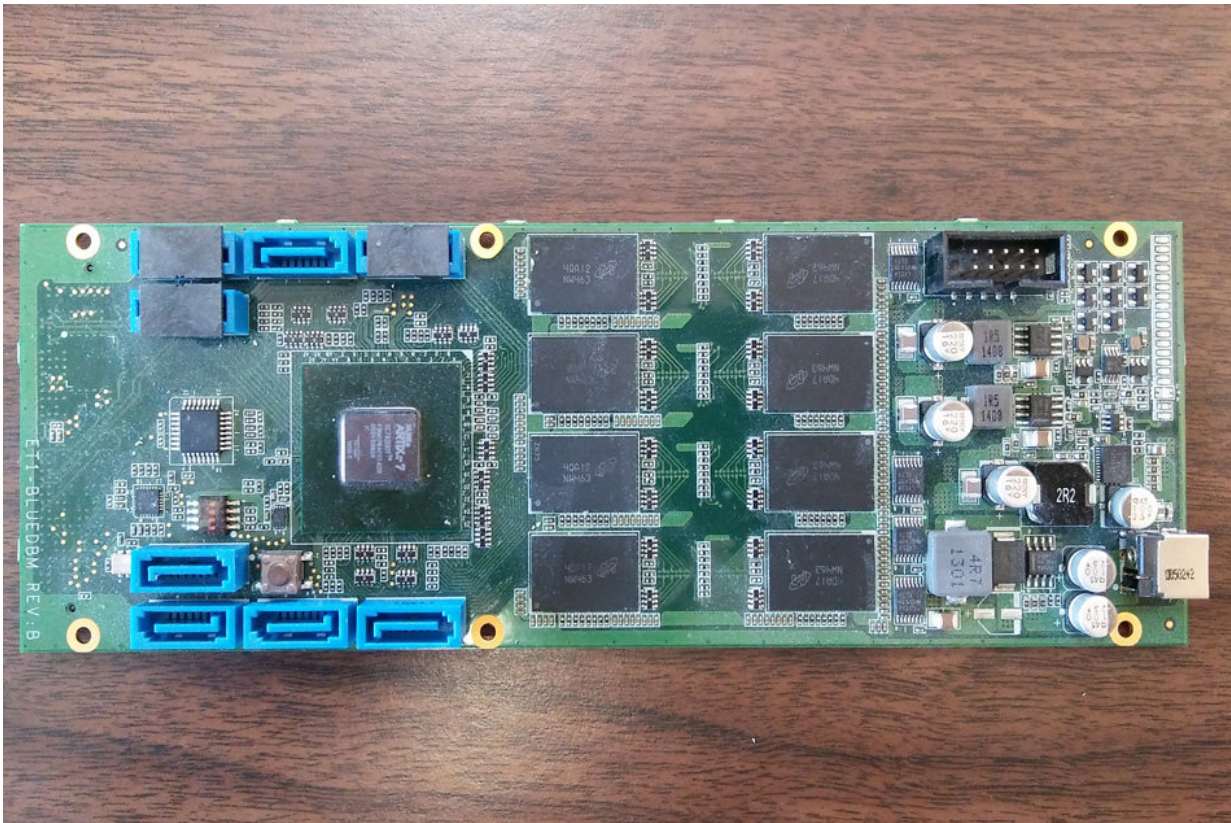# Device allows a personal computer to process huge graphs

May 31 2018, by Rob Matheson

Researchers from MIT's Computer Science and Artificial Intelligence Laboratory (CSAIL) have designed a device that helps cheap flash storage process massive graphs on a personal computer. The device (pictured here) consists of a flash chip array (eight black chips) and computation "accelerator" (square piece directly to the left of the array). A novel algorithm sorts all access requests for graph data into a sequential order that flash can access quickly and easily, while merging some requests to reduce the overhead of sorting. Credit: Massachusetts Institute of Technology

In data-science parlance, graphs are structures of nodes and connecting lines that are used to map scores of complex data relationships. Analyzing graphs is useful for a broad range of applications, such as ranking webpages, analyzing social networks for political insights, or plotting neuron structures in the brain.

Consisting of billions of nodes and lines, however, large graphs can reach terabytes in size. The graph data are typically processed in expensive dynamic random access memory (DRAM) across multiple power-hungry servers.

Researchers from MIT's Computer Science and Artificial Intelligence Laboratory (CSAIL) have now designed a device that uses cheap flash storage—the type used in smartphones—to process massive graphs using only a single personal computer.

