Changing the rules of computing could lighten Big Data's impact on the internet

25 March 2020, by Gabe Cherry

Credit: CC0 Public Domain

At a time when we're relying on the internet to an unprecedented degree in our daily lives, a team of U-M researchers led by Mosharaf Chowdhury and Harsha Madhyastha has found a way for tech companies, banks and health systems to squeeze more capacity out of our existing infrastructure.

A change to the design of big-data software tool Apache Spark could enable the world's biggest users of computing power to crunch through massive tasks up to 16 times faster while lightening their burden on the internet. Chowdhury is an assistant professor and Madhyastha is an associate professor, both of computer science and engineering. The modification, called Sol, is now available for download on GitHub.

Spark is an open-source electronic framework that serves as a task manager, coordinating vast networks of individual computers to work together as a single machine on big computing tasks. One of the most widely-used tools of its kind in the world, it's used by every major tech company as well as banks, telecommunications companies, governments and many others.

When Spark was built a decade ago, most of this work took place at large data centers, where vast banks of machines were located at a single site. But today, it's increasingly being used to connect machines that are spread across the globe and connected by the internet.

Chowdhury helped build Spark during his time as a graduate student at the University of California Berkeley. He explains that it parcels out work to individual machines using a component called an execution engine. It was designed primarily for large data centers, where groups of machines on the same local network could communicate quickly with each other. But it's less efficient when machines are thousands of miles apart, connected by the relatively narrow pipe of the internet.

"Spark's existing execution engine makes decisions about where to send work at the very last minute—only after the CPU signals that it's ready for more work does it send a new task," Chowdhury said. "That approach maximizes flexibility, and it makes sense when a task is housed in a single data center. But that communication takes much longer between machines that are connected by the internet. The last-minute approach often leaves CPUs underutilized, meaning they're sitting around waiting for work."

So Chowdhury and Madhyastha, working with graduate student research assistants Fan Lai and Jie You as well as undergraduate student Xiangfeng Zhu, wrote a new execution engine called Sol. Sol takes a more proactive approach; instead of waiting for CPUs to signal that they're ready for a new job, it guesses which ones will be next in line and actively pushes new tasks to them. It also instructs machines to process data locally when possible instead of constantly moving it between machines.

This means less shuffling of data and commands between machines, lessening the burden on the

internet and speeding data processing.
Chowdhury's team has found that it speeds computation dramatically, making common tasks four to 16 times faster.

While the currently available release is a research version of the software rather than a more polished product, Chowdhury says releasing it in its current form is a way to drive research at a time when speed is essential.

"Fan Lai is already making himself available to help those who want to try it," he said. "We're doing everything we can to move quickly."

The paper is titled "Sol: Fast Distributed Computation Over Slow Networks."

More information: Sol: Fast Distributed Computation Over Slow Networks. www.usenix.org/system/files/nsdi20-paper-lai.pdf

Provided by University of Michigan