Defibrillator for stalled software

3 August 2011

Graphic: Christine Daniloff

It's happened to everyone: You're using a familiar piece of software to do something you've done a thousand times before - say, find a particular word in a document - and all of a sudden the program just stops working. You click the cursor and move the mouse, but nothing changes on-screen, and finally you just quit the program, losing whatever work you'd done since the last time you saved.

Often, a stalled program has gone into what computer scientists call an infinite loop, where it keeps executing a single block of code over and over. At the 25th European Conference on Object-Oriented Programming in Lancaster, England, in July, researchers from MIT's Computer Science and Artificial Intelligence Laboratory (CSAIL) presented a new tool that automatically interrupts infinite loops and moves on to the next line of code in the computer program. In tests, their system restored five different programs to stable enough states that data could be saved and the programs exited safely; in the majority of cases, the programs also provided at least a partial solution to the computations they were trying to perform when they got hung up.

Loops are among the most basic building blocks of computer programs. They allow a programmer to specify, in a single step, a procedure that has to be performed on many pieces of data in sequence. For instance, the search function in a word processor might have to look at thousands of individual letters in even a fairly short document, comparing each of them to the letters in a search term. If it doesn't find a match, it will move onto the next letter and "loop" back to re-execute the code that does the comparing.

Wheel spinning

A commercial program might contain tens of thousands of loops, and a slight error in the code for any one of them could lead to an infinite loop, in which the computer doesn't know when to stop repeating the same operation. Computer science professor Martin Rinard and his graduate students Michael Carbin, Sasa Misailovic and Michael Kling developed a software tool that they call Jolt, which recognizes infinite loops by monitoring the program's use of memory. A computer user who's worried that his or her computer has entered an infinite loop could activate Jolt, which would take a series of "snapshots" of the computer's memory after each iteration of a loop.

"The snapshots could be completely different," explains Carbin, who is first author on the paper. "That can be an indicator that your program is computing. It may be doing something useful for you, so maybe you don't want to break out of this. But if it's not, and it has exactly the same state, then clearly it's stuck in an infinite loop."

Jolt works in conjunction with a compiler, a program that translates code written in a high-level programming language into rudimentary instructions that a computer can understand. When

an application is being compiled, Jolt marks the beginnings and ends of all the loops indicated in the source code. If the compiled application later stalls, Jolt simply forces it to skip ahead to the first instruction following the loop it's stuck in.

Keeping tabs on all the loops in a program, however, causes it to run 7 or 8 percent slower, Carbin says. And getting commercial software developers to use Jolt when compiling their code could be a tall order. So the CSAIL researchers are working on a version of Jolt that operates directly on compiled applications, whose instructions consist entirely of fixed-length sequences of binary numbers. This binary version of Jolt, the researchers explain, will be called Bolt.

Course correction

Bolt uses the same infinite-loop detection mechanism that Jolt did, and in early tests, it seems to work well with binary files. The steeper challenge is determining what instruction to jump to once a loop has been identified. A function written in a high-level programming language might invoke other functions, which could invoke still other functions. But at the binary level, those nested function calls just look like a long list of numbers. Figuring out where one function ends and another begins is no easy task.

But Kling has developed a clever algorithm that can identify the highest-level function in operation at a given time - the one that has initiated all the others - which could help Bolt orient itself. And even if the system can't make an informed decision, Rinard explains, it could just start hopping around to new instructions at random, until it finds one that breaks the impasse.

Randomly modifying code on the fly may sound antithetical to the whole idea of computer engineering, but it's an approach that's paid off for Rinard's group, in research on websites that can weather malicious attacks and software that adjusts to changing hardware conditions, among other things.

"The vast majority of software engineering or programming-language researchers are shackled to this notion of full correctness or full soundness: You can't change anything in the program if there's even the possibility of getting a slightly wrong answer," says Westley Weimer, an assistant professor of computer science at the University of Virginia. "One of the things that's really characterized Martin's research over the last 10 or 15 years is casting off the shackles of soundness in favor of approaches that are probably correct but dramatically more useful in real life."

Weimer acknowledges that for Bolt, determining what instruction to jump to is a "difficult" problem, and because of some fundamental results in computer science, "I know for a fact that he'll never be able to get it exactly right." But, Weimer says, "The vast majority of infinite loops he will be able to figure out." There might be a few instances where Bolt will guess wrong, Weimer says. "But the comparison - and this is important for this kind of work - is that if he doesn't jump, you're stuck in an infinite loop. The comparison is not that it was working fine and he messed it up."

This story is republished courtesy of MIT News (web.mit.edu/newsoffice/), a popular site that covers news about MIT research, innovation and teaching.