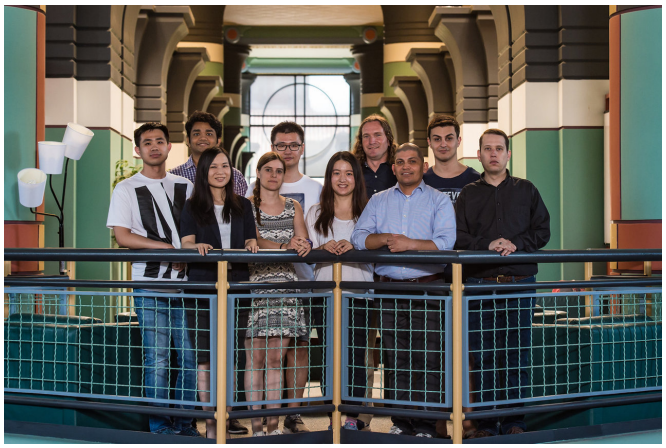
A system purely for developing high-performance, big data codes

11 June 2018



Rice University's PlinyCompute team includes (from left) Shangyu Luo, Sourav Sikdar, Jia Zou, Tania Lorido, Binhang Yuan, Jessica Yu, Chris Jermaine, Carlos Monroy, Dimitrije Jankov and Matt Barnett. Credit: Jeff Fitlow/Rice University

Computer scientists from Rice University's DARPA-funded Pliny Project believe they have the answer for every stressed-out systems programmer who has struggled to implement complex objects and workflows on 'big data' platforms like Spark and thought: "Isn't there a better way?"

Rice's PlinyCompute will be unveiled here Thursday at the 2018 ACM SIGMOD conference. In a peer-reviewed conference paper, the team describes PlinyCompute as "a system purely for developing high-performance, big data codes."

Like Spark, PlinyCompute aims for ease of use and broad versatility, said Chris Jermaine, the Rice computer science professor leading the platform's development. Unlike Spark, PlinyCompute is designed to support the intense kinds of computation that have only previously been possible with supercomputers, or high-performance computers (HPC).

"With machine learning, and especially deep learning, people have seen what complex analytics algorithms can do when they're applied to big data," Jermaine said. "Everyone, from Fortune 500 executives to neuroscience researchers, is clamoring for more and more complex algorithms, but systems programmers have mostly bad options for providing that today. HPC can provide the performance, but it takes years to learn to write code for HPC, and perhaps worse, a tool or library that might take days to create with Spark can take months to program on HPC.

"Spark was built for big data, and it supports things that HPC doesn't, like easy load balancing, fault tolerance and resource allocation, which are an absolute must for data-intensive tasks," he said. "Because of that, and because development times are far shorter than with HPC, people are building new tools that run on top of Spark for complex tasks like machine learning, graph analytics and more."

Because Spark wasn't designed with complex computation in mind, its computational performance can only be pushed so far, said Jia Zou, a Rice research scientist and first author of the ACM SIGMOD paper describing PlinyCompute.

Rice University's PlinyCompute is a big data platform designed specifically for developing high-performance and data-intensive codes. Credit: Pliny Project/Rice University

"Spark is built on top of the Java Virtual Machine, or JVM, which manages runtimes and abstracts away most of the details regarding memory management," said Zou, who spent six years researching large-scale analytics and data management systems at IBM Research-China before joining Rice in 2015. "Spark's performance suffers from its reliance on the JVM, especially as computational demands increase for tasks like training deep neural networks for deep learning.

"PlinyCompute is different because it was designed for high performance from the ground up," Zou said. "In our benchmarking, we found PlinyCompute was at least twice as fast and in some cases 50 times faster at implementing complex object manipulation and library-style computations as compared to Spark."

She said the tests showed that PlinyCompute outperforms comparable tools for construction of high-performance tools and libraries.

Jermaine said not all programmers will find it easy to write code for PlinyCompute. Unlike the Java-based coding required for Spark, PlinyCompute libraries and models must be written in C++.

"There's more flexibility with PlinyCompute," Jermaine said. "That can be a challenge for people who are less experienced and knowledgeable about C++, but we also ran a side-by-side comparison of the number of lines of code that were needed to complete various implementations, and for the most part there was no significant difference between PlinyCompute and Spark."

The Pliny Project, which launched in 2014, is an $11 million, DARPA-funded effort to create sophisticated programming tools that can both "autocomplete" and "autocorrect" code for programmers, in much the same way that software completes search queries and corrects spelling on web browsers and smartphones. Pliny uses machine learning to read and learn from billions of lines of open-source computer programs, and Jermaine said PlinyCompute was born from this effort.

"It's a computationally complex machine learning application, and there really wasn't a good tool for creating it," he said. "Early on, we recognized that PlinyCompute was a tool that could be applied to problems far beyond what we were using it for in the Pliny Project."

More information: Jia Zou et al, PlinyCompute, *Proceedings of the 2018 International Conference on Management of Data - SIGMOD '18* (2018). DOI: 10.1145/3183713.3196933

Provided by Rice University