

Code green: Energy-efficient programming to curb computers' power use

May 31 2011

Soaring energy consumption by ever more powerful computers, data centers and mobile devices has many experts looking to reduce the energy use of these devices. Most projects so far focus on more efficient cooling systems or energy-saving power modes.

A University of Washington project sees a role for programmers to reduce the energy appetite of the ones and zeroes in the code itself. Researchers have created a system, called EnergJ, that reduces [energy consumption](#) in simulations by up to 50 percent, and has the potential to cut energy by as much as 90 percent. They will present the research next week in San Jose at the Programming Language Design and Implementation annual meeting.

"We all know that energy consumption is a big problem," said author Luis Ceze, a UW assistant professor of computer science and engineering. "With our system, mobile phone users would notice either a smaller phone, or a longer [battery life](#), or both. Computing centers would notice a lower energy bill."

The basic idea is to take advantage of processes that can survive tiny errors that happen when, say, voltage is decreased or correctness checks are relaxed. Some examples of possible applications are streaming audio and video, games and real-time image recognition for augmented-reality applications on [mobile devices](#).

"[Image recognition](#) already needs to be tolerant of little problems, like a

speck of dust on the screen," said co-author Adrian Sampson, a UW doctoral student in computer science and engineering. "If we introduce a few more dots on the image because of errors, the algorithm should still work correctly, and we can save energy."

The UW system is a general framework that creates two interlocking pieces of code. One is the precise part – for instance, the encryption on your bank account's password. The other portion is for all the processes that could survive occasional slipups.

The software creates an impenetrable barrier between the two pieces.

"We make it impossible to leak data from the approximate part into the precise part," Sampson said. "You're completely guaranteed that can't happen."

While computers' [energy use](#) is frustrating and expensive, there is also a more fundamental issue at stake. Some experts believe we are approaching a limit on the number of transistors that can run on a single microchip. The so-called "dark silicon problem" says that as we boost computer speeds by cramming more transistors onto each chip, there may no longer be any way to supply enough power to the chip to run all the transistors.

The UW team's approach would work like a dimmer switch, letting some transistors run at a lower voltage. Approximate tasks could run on the dimmer regions of the chip.

"When I started thinking about this, it became more and more obvious that this could be applied, at least a little bit, to almost everything," Sampson said. "It seemed like I was always finding new places where it could be applied, at least in a limited way."

Researchers would use the program with a new type of hardware where some transistors have a lower voltage, the force on electrons in the circuit. This slightly increases the risk of random errors; EnerJ shuttles only approximate tasks to these transistors.

"If you can afford one error every 100,000 operations or so, you can already save a lot of energy," Ceze said.

Other ways to use hardware to save energy are lowering the refresh rate and reducing voltage of the memory chip.

Simulations of such hardware show that running EnerJ would cut energy by about 20 to 25 percent, on average, depending on the aggressiveness of the approach. For one program the energy saved was almost 50 percent. Researchers are now designing hardware to test their results in the lab.

Today's computers could also use EnerJ with a purely software-based approach. For example, the computer could round off numbers or skip some extra accuracy checks on the approximate part of the code to save [energy](#) – researchers estimate between 30 and 50 percent savings based on software alone.

Combining the software and hardware methods they believe they could cut power use by about 90 percent.

"Our long-term goal would be 10 times improvement in battery life," Ceze said. "I don't think it is totally out of the question to have an order of magnitude reduction if we continue squeezing unnecessary accuracy."

The [program](#) is called EnerJ because it is an extension for the Java programming language. The team hopes to release the code as an open-source tool this summer.

Provided by University of Washington

Citation: Code green: Energy-efficient programming to curb computers' power use (2011, May 31) retrieved 29 April 2024 from <https://phys.org/news/2011-05-code-green-energy-efficient-curb-power.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.