System better allocates time-sensitive data processing across cores to maintain quick user-response times

22 February 2019, by Rob Matheson

A new system by MIT researchers improves the efficiency of high-speed operations in data centers by better assigning time-sensitive data processing across CPU cores and ensuring hardware runs productively. Credit: Massachusetts Institute of Technology

Today's data centers eat up and waste a good amount of energy responding to user requests as fast as possible, with only a few microseconds delay. A new system by MIT researchers improves the efficiency of high-speed operations by better assigning time-sensitive data processing across central processing unit (CPU) cores and ensuring hardware runs productively.

Data centers operate as distributed networks, with numerous web and mobile applications implemented on a single server. When users send requests to an app, bits of stored data are pulled from hundreds or thousands of services across as many servers. Before sending a response, the app must wait for the slowest service to process the data. This lag time is known as tail latency.

Current methods to reduce tail latencies leave tons of CPU cores in a server open to quickly handle incoming requests. But this means that cores sit idly for much of the time, while servers continue using energy just to stay powered on. Data centers can contain hundreds of thousands of servers, so even small improvements in each server's efficiency can save millions of dollars.

Alternatively, some systems reallocate cores across apps based on workload. But this occurs over milliseconds—around one-thousandth the desired speed for today's fast-paced requests. Waiting too long can also degrade an app's performance, because any information that's not processed before an allotted time doesn't get sent to the user.

In a paper being presented at the USENIX Networked Systems Design and Implementation conference next week, the researchers developed a faster core-allocating system, called Shenango, that reduces tail latencies, while achieving high efficiencies. First, a novel algorithm detects which apps are struggling to process data. Then, a software component allocates idle cores to handle the app's workload.

"In data centers, there's a tradeoff between efficiency and latency, and you really need to reallocate cores at much finer granularity than every millisecond," says first author Amy Ousterhout, a Ph.D. student in the Computer Science and Artificial Intelligence Laboratory (CSAIL). Shenango lets servers "manage operations that occur at really short time scales and do so efficiently."

Energy and cost savings will vary by data center, depending on size and workloads. But the overall aim is to improve data center CPU utilization, so that every core is put to good use. The best CPU

utilization rates today sit at about 60 percent, but the researchers say their system could potentially boost that figure to 100 percent.

"Data center utilization today is quite low," says co-author Adam Belay, an assistant professor of electrical engineering and computer science and a CSAIL researcher. "This is a very serious problem [that can't] be solved in a single place in the data center. But this system is one critical piece in driving utilization up higher."

Joining Ousterhout and Belay on the paper are Hari Balakrishnan, the Fujitsu Chair Professor in the Department of Electrical Engineering and Computer Science, and CSAIL Ph.D. students Jonathan Behrens and Joshua Fried.

Efficient congestion-detection

In a real-world data center, Shenango—algorithm and software—would run on each server in a data center. All the servers would be able to communicate with each other.

The system's first innovation is a novel congestion-detection algorithm. Every five microseconds the algorithm checks data packets queued for processing for each app. If a packet is still waiting from the last observation, the algorithm notes there's at least a 5-microsecond delay. It also checks if any computation processes, called threads, are waiting to be executed. If so, the system considers that a "congested" app.

It seems simple enough. But the queue's structure is important to achieving microsecond-scale congestion detection. Traditional thinking meant having the software check the timestamp of each queued-up data packet, which would take too much time.

The researchers implement the queues in efficient structures known as "ring buffers." These structures can be visualized as different slots around a ring. The first inputted data packet goes into a starting slot. As new data arrive, they're dropped into subsequent slots around the ring. Usually, these structures are used for first-in-first-out data processing, pulling data from the starting slot and working toward the ending slot.

The researchers' system, however, only stores data packets briefly in the structures, until an app can process them. In the meantime, the stored packets can be used for congestion checks. The algorithm need only compare two points in the queue—the location of the first packet and where the last packet was five microseconds ago—to determine if packets are encountering a delay.

"You can look at these two points, and track their progress every five microseconds, to see how much data has been processed," Fried says. Because the structures are simple, "you only have to do this once per core. If you're looking at 24 cores, you do 24 checks in five microseconds, which scales nicely."

Smart allocation

The second innovation is called the IOKernel, the central software hub that steers data packets to appropriate apps. The IOKernel also uses the congestion detection algorithm to quickly allocate cores to congested apps orders of magnitude more quickly than traditional approaches.

For instance, the IOKernel may see an incoming data packet for a certain app that requires microsecond processing speeds. If the app is congested due to a lack of cores, the IOKernel immediately devotes an idle core to the app. If it also sees another app running cores with less time-sensitive data, it will grab some of those cores and reallocate them to the congested app. The apps themselves also help out: If an app isn't processing data, it alerts the IOKernel that its cores can be reallocated. Processed data goes back to the IOKernel to send the response.

"The IOKernel is concentrating on which apps need cores that don't have them," Behrens says. "It's trying to figure out who's overloaded and needs more cores, and gives them cores as quickly as possible, so they don't fall behind and have huge latencies."

The tight communication between the IOKernel, algorithm, apps, and server hardware is "unique in

data centers" and allows Shenango to function seamlessly, Belay says: "The system has global visibility into what's happening in each server. It sees the hardware providing the packets, what's running where in each core, and how busy each of the apps are. And it does that at the microsecond scale."

Next, the researchers are refining Shenango for real-world data center implementation. To do so, they're ensuring the software can handle a very high data throughput and has appropriate security features.

Provided by Massachusetts Institute of Technology