PHYS.ORG

When good enough is better: System finds sections of computer programs where accuracy can be traded for speed

May 13 2010, by Larry Hardesty

Graphic: Christine Daniloff

"It used to be that people used computers for computations where there was a single, hard, logical right answer," says Martin Rinard, a professor of computer science at MIT. "Now, the landscape is changing." When you do a Google search, for instance, the exact order of the first few results may not matter as much as getting an answer quickly. In the Internet age, when Web servers are performing computations for thousands of users at once, and sending the results across thousands of miles of optical fiber, programs that can efficiently find adequate solutions to a problem are often preferable to ones that inefficiently find the perfect solution.

Researchers in Rinard's group at the Computer Science and Artificial Intelligence Laboratory have developed a system that automatically looks through computer code for areas where a little bit of accuracy can be traded for significant increases in speed. In one set of tests, the system halved the time that it takes to encode video data for transmission over the Internet, with no noticeable effect on the video quality (see the related video). But the same approach could have advantages for any system that needs to process data in real time, such as stock traders' analytic software, or location-tracking or environmental-monitoring systems that use networks of sensors. It could also pay dividends for systems that need to look for patterns in huge masses of data, such as the recommendation engines at sites like Netflix or Amazon.

The researchers unveiled their system last week, at the International Conference on Software Engineering in South Africa. In a paper they presented there, they concentrated on a version of the system that alerts programmers to sections of their code where accuracy could be traded for speed. But the system could just as readily make that tradeoff automatically, on the fly. For instance, video-chat software running on a laptop could use the standard method of encoding data when the laptop's processor was otherwise idle. But if the processor got overburdened trying to handle several applications at once, the video-chat software could switch to the less computationally intensive version of the encoder.

Perforating videos. The top video was created using an algorithm commonly found in videoconferencing software. A new MIT system automatically modified the algorithm so that it took only half as long to produce the bottom video, with little loss of quality.

Cutting corners

The researchers’ system exploits a remarkably simple trick. A large computer program will usually feature numerous instances of what’s called a loop — a process that’s repeated over and over again. Suppose that you were writing a program that needed to find the average of a long list of numbers. To begin with, the program would simply step through the list, adding each number to a running total: That’s the loop. Then it would divide the total by the length of the list.

The CSAIL researchers who developed the new system — Rinard, research scientist Stelios Sidiroglou-Douskos and graduate students Hank Hoffmann and Sasa Misailovic — call their new technique “loop perforation” because it punches holes in loops: It simply skips every other step in the loop, or every two steps, or whatever it can get away with skipping without sacrificing too much accuracy. In the case of the program for averaging numbers, it might skip the first number on the list but add the second to the running total; skip the third but add the fourth; and so on. Since it’s adding up only half as many numbers, it takes only half as much time. But if the numbers follow a fairly normal probabilistic distribution, the answer will be close to what it would have been, anyway: The average height of 1,000 randomly selected Americans is probably not that far from the average height of 500 of them.

The researchers’ system is itself a loop: it searches through a program, perforating each loop in turn, executing the program and using standard measures to gauge the effect on performance. It then determines which loops’ perforation provides the greatest increases in speed with the smallest drop in quality.

Loopy insight

Loop perforation is so simple, and its effects so dramatic, that it may seem odd that it isn’t already in wide use. But for computer scientists, it’s very counterintuitive. “There’s a visceral sort of reaction against it,” says Hoffmann, “because people spend a lot of time thinking, ‘This is the best way to do this.’ And what we’re saying is that you can throw out half of what you thought was the best way to do that, and it still produces a reasonable answer.” Hoffmann recalls, for instance, that one of the applications on which the researchers tested their system was a machine-learning application, which learned to perform a classification task by looking for patterns in sample data. “Sasa knew a lot more than we did about what that app was doing,” Hoffmann says. When the system suggested the perforation of one particular loop, “Sasa was like, ‘This is wrong. You can’t do that.” But the application seemed to work perfectly well with the perforated loop. “It seems like the closer you are to the application, the more reluctant you are to accept these things,” Hoffman says.

“I like the simplicity of the technique and also the generality of it,” says Cristian Cadar, a lecturer in the Department of Computing at Imperial College London. “You can apply it to a wide variety of programs.” But, Cadar adds, “the main impediment to adoption of this technique, at least in the automatic form, is that developers are reluctant to adopt a technique where they don’t exactly understand what it does to the program.”

To allay developers’ fears, Rinard and his group are working to explain the technique’s success with greater mathematical rigor. And since loop perforation is such uncharted territory, Rinard suggests, its theoretical exploration could have totally unforeseen consequences. “You never know what’s going to happen when you understand something,” he says. “The great aspect of science is these unpredictable, unanticipated breakthroughs that come just because you’re curious.”

Provided by Massachusetts Institute of Technology

Citation: When good enough is better: System finds sections of computer programs where accuracy can be traded for speed (2010, May 13) retrieved 23 May 2024 from https://phys.org/news/2010-05-good-sections-accuracy.html