah, I guess I don't know what to say. It's changed a lot, and I'm trying to think about what trends have occurred over the years. I've seen some changes in the balance between intellectual curiosity and focus on getting rich (or at least getting a job). Especially during the times when people were starting companies or receiving generous stock options, and Silicon Valley was producing billionaires by the day, a lot of students were motivated quite a lot by the money, and that's one of the reasons that they were getting into computer science -- foreseeing the prospect of a very profitable future for a good portion of them. But the dot-com collapse weeded out those people who had get-rich-quick in their minds, and now we see an undergraduate population that is more interested in the learning as opposed to the profit potential.

**UBIQUITY:** How would you advise a brand new undergraduate who showed up at Toronto and hadn't declared a major?

**SEVCIK:** Would I advise them to go in to computer science? I certainly wouldn't discourage them. I was lucky enough to get in to computer science at a very opportune time, when the field was small and when I could go whatever direction I wanted. That's no longer true, but there are still ample opportunities. However, I would encourage them as well to have a strong second major, or at least a strong minor in some other field, so that they're putting together their knowledge of computing with an application area as well.

**UBIQUITY:** Such as?

**SEVCIK:** Well, business is one obvious one, biology and chemistry are other obvious ones if you want to go the medical applications route. And, as you mentioned, we have an overwhelming amount of data coming at us from every direction these days. So an area that will be growing for many years is a combination of computer science and statistics, with the ability to go into the general area of data mining to try to find what the meaning is in these huge masses of data.

**UBIQUITY:** What has most surprised you about your experience with computers and computer science?

**SEVCIK:** That will take some thought. I guess the thing that's most surprised me is that Moore's Law has continued to hold for so many years. Even when I started in this field, I always thought that there were physical constraints on the minimal size and the maximal speed of the components we can build. But the physicists and the engineers that design computers have done an incredible job at always decreasing the price and increasing the performance each year. And from a performance evaluation point of view, and from a parallel computing point of view, that presents a great challenge to keep up with the curve.

**UBIQUITY:** What do you see then as you look out into the future of computer science?

**SEVCIK:** I think it's inevitable that either more and more knowledge from other fields will be brought into computer science, or else computer science will more or less move into the other fields. There has to be unification of the computing aspect with the other science aspects. And rather than having experts in one or the other and having them talk to each other, it would be ideal if the type of education were more integrated in all these fields.

**UBIQUITY:** But isn't that in contention with your other point -- that there's just too much to cover even in your own subspecialty?

**SEVCIK:** Yes, it is. I admit it. And I'm not sure what the solution is.

**UBIQUITY:** I can't resist asking you a mood question. It sounds silly, but what's the mood of computer science people now? Are they highly optimistic? Do they feel loved and well-funded?

**SEVCIK:** They're highly optimistic and are all very dedicated to what they do. They all have new ideas in their fields that they're eager to pursue. Computer science was unique when I started out in it because, when I joined our department, the age demographics were that almost everybody was a young assistant professor. And over the years, a lot of those same people are still around our department, but of

course we've been joined by an overwhelming number of new people at all ranks and levels. So it's no longer as young a field as it was. We actually have a lot of retirements coming up in the next five years in our department. Yet the enthusiasm has been maintained.

**UBIQUITY:** You mentioned Nortel before, and its interaction with the university; what is the current level of interaction between academia and industry?

**SEVCIK:** We've had some very good interactions with a number of companies over the years. Probably number one is Nortel. Over the years, they supported research at University of Toronto in computer science and electrical engineering to an extensive degree and we had a lot of contacts with them. Another one is IBM, which has one of its primary development labs situated in Toronto, where much of the development of the DB2 database product is done. I personally have been involved with them, and it's a great opportunity to have our students go work there for the summer, get involved in the kinds of advanced development and research issues that face people who are trying to maintain a leading database product. And the third company that's been very good is Bell Canada, which has sponsored lots of research and lots of infrastructure in our department lately. I think a lot of the faculty members feel that the opportunity to take their theoretical developments and put them into practice in an industrial context lends a lot of credibility to their work and adds to their motivation.

**UBIQUITY:** Well what should be done? What do you see as the next big vision?

**SEVCIK:** I think I have to come back to the overwhelming amount of data that we're seeing now. I used to spend a lot of time going through my files and seeing which ones I could delete to save space. But these days a lot of very enlightened people never throw away anything. They keep every bit of information that reaches them, and they just use search tools to find the right stuff at the right time. And that availability of the huge amounts of information and the challenge of making use of it in the best way is ultimately the biggest challenge that I see ahead, and it will keep a lot of people busy for a very long time.

[END]

Ken Sevcik:

<<http://www.cs.toronto.edu/DCS/People/Faculty/kcs.html>>

*Ubiquity, Volume 6, Issue 3 (February 2 - February 9, 2005)*

Source: <<http://www.acm.org/ubiquity>>