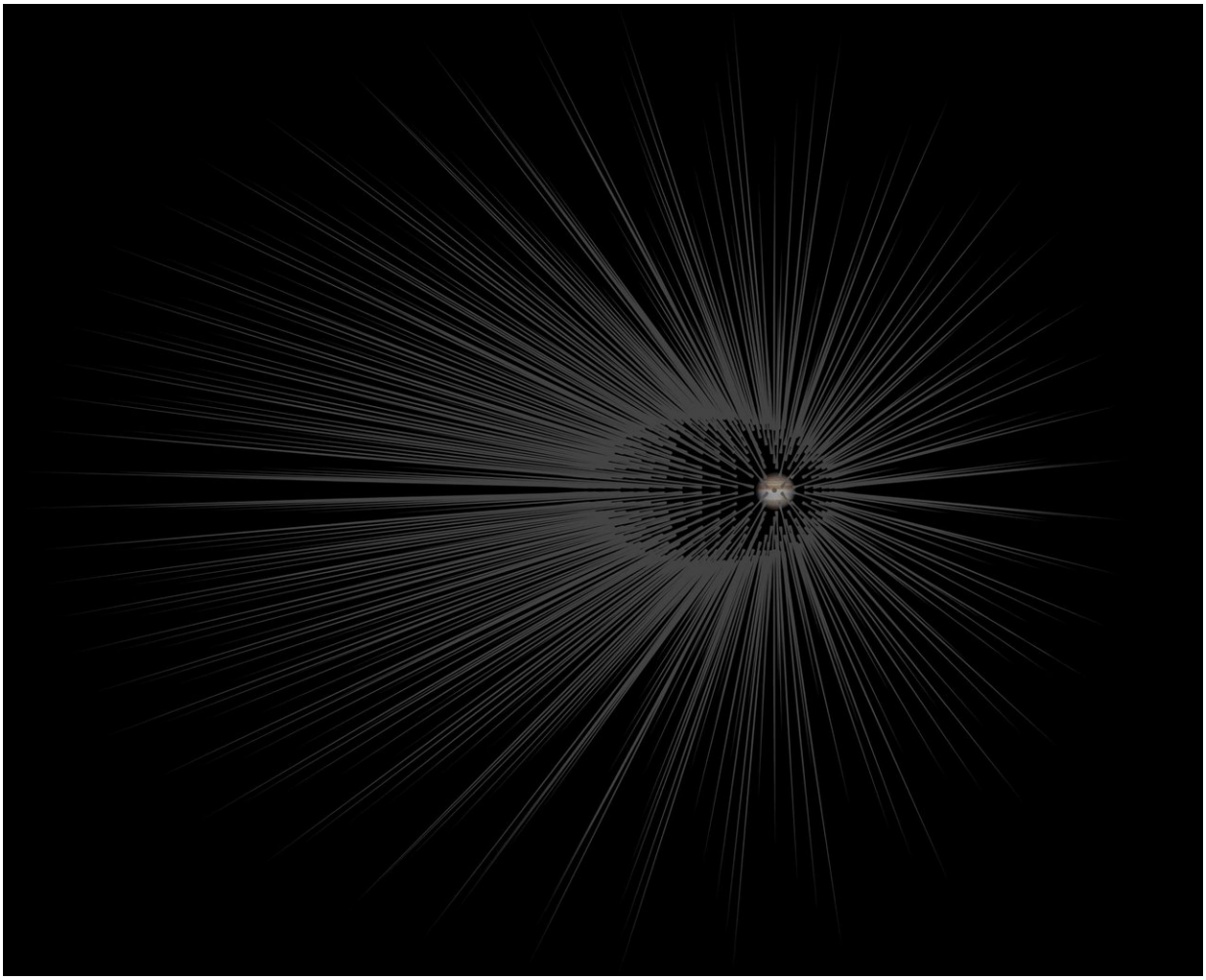
Open-source software for data from high-energy physics

October 31 2017



Proposed filaments of dark matter surrounding Jupiter could be part of the mysterious 95 percent of the universe's mass-energy. Credit: NASA/JPL-Caltech

Most of the universe is dark, with dark matter and dark energy comprising more than 95 percent of its mass-energy. Yet we know little about dark matter and energy. To find answers, scientists run huge high-energy physics experiments. Analyzing the results demands high-performance computing – sometimes balanced with industrial trends.