Flash storage is typically far slower than DRAM at processing graph data. But the researchers developed a device consisting of a flash chip array and computation "accelerator," that helps flash achieves DRAM-like performance.

Powering the device is a novel algorithm that sorts all access requests for graph data into a sequential order that flash can access quickly and easily. It also merges some requests to reduce the overhead—the combined computation time, memory, bandwidth, and other computing resources—of sorting.

The researchers ran the device against several traditional high-performance systems processing several large graphs, including the massive Web Data Commons Hyperlink Graph, which has 3.5 billion nodes and 128 billion connecting lines. To process that graph, the
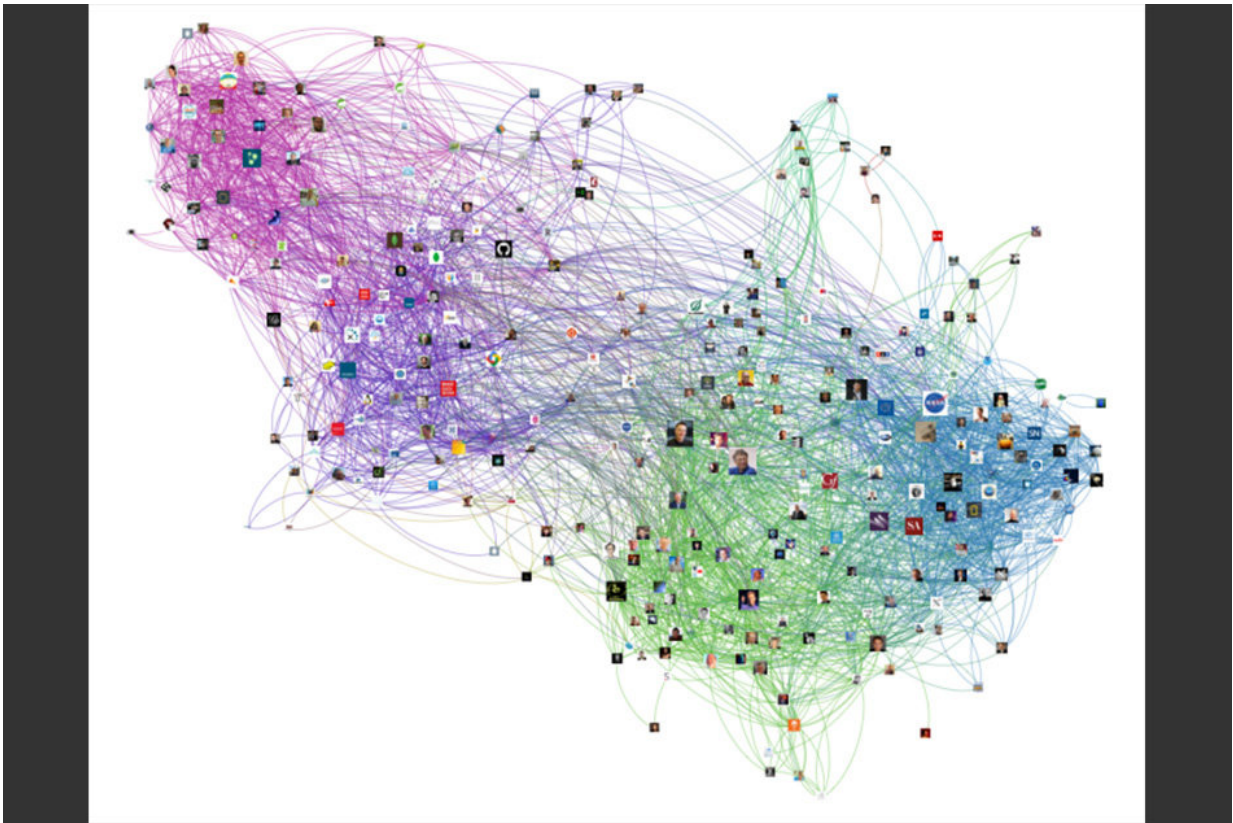
traditional systems all required a server that cost thousands of dollars and contained 128 gigabytes of DRAM. The researchers achieved the same performance by plugging two of their devices—totaling 1 gigabyte of DRAM and 1 terabyte of flash—into a desktop computer. Moreover, by combining several devices, they could process massive graphs—up to 4 billion nodes and 128 billion connecting lines—that no other system could handle on the 128-gigabyte server.

