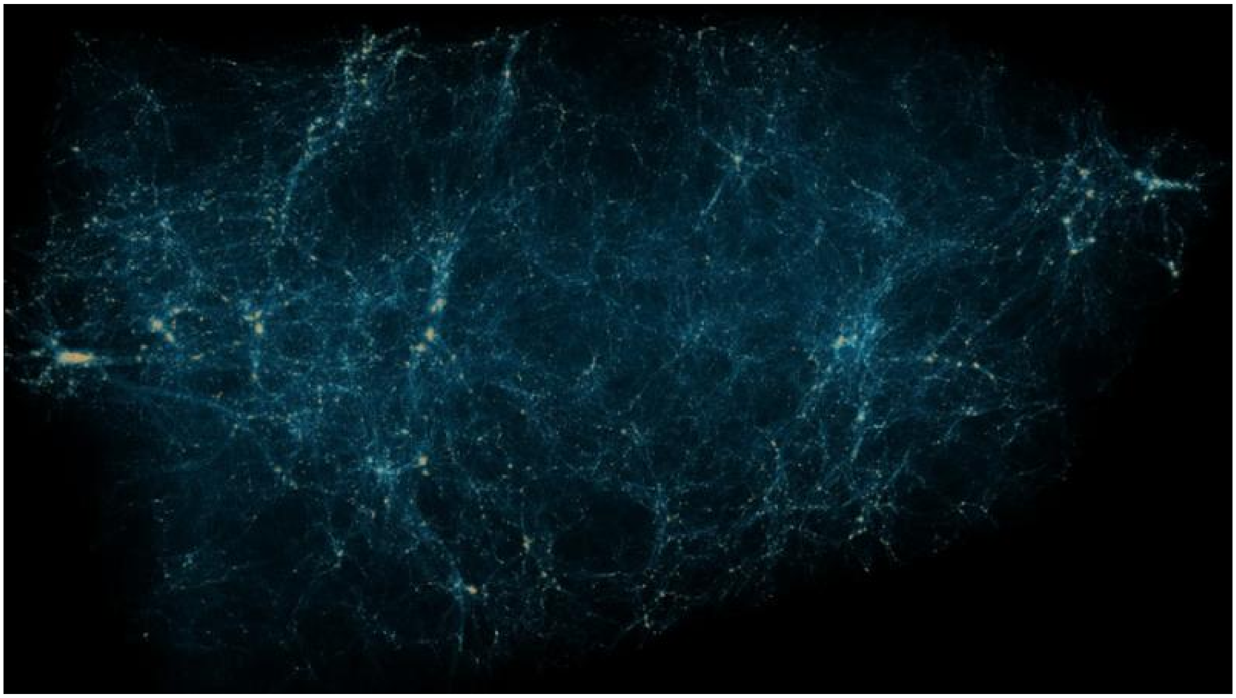
How to build software for a computer 50 times faster than anything in the world

June 16 2017, by Brian Grabowski



This image is a small portion of an output from the "Q Continuum" cosmology simulation; the full simulation evolves more than half a trillion particles. Exascale systems will further enable researchers to run advanced simulations like this to shed more light on the key ingredients that make up our universe. Credit: Hardware/Hybrid Accelerated Cosmology Code (HACC) team

Imagine you were able to solve a problem 50 times faster than you can now. With this ability, you have the potential to come up with answers to even the most complex problems faster than ever before.

Researchers behind the U.S. Department of Energy's (DOE) Exascale Computing Project want to make this capability a reality, and are doing so by creating tools and technologies for exascale supercomputers – computing systems at least 50 times faster than those used today. These tools will advance researchers' ability to analyze and visualize complex phenomena such as cancer and nuclear reactors, which will accelerate scientific discovery and innovation.

Developing layers of software that support and connect hardware and applications is critical to making these next-generation systems a reality.

"These software environments have to be robust and flexible enough to handle a broad spectrum of applications, and be well integrated with hardware and application software so that applications can run and operate seamlessly," said Rajeev Thakur, a computer scientist at the DOE's Argonne National Laboratory and the director of software technology for the Exascale Computing Project (ECP).

Researchers in Argonne's Mathematics and Computer Science Division are collaborating with colleagues from five other core ECP DOE national laboratories – Lawrence Berkeley, Lawrence Livermore, Sandia, Oak Ridge and Los Alamos – in addition to other labs and universities.

Their goal is to create new and adapt existing software technologies to operate at exascale by overcoming challenges found in several key areas, such as memory, power and computational resources.