After four years of running computing for the Large Hadron Collider CMS experiment at CERN near Geneva, Switzerland – part of the work that revealed the Higgs boson – Oliver Gutsche, a scientist at Department of Energy's (DOE) Fermi National Accelerator Laboratory, turned to the search for dark matter. "The Higgs boson had been predicted, and we knew approximately where to look," he says. "With dark matter, we don't know what we're looking for."

To learn about dark matter, Gutsche needs more data. Once that information is available, physicists must mine it. They are exploring computational tools for the job, including Apache Spark open-source software.

In searching for dark matter, physicists study results from colliding particles. "This is trivial to parallelize," breaking the job into pieces to get answers faster, Gutsche explains. "Two PCs can each process a collision," meaning researchers can employ a computer grid to analyze data.

Much of the work in high-energy physics, though, depends on software the scientists develop. "If our graduate students and postdocs only know our proprietary tools, then they'll have trouble if they go to industry," where such software is unavailable, Gutsche notes. "So I started to look into Spark."

Spark is a data-reduction tool made for unstructured text files. That creates a challenge – accessing the high-energy physics data, which are

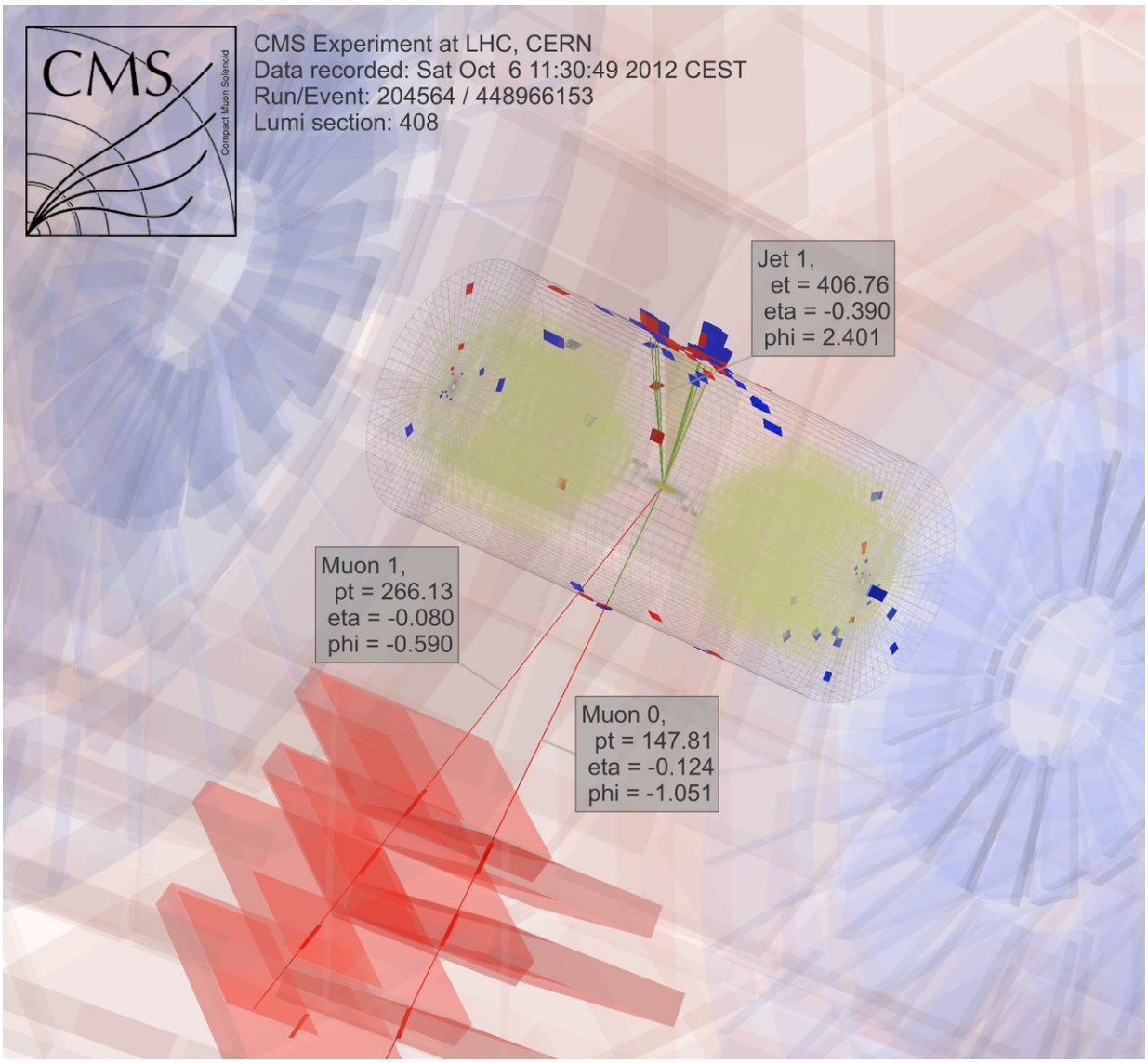
in an object-oriented format. Fermilab computer science researchers Saba Sehrish and Jim Kowalkowski are tackling the task.

Spark offered promise from the beginning, with some particularly interesting features, Sehrish says. "One was in-memory, large-scale distributed processing" through high-level interfaces, which makes it easy to use. "You don't want scientists to worry about how to distribute data and write parallel code," she says. Spark takes care of that.

Another attractive feature: Spark is a supported research platform at the National Energy Research Scientific Computing Center (NERSC), a DOE Office of Science user facility at the DOE's Lawrence Berkeley National Laboratory. "This gives us a support team that can tune it," Kowalkowski says. Computer scientists like Sehrish and Kowalkowski can add capabilities, but making the underlying code work as efficiently as possible requires Spark specialists, some of whom work at NERSC.

Kowalkowski summarizes Spark's desirable features as "automated scaling, automated parallelism and a reasonable programming model."

In short, he and Sehrish want to build a system allowing researchers to run an analysis that performs extremely well on large-scale machines without complications and through an easy user interface.



To search for dark matter, scientists collect and analyze results from colliding particles, an extremely computationally intense process. Credit: CMS CERN

Just being easy to use, though, is not enough when dealing with data from high-energy physics. Spark appears to satisfy both ease-of-use and performance goals to some degree. Researchers are still investigating some aspects of its performance for high-energy physics applications, but computer scientists can't have everything. "There is a compromise,"

Sehrish states. "When you're looking for more performance, you don't get ease of use."

The Fermilab scientists selected Spark as an initial choice for exploring big-data science, and dark matter is just the first application under testing. "We need several real-use cases to understand the feasibility of using Spark for an analysis task," Sehrish says. With scientists like Gutsche at Fermilab, dark matter was a good place to start. Sehrish and Kowalkowski want to simplify the lives of scientists running the analysis. "We work with scientists to understand their data and work with their analysis," Sehrish says. "Then we can help them better organize data sets, better organize analysis tasks."

As a first step in that process, Sehrish and Kowalkowski must get data from high-energy physics experiments into Spark. Notes Kowalkowski, "You have petabytes of data in specific experimental formats that you have to turn into something useful for another platform."

The starting data for the dark-matter implementation are formatted for high-throughput computing platforms, but Spark doesn't handle that configuration. So software must read the original data format and convert it to something that works well with Spark.

In doing this, Sehrish explains, "you have to consider every decision at every step, because how you structure the data, how you read it into memory and design and implement operations for high performance is all linked."

Each of those data-handling steps affects Spark's performance. Although it's too early to tell how much performance can be pulled from Spark when analyzing dark-matter data, Sehrish and Kowalkowski see that Spark can provide user-friendly code that allows high-energy physics researchers to launch a job on hundreds of thousands of cores. "Spark is

good in that respect," Sehrish says. "We've also seen good scaling – not wasting computing resources as we increase the dataset size and the number of nodes."

No one knows if this will be a viable approach until determining Spark's peak performance for these applications. "The main key," Kowalkowski says, "is that we are not convinced yet that this is the technology to go forward."

In fact, Spark itself changes. Its extensive open-source use creates a constant and rapid development cycle. So Sehrish and Kowalkowski must keep their code up with Spark's new capabilities.

"The constant cycle of growth with Spark is the cost of working with high-end technology and something with a lot of development interests," Sehrish says.

It could be a few years before Sehrish and Kowalkowski make a decision on Spark. Converting software created for high-throughput computing into good high-performance computing tools that are easy to use requires fine tuning and team work between experimental and computational scientists. Or, you might say, it takes more than a shot in the dark.

More information: Apache Spark: spark.apache.org/

Provided by US Department of Energy