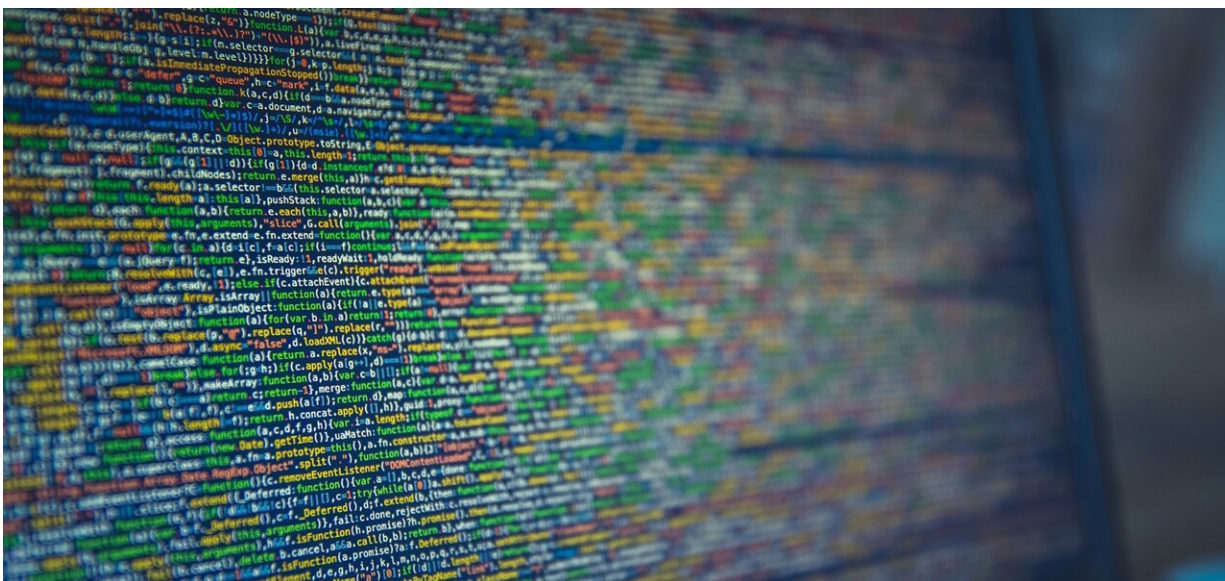


A new, more user-friendly language for programming supercomputers

July 3 2019, by Tom Abate



Can we create a programming environment that doesn't require every researcher to be a computer scientist? Credit: Unsplash/Markus Spiske

Decades ago, computers were costly, complex and rare.

The personal computer revolution changed all that, providing most of us with readily accessible and cheaper gadgets that were smaller, faster and easier to use. Scientists benefited too. They developed computerized techniques to study the inner workings of cells, the orbits of planets around distant stars and other phenomena once far beyond their powers

of observation.

But for researchers at the cutting edge, a certain irony has emerged: New and sophisticated instruments are starting to produce so much data that supercomputers are needed to analyze experimental results. And scientists who try to analyze such huge datasets often struggle to master the complexity of the software needed to program the hardware.

Enter Regent, a new programming language developed by a group led by Stanford computer scientist Alex Aiken. Among other things, Regent makes supercomputers easier to use. "We wanted to create a programming environment that doesn't require every researcher to be a computer scientist," says Aiken, the Alcatel-Lucent Professor in Communications and Networking.

Regent helps solve one of the biggest challenges in supercomputing: Today's supercomputers are far more complex than ever before, and existing [programming languages](#) have struggled to keep pace. A supercomputer may appear in the popular imagination to be one giant machine, but it is in fact an array of thousands of microprocessors that work together. Scientists typically program these arrays using C++, a software language invented some 40 years ago—an eon in computer science time. Back then, the predominant microprocessor was the [central processing unit](#), or CPU, the chip that launched the PC revolution. CPUs solve large problems quickly, one computation after another, in what programmers call a serial fashion.

More recently, however, a second type of microprocessor has become important to supercomputing: the graphics processing unit, or GPU. First used to control millions of pixels on computer screens to improve the visuals of video games, GPUs can perform many similar computations simultaneously, or in parallel, as programmers would say. Parallel processing has proven extremely useful in applications such as machine

learning. C++ has been upgraded to keep up with these and other hardware changes. Unfortunately, the accretion of patches has made the language increasingly difficult to use. Regent, however, makes it easier for a supercomputer programmer to do things like assign serial processing tasks to CPUs and parallel processing tasks to GPUs.

Once Regent has framed the program on a conceptual level, the programmer's intentions are translated—or, to use the technical term, compiled—into a second software layer called Legion, which Aiken also developed. Legion generates machine code—precise instructions directing the supercomputer's hardware how to carry out the program. The tight integration between Regent and Legion makes it easier for programmers to make other important decisions; notably, where to store the data that the supercomputer must analyze.

Elliott Slaughter, a scientist at the SLAC National Accelerator Laboratory who has worked on Regent and Legion almost since their inception, says the integration between the two layers saves programmers both money and time. Computers consume energy, which has a cost. But the [energy cost](#) of moving data can be 100 times the cost of performing computations on that data. Moreover, big experiments often rely on instruments that collect enormous amounts of data. Slaughter said some instruments can collect the data equivalent of 20 video DVDs every second for experiments lasting 15 minutes. Even moving at the speed of light over fiber optics, getting that much data from instrument to supercomputer may create lags that could gum up the analysis. "Where you put the data turns out to be one of the most [important decisions](#) a programmer makes," Slaughter says. Regent and Legion save money and time by giving the [programmer](#) unprecedented control over where to store the data while it is awaiting computation.

Will Regent become widespread? The researchers say new languages must overcome a great deal of inertia. "Regent is a very different way of

programming," Aiken says. "It will take a while for researchers to adopt the required mindset."

But two factors operate in its favor. First, supercomputing hardware continues to improve. The U.S. Department of Energy is driving developments with its Exascale Computing Project, which aims to achieve a 50-fold increase in supercomputational power sometime around 2021. DOE is supporting software projects, including Regent, to help programming keep pace.

Moreover, many scientists who would like to use supercomputers are unfamiliar with the current tools and leery of the steep learning curve required to program big experiments. Even experienced [supercomputer](#) programmers may find the current system cumbersome and wonder if there isn't a better way. "We regularly to talk to scientists who realize how much easier Regent makes life for them," Aiken said.

Provided by Stanford University

Citation: A new, more user-friendly language for programming supercomputers (2019, July 3) retrieved 28 April 2024 from <https://techxplore.com/news/2019-07-user-friendly-language-supercomputers.html>

<p>This document is subject to copyright. Apart from any fair dealing for the purpose of private study or research, no part may be reproduced without the written permission. The content is provided for information purposes only.</p>
--