

Conquering the chaos in modern, multiprocessor computers

March 10 2010, by Hannah Hickey

(PhysOrg.com) -- Computers should not play dice. That, to paraphrase Einstein, is the feeling of a University of Washington computer scientist with a simple manifesto: If you enter the same computer command, you should get back the same result. Unfortunately, that is far from the case with many of today's machines. Beneath their smooth exteriors, modern computers behave in wildly unpredictable ways, said Luis Ceze, a UW assistant professor of computer science and engineering.

"With older, single-processor systems, computers behave exactly the same way as long as you give the same commands. Today's computers are non-deterministic. Even if you give the same set of commands, you might get a different result," Ceze said.

He and UW associate professors of computer science and engineering Mark Oskin and Dan Grossman and UW graduate students Owen Anderson, Tom Bergan, Joseph Devietti, Brandon Lucia and Nick Hunt have developed a way to get modern, multiple-processor computers to behave in predictable ways, by automatically parceling sets of commands and assigning them to specific places. Sets of commands get calculated simultaneously, so the well-behaved program still runs faster than it would on a single processor.

Next week at the International Conference on Architectural Support for Programming Languages and Operating Systems in Pittsburgh, Bergan will present a software-based version of this system that could be used on existing machines. It builds on a more general approach the group

published last year, which was recently chosen as a top paper for 2009 by the Institute of Electrical and Electronics Engineers' journal [Micro](#).

In the old days one computer had one processor. But today's consumer standard is dual-core processors, and even quad-core machines are appearing on store shelves. Supercomputers and servers can house hundreds, even thousands, of processing units.

On the plus side, this design creates computers that run faster, cost less and use less power for the same performance delivered on a single [processor](#). On the other hand, multiple processors are responsible for elusive errors that freeze Web browsers and crash programs.

It is not so different from the classic chaos problem in which a butterfly flaps its wings in one place and can cause a hurricane across the globe. Modern shared-memory computers have to shuffle tasks from one place to another. The speed at which the information travels can be affected by tiny changes, such as the distance between parts in the [computer](#) or even the temperature of the wires. Information can thus arrive in a different order and lead to unexpected errors, even for tasks that ran smoothly hundreds of times before.

"With multi-core systems the trend is to have more bugs because it's harder to write code for them," Ceze said. "And these concurrency bugs are much harder to get a handle on."

One application of the UW system is to make errors reproducible, so that programs can be properly tested.

"We've developed a basic technique that could be used in a range of systems, from cell phones to data centers," Ceze said. "Ultimately, I want to make it really easy for people to design high-performing, low-energy and secure systems."

Last year Ceze, Oskin, and Peter Godman, a former director at Isilon Systems, founded a company to commercialize their technology. Petra is initially named after the Greek word for rock because it hopes to develop "rock-solid systems," Ceze said. The Seattle-based startup will soon release its first product, Jinx, which makes any errors that are going to crop up in a program happen quickly.

"We can compress the effect of thousands of people using a program into a few minutes during the software's development," Ceze said. "We want to allow people to write code for multi-core systems without going insane."

The company already has some big-name clients trying its product, Ceze said, though it is not yet disclosing their identities.

"If this erratic behavior irritates us, as software users, imagine how it is for banks or other mission-critical applications."

More information:

<http://www.ece.cmu.edu/CALCM/asplos10/doku.php>

Provided by University of Washington

Citation: Conquering the chaos in modern, multiprocessor computers (2010, March 10) retrieved 26 April 2024 from

<https://phys.org/news/2010-03-conquering-chaos-modern-multiprocessor.html>

This document is subject to copyright. Apart from any fair dealing for the purpose of private study or research, no part may be reproduced without the written permission. The content is provided for information purposes only.