"The bottom line is that we can maintain performance with much smaller, fewer, and cooler—as in temperature and power consumption—machines," says Sang-Woo Jun, a CSAIL graduate student and first author on a paper describing the device, which is being presented at the International Symposium on Computer Architecture (ISCA).

The device could be used to cut costs and energy associated with graph analytics, and even improve performance, in a broad range of applications. The researchers, for instance, are currently creating a program that could identify genes that cause cancers. Major tech companies such as Google could also leverage the devices to reduce their energy footprint by using far fewer machines to run analytics.

"Graph processing is such a general idea," says co-author Arvind, the Johnson Professor in Computer Science Engineering. "What does page ranking have in common with gene detection? For us, it's the same computation problem—just different graphs with different meanings. The type of application someone develops will determine the impact it has on society."

Paper co-authors are CSAIL graduate student Shuotao Xu, and Andy Wright and Sizhuo Zhang, two graduate students in CSAIL and the Department of Electrical Engineering and Computer Science.

The researchers were able to process several large graphs — with up to 3.5 billion nodes and 128 billion connecting lines — by plugging two of their devices, totaling 1 gigabyte of DRAM and 1 terabyte of flash, into a desktop computer. Traditional systems all required a server that cost thousands of dollars and contained 128 gigabytes of DRAM to process the graphs. Credit: Massachusetts Institute of Technology

Sort and reduce

In graph analytics, a system will basically search for and update a node's value based on its connections with other nodes, among other metrics. In webpage ranking, for instance, each node represents a webpage. If node A has a high value and connects to node B, then node B's value will also

increase.

Traditional systems store all graph data in DRAM, which makes them fast at processing the data but also expensive and power-hungry. Some systems offload some data storage to flash, which is cheaper but slower and less efficient, so they still require substantial amounts of DRAM.

The researchers' device runs on what the researchers call a "sort-reduce" algorithm, which solves a major issue with using flash as the primary storage source: waste.

Graph analytics systems require access to nodes that may be very far from one another across a massive, sparse graph structure. Systems generally request direct access to, say, 4 to 8 bytes of data to update a node's value. DRAM provides that direct access very quickly. Flash, however, only accesses data in 4- to 8-kilobyte chunks, but still only updates a few bytes. Repeating that access for every request while jumping across the graph wastes bandwidth. "If you need to access the entire 8 kilobytes, and use only 8 bytes and toss the rest, you end up throwing 1,000 times performance away," Jun says.

The sort-reduce algorithm instead takes all direct access requests and sorts them in sequential order by identifiers, which show the destination of the request—such as grouping together all updates for node A, all for node B, and so on. Flash can then access kilobyte-sized chunks of thousands of requests at once, making it far more efficient.

To further save computation power and bandwidth, the algorithm simultaneously merges the data into the smallest groupings possible. Whenever the algorithm notes matching identifiers, it sums those into a single data packet—such as A1 and A2 becoming A3. It continues doing so, creating increasingly smaller packets of data with matching identifiers, until it produces the smallest possible packet to sort. This

drastically reduces the amount of duplicate requests to access.

Using the sort-reduce algorithm on two large graphs, the researchers reduced the total data that needed to be updated in flash by about 90 percent.

Offloading computation

The sort-reduce algorithm is computation-intensive for a host computer, however, so the researchers implemented a custom accelerator in the device. The accelerator acts as a midway point between the host and flash chips, executing all computation for the algorithm. This offloads so much power to the accelerator that the host can be a low-powered PC or laptop that manages sorted data and executes other minor tasks.

"Accelerators are supposed to help the host compute, but we've come so far [with the computations] that the host becomes unimportant," Arvind says.

"The MIT work shows a new way to perform analytics on very large graphs: Their work exploits flash memory to store the graphs and exploits 'field-programmable gate arrays' [custom integrated circuits] in an ingenious way to perform both the analytics and the data processing required to use flash memory effectively," says Keshav Pingali, a professor of computer science at the University of Texas at Austin. "In the long run, this may lead to systems that can process large amounts of data efficiently on laptops or desktops, which will revolutionize how we do big-data processing."

Because the host can be so low-powered, Jun says, a long-term goal is to create a general-purpose platform and software library for consumers to develop their own algorithms for applications beyond graph analytics. "You could plug this platform into a laptop, download [the software],

and write simple programs to get server-class performance on your laptop," he says.

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