

New version of breakthrough memory management scheme better accommodates commercial chips

September 21 2016, by Larry Hardesty



MIT researchers have found a new way of managing memory on computer chips that uses circuit space much more efficiently and is more consistent with existing chip designs. Credit: Jose-Luis Olivares/MIT

A year ago, researchers from MIT's Computer Science and Artificial Intelligence Laboratory unveiled a fundamentally new way of managing memory on computer chips, one that would use circuit space much more efficiently as chips continue to comprise more and more cores, or processing units. In chips with hundreds of cores, the researchers' scheme could free up somewhere between 15 and 25 percent of on-chip memory, enabling much more efficient computation.

Their scheme, however, assumed a certain type of computational behavior that most modern chips do not, in fact, enforce. Last week, at the International Conference on Parallel Architectures and Compilation Techniques—the same conference where they first reported their scheme—the researchers presented an updated version that's more consistent with existing [chip](#) designs and has a few additional improvements.

The essential challenge posed by multicore chips is that they execute instructions in parallel, while in a traditional computer program, instructions are written in sequence. Computer scientists are constantly working on ways to make parallelization easier for computer programmers.

The initial version of the MIT researchers' scheme, called Tardis, enforced a standard called sequential consistency. Suppose that different parts of a program contain the sequences of instructions ABC and XYZ. When the program is parallelized, A, B, and C get assigned to core 1; X, Y, and Z to core 2.

Sequential consistency doesn't enforce any relationship between the relative execution times of instructions assigned to different cores. It doesn't guarantee that core 2 will complete its first instruction—X—before core 1 moves onto its second—B. It doesn't even guarantee that core 2 will begin executing its first

instruction—X—before core 1 completes its last one—C. All it guarantees is that, on core 1, A will execute before B and B before C; and on core 2, X will execute before Y and Y before Z.

The first author on the new paper is Xiangyao Yu, a graduate student in electrical engineering and computer science. He is joined by his thesis advisor and co-author on the earlier paper, Srinivas Devadas, the Edwin Sibley Webster Professor in MIT's Department of Electrical Engineering and Computer Science, and by Hongzhe Liu of Algonquin Regional High School and Ethan Zou of Lexington High School, who joined the project through MIT's Program for Research in Mathematics, Engineering and Science (PRIMES) program.

Planned disorder

But with respect to reading and writing data—the only type of operations that a memory-management scheme like Tardis is concerned with—most modern chips don't enforce even this relatively modest constraint. A standard chip from Intel might, for instance, assign the sequence of read/write instructions ABC to a core but let it execute in the order ACB.

Relaxing standards of consistency allows chips to run faster. "Let's say that a core performs a write operation, and the next instruction is a read," Yu says. "Under sequential consistency, I have to wait for the write to finish. If I don't find the data in my cache [the small local memory bank in which a core stores frequently used data], I have to go to the central place that manages the ownership of data."

"This may take a lot of messages on the network," he continues. "And depending on whether another core is holding the data, you might need to contact that core. But what about the following read? That instruction is sitting there, and it cannot be processed. If you allow this reordering,

then while this write is outstanding, I can read the next instruction. And you may have a lot of such instructions, and all of them can be executed."

Tardis uses chip space more efficiently than existing memory management schemes because it coordinates cores' memory operations according to "logical time" rather than chronological time. With Tardis, every data item in a shared memory bank has its own time stamp. Each core also has a counter that effectively time stamps the operations it performs. No two cores' counters need agree, and any given core can keep churning away on data that has since been updated in main memory, provided that the other cores treat its computations as having happened earlier in time.

Division of labor

To enable Tardis to accommodate more relaxed consistency standards, Yu and his co-authors simply gave each core two counters, one for read operations and one for write operations. If the core chooses to execute a read before the preceding write is complete, it simply gives it a lower time stamp, and the chip as a whole knows how to interpret the sequence of events.

Different chip manufacturers have different consistency rules, and much of the new paper describes how to coordinate counters, both within a single core and among cores, to enforce those rules. "Because we have time stamps, that makes it very easy to support different consistency models," Yu says. "Traditionally, when you don't have the time stamp, then you need to argue about which event happens first in physical time, and that's a little bit tricky."

"The new work is important because it's directly related to the most popular relaxed-consistency model that's in current Intel chips," says

Larry Rudolph, a vice president and senior researcher at Two Sigma, a hedge fund that uses artificial-intelligence and distributed-computing techniques to devise trading strategies. "There were many, many different consistency models explored by Sun Microsystems and other companies, most of which are now out of business. Now it's all Intel. So matching the consistency model that's popular for the current Intel chips is incredibly important."

As someone who works with an extensive distributed-computing system, Rudolph believes that Tardis' greatest appeal is that it offers a unified framework for managing memory at the [core](#) level, at the level of the computer network, and at the levels in between. "Today, we have caching in microprocessors, we have the DRAM [dynamic random-access memory] model, and then we have storage, which used to be disk drive," he says. "So there was a factor of maybe 100 between the time it takes to do a cache access and DRAM access, and then a factor of 10,000 or more to get to disk. With flash [memory] and the new nonvolatile RAMs coming out, there's going to be a whole hierarchy that's much nicer. What's really exciting is that Tardis potentially is a model that will span consistency between processors, storage, and distributed file systems."

More information: Tardis 2.0: Optimized Time Traveling Coherence for Relaxed Consistency Models. [dx.doi.org/10.1145/2967938.2967942](https://doi.org/10.1145/2967938.2967942)

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

Citation: New version of breakthrough memory management scheme better accommodates

commercial chips (2016, September 21) retrieved 21 June 2024 from
<https://phys.org/news/2016-09-version-breakthrough-memory-scheme-accommodates.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.