Checkpoint/restart

Argonne computer scientist Franck Cappello leads an ECP project focused on advanced checkpoint/restart, a defense mechanism for withstanding failures that happen when applications are running.

"Given their complexity, faults in high-performance systems are a common occurrence, and some of them lead to failures that cause parallel applications to crash," Cappello said.

"Many ECP applications already feature checkpoint/restart, but because we're moving towards an even more complex system at exascale, we need more sophisticated methods for it. For us, that means providing an effective and efficient checkpoint/restart for ECP applications that lack it, and providing other applications a more efficient and scalable checkpoint/restart."

Cappello also leads a project that focuses on reducing the large amounts of data that
is generated by these machines, which is expensive to store and communicate effectively.

"We're developing techniques that can reduce data volume by at least a factor of 10. The problem with this is that you add some margin of error when you reduce the data," Cappello said.

"The focus then is on controlling the margin of error; you want to control the error so it doesn't affect the scientific result in the end while still being efficient at reduction, and this is one of the challenges we are looking at."

Memory

For information that is stored on exascale systems, researchers need data management controls for memory, power and processing cores. Argonne computer scientist Pete Beckman is investigating methods for managing all three through a project known as Argo.

"The efficiency of memory and storage have to keep up with the increase in computation rates and data movement requirements that will exist at exascale," Beckman said.

"But how memory is arranged in systems and the technology used for it is also changing, and has more layers," he said. "So we have to account for these changes, in addition to anticipating and designing around the future needs of the applications that will use these systems."

With added layers of memory on exascale systems, researchers must develop complementary software for regulating these memory technologies that give users control over the process.

"Having controls in place is important because where you choose to store information affects how quickly you can retrieve it," Beckman said.

Power

Another key resource that Beckman and Argo Project researchers are studying is power. As with memory, methods for allocating power resources could speed up or slow computation within a high-performance system. Researchers are interested in developing software technologies that could enhance users' control over this resource.

"Power limits may not be at the top of the list when you're dealing with smaller systems, but when you're talking about tens of megawatts of power, which is what we'll need in the future, how an application uses that power becomes an important distinguishing characteristic," Beckman said.

"The goal for us is to achieve a level of control that maximizes the user's abilities while maintaining efficiency and minimizing cost," he said.

Processing Cores

Ultra-fine controls are also needed for managing cores within an exascale system.

"With each generation of supercomputers we keep adding processing cores, but the system software that makes them work needs ways to partition and manage all the cores," Beckman said. "And since we're dealing millions of cores, even making small adjustments can have a tremendous impact on what we're able to do; improving performance by say, two to three percent, is equivalent to thousands of laptops' worth of computation."

One concept Beckman and fellow researchers are exploring to better manage cores is containerization, a method for grouping a select number of cores together and treating them as a unit, or "container," that can be controlled independently.

"The tools we have now to manage cores are not as precise, making it harder to regulate how much work is being done by one set of cores over another," Beckman said. "But we're borrowing and adapting container concepts into high-performance computing to give users the ability to operate and manage how they're using those cores more carefully and directly."

Software Libraries

Applications rely on software libraries – high-quality, reusable software collections – to support simulations and other functionalities. To make these capabilities accessible at exascale, Argonne researchers are working to scale existing libraries.

"Libraries provide important capabilities, including solutions to numerical problems," said Argonne mathematician Barry Smith, who leads a project focused on scaling two libraries known as PETSc and TAO.

PETSc and TAO are widely used for large-scale numerical simulations. PETSc is a library that provides solutions to specific numerical calculations. TAO is a library that provides solutions to large-scale optimization problems, such as calculating the most cost-effective strategy for reloading fuel rods in a nuclear reactor.

In addition to scaling diverse software libraries, ECP scientists are also looking for ways to improve their quality and compatibility.

"Libraries have traditionally been developed independently, and due to the different strategies used to design and implement them, it's been difficult to use multiple libraries in combinations. But large applications, like those that will run at exascale, need to be able to use all the layers of the software stack in combination," said Argonne computational scientist Lois Curfman McInnes.

McInnes is co-leading the xSDK project, which is determining community policies to regulate the implementation of software packages. Such policies will make it easier for diverse libraries to be compatible with one another.

"These efforts bring us one step closer to realizing a robust and agile exascale environment that can aid scientists in tackling great challenges," McInnes said.

Provided by Argonne National Laboratory

Citation: How to build software for a computer 50 times faster than anything in the world (2017, June 16) retrieved 17 July 2024 from https://phys.org/news/2017-06-software-faster-world.html