

OpenACC Enables Astrophysics Researchers to Gain Insight into Dark Energy



OpenACC
Directives for Accelerators

CHALLENGE

Accelerate large HPC code on GPU-powered supercomputers using simple, portable programming approach.

SOLUTION

Leveraged PGI Accelerator compilers with OpenACC on OLCF's Titan supercomputer.

RESULTS

When accelerated, the reactions module was 4.4 times faster than it was when run on a more conventional multi-core computer with 16 cores. Researchers can now run more scientifically interesting simulations by modeling larger nuclear reaction networks in 3D and plan to accelerate the entire application with OpenACC.

Project Overview

Maximilian Katz and Adam Jacobs, PhD candidates in the Department of Physics and Astronomy at Stony Brook University aim to determine the elusive nature of dark energy by understanding the causes of stellar explosions. Katz studies the merger of two stellar remnants, while Jacobs focuses on studying an alternative model in which a stellar remnant gravitationally pulls matter from a companion star and experiences a double detonation.

To that end, Katz and Jacobs are focused on developing computational methods for understanding the origins of type Ia supernovae. Jacobs uses the Fortran-based [MAESTRO](#) to understand the double-detonation origin model. These codes are optimized for studying different phases of explosive stellar events.

Katz uses the software CASTRO, a three-dimensional compressible hydrodynamics code specifically designed for studying the type of astrophysical fluid flows that occur in stellar explosions. Whereas MAESTRO is designed especially to model subsonic fluid flows, CASTRO is a more conventional compressible hydrodynamics code capable of modelling supersonic fluid flows which MAESTRO cannot.

"CASTRO and MAESTRO's microphysics modules are great for GPU acceleration because they only need data that's already on the node, which means it doesn't involve adding the complexity of a multi-node supercomputer," said Jacobs. "The calculation for each individual cell of data is independent, so they can be readily vectorized and massively parallelized."

The team is motivated to accelerate the code, so more complex sets of nuclear reactions can be modeled in 3D simulations—which would be a huge development in the field. "Typically, only the simplest sets of reactions can be modelled in 3D," said Jacobs. "The nuclear reactions, even the simplest ones that we tend to use for computational efficiency, take up about 10-20% of a typical MAESTRO calculation, so we expect speeding them up will have a substantial impact on the code."

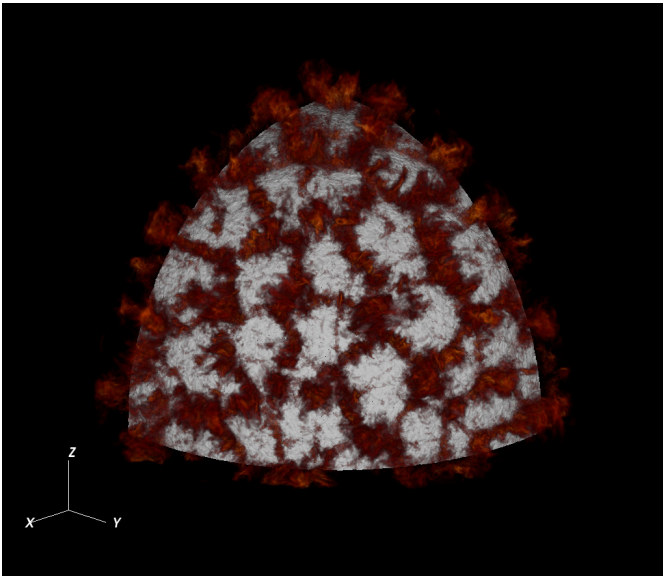
Challenge

The researchers faced two main challenges. First, astrophysical systems contain many widely separate length scales that need to be simulated simultaneously.

Secondly, they must accