MIT News
ON CAMPUS AND AROUND THE WORLD

A novel technique developed by MIT researchers compresses “objects” in memory for the first time, freeing up more memory used by computers, allowing them to run faster and perform more tasks simultaneously.

Image: Christine Daniloff, MIT

A novel data-compression technique for faster computer programs

Researchers free up more bandwidth by compressing “objects” within the memory hierarchy.

Rob Matheson | MIT News Office
April 16, 2019

A novel technique developed by MIT researchers rethinks hardware data compression to free up more memory used by computers and mobile devices, allowing them to run faster and perform more tasks simultaneously.

Data compression leverages redundant data to free up storage capacity, boost computing speeds, and provide other perks. In current computer systems, accessing main memory is very expensive compared to actual computation. Because of this, using data compression in the memory helps improve performance, as it reduces the frequency and amount of data programs need to fetch from main memory.

Memory in modern computers manages and transfers data in fixed-size chunks, on which traditional compression techniques must operate. Software, however, doesn’t naturally store its data in fixed-size chunks. Instead, it uses “objects,” data structures that contain various types of data and have variable sizes. Therefore, traditional hardware compression techniques handle objects poorly.

In a paper being presented at the ACM International Conference on Architectural Support for Programming Languages and Operating Systems this week, the MIT researchers describe the

RELATED

Paper: “Compress Objects, Not Cache Lines: An Object-Based Compressed Memory Hierarchy”

Po-An Tsai

Daniel Sanchez

Computer Science and Artificial Intelligence Laboratory

Department of Electrical Engineering and Computer Science

School of Engineering

first approach to compress objects across the memory hierarchy. This reduces memory usage while improving performance and efficiency.

Programmers could benefit from this technique when programming in any modern programming language — such as Java, Python, and Go — that stores and manages data in objects, without changing their code. On their end, consumers would see computers that can run much faster or can run many more apps at the same speeds. Because each application consumes less memory, it runs faster, so a device can support more applications within its allotted memory.

In experiments using a modified Java virtual machine, the technique compressed twice as much data and reduced memory usage by half over traditional cache-based methods.

“The motivation was trying to come up with a new memory hierarchy that could do object-based compression, instead of cache-line compression, because that’s how most modern programming languages manage data,” says first author Po-An Tsai, a graduate student in the Computer Science and Artificial Intelligence Laboratory (CSAIL).

“All computer systems would benefit from this,” adds co-author Daniel Sanchez, a professor of computer science and electrical engineering, and a researcher at CSAIL. “Programs become faster because they stop being bottlenecked by memory bandwidth.”

The researchers built on their prior work that restructures the memory architecture to directly manipulate objects. Traditional architectures store data in blocks in a hierarchy of progressively larger and slower memories, called “caches.” Recently accessed blocks rise to the smaller, faster caches, while older blocks are moved to slower and larger caches, eventually ending back in main memory. While this organization is flexible, it is costly: To access memory, each cache needs to search for the address among its contents.

“Because the natural unit of data management in modern programming languages is objects, why not just make a memory hierarchy that deals with objects?” Sanchez says.

In a paper published last October, the researchers detailed a system called Hotpads, that stores entire objects, tightly packed into hierarchical levels, or “pads.” These levels reside entirely on efficient, on-chip, directly addressed memories — with no sophisticated searches required.

Programs then directly reference the location of all objects across the hierarchy of pads. Newly allocated and recently referenced objects, and the objects they point to, stay in the faster level. When the faster level fills, it runs an “eviction” process that keeps recently referenced objects but kicks down older objects to slower levels and recycles objects that are no longer useful, to free up space. Pointers are then updated in each object to point to the new locations of all moved objects. In this way, programs can access objects much more cheaply than searching through cache levels.

For their new work, the researchers designed a technique, called “Zippads,” that leverages the Hotpads architecture to compress objects. When objects first start at the faster level, they’re uncompressed. But when they’re evicted to slower levels, they’re all compressed. Pointers in all objects across levels then point to those compressed objects, which makes them easy to recall back to the faster levels and able to be stored more compactly than prior techniques.

A compression algorithm then leverages redundancy across objects efficiently. This technique uncovers more compression opportunities than previous techniques, which were limited to finding redundancy within each fixed-size block. The algorithm first picks a few representative objects as “base” objects. Then, in new objects, it only stores the different data between those objects and the representative base objects.

Brandon Lucia, an assistant professor of electrical and computer engineering at Carnegie Mellon University, praises the work for leveraging features of object-oriented programming languages to better compress memory. “Abstractions like object-oriented programming are added to a system to make programming simpler, but often introduce a cost in the performance or efficiency of the system,” he says. “The interesting thing about this work is that it uses the existing object abstraction as a way of making memory compression more effective, in turn making the system faster and more efficient with novel computer architecture features.”

Topics: Research Computer science and technology Algorithms

Computer Science and Artificial Intelligence Laboratory (CSAIL)

Electrical Engineering & Computer Science (eecs) School of Engineering

---