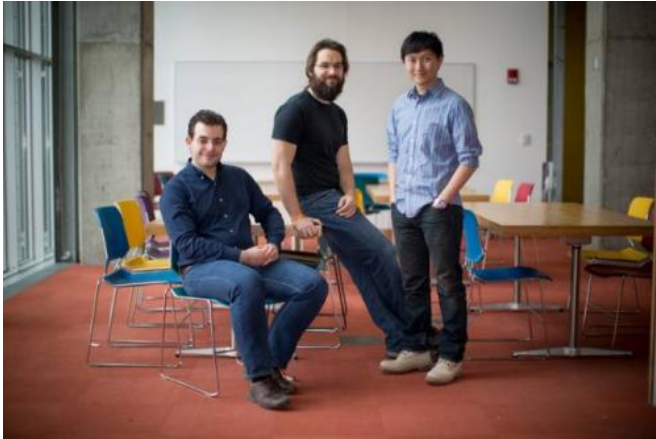


New approach to distributing computations could make multicore chips much faster

18 February 2015, by Larry Hardesty



(From left) Daniel Sanchez, Nathan Beckmann, and Po-An Tsai. Credit: Bryce Vickmark

Computer chips' clocks have stopped getting faster. To keep delivering performance improvements, chipmakers are instead giving chips more processing units, or cores, which can execute computations in parallel.

But the ways in which a chip carves up computations can make a big difference to performance. In a 2013 paper, Daniel Sanchez, the TIBCO Founders Assistant Professor in MIT's Department of Electrical Engineering and Computer Science, and his student, Nathan Beckmann, described a system that cleverly distributes data around multicore chips' memory banks, improving execution times by 18 percent on average while actually increasing energy efficiency.

This month, at the Institute of Electrical and Electronics Engineers' International Symposium on High-Performance Computer Architecture, members of Sanchez's group have been nominated for a best-paper award for an extension of the system that controls the distribution of not

only data but computations as well. In simulations involving a 64-core chip, the system increased computational speeds by 46 percent while reducing power consumption by 36 percent.

"Now that the way to improve performance is to add more cores and move to larger-scale parallel systems, we've really seen that the key bottleneck is communication and memory accesses," Sanchez says. "A large part of what we did in the previous project was to place data close to [computation](#). But what we've seen is that how you place that computation has a significant effect on how well you can place data nearby."

Disentanglement

The problem of jointly allocating computations and data is very similar to one of the canonical problems in chip design, known as "place and route." The place-and-route problem begins with the specification of a set of logic circuits, and the goal is to arrange them on the chip so as to minimize the distances between circuit elements that work in concert.

This problem is what's known as NP-hard, meaning that as far as anyone knows, for even moderately sized chips, all the computers in the world couldn't find the optimal solution in the lifetime of the universe. But chipmakers have developed a number of algorithms that, while not absolutely optimal, seem to work well in practice.

Adapted to the problem of allocating computations and data in a 64-core chip, these algorithms will arrive at a solution in the space of several hours. Sanchez, Beckmann, and Po-An Tsai, another student in Sanchez's group, developed their own algorithm, which finds a solution that is more than 99 percent as efficient as that produced by standard place-and-route algorithms. But it does so in milliseconds.

"What we do is we first place the data roughly," Sanchez says. "You spread the data around in such a way that you don't have a lot of [memory] banks overcommitted or all the data in a region of the chip. Then you figure out how to place the [computational] threads so that they're close to the data, and then you refine the placement of the data given the placement of the threads. By doing that three-step solution, you disentangle the problem."

In principle, Beckmann adds, that process could be repeated, with computations again reallocated to accommodate data placement and vice versa. "But we achieved 1 percent, so we stopped," he says. "That's what it came down to, really."

Keeping tabs

The MIT researchers' system monitors the chip's behavior and reallocates data and threads every 25 milliseconds. That sounds fast, but it's enough time for a computer chip to perform 50 million operations.

During that span, the monitor randomly samples the requests that different cores are sending to memory, and it stores the requested memory locations, in an abbreviated form, in its own memory circuit.

Every core on a chip has its own cache—a local, high-speed memory bank where it stores frequently used data. On the basis of its samples, the monitor estimates how much cache space each core will require, and it tracks which cores are accessing which [data](#).

The monitor does take up about 1 percent of the [chip](#)'s area, which could otherwise be allocated to additional computational circuits. But Sanchez believes that chipmakers would consider that a small price to pay for significant [performance improvements](#).

"There was a big National Academy study and a DARPA-sponsored [information science and technology] study on the importance of communication dominating computation," says David Wood, a professor of computer science at the University of Wisconsin at Madison. "What you

can see in some of these studies is that there is an order of magnitude more energy consumed moving operands around to the computation than in the actual computation itself. In some cases, it's two orders of magnitude. What that means is that you need to not do that."

The MIT researchers "have a proposal that appears to work on practical problems and can get some pretty spectacular results," Wood says. "It's an important problem, and the results look very promising."

More information: "Scaling Distributed Cache Hierarchies through Computation and Data Co-Scheduling": people.csail.mit.edu/sanchez/papers/2015.cdcs.hpca.pdf

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

Provided by Massachusetts Institute of Technology

APA citation: New approach to distributing computations could make multicore chips much faster (2015, February 18) retrieved 14 October 2019 from <https://phys.org/news/2015-02-approach-multicore-chips-faster.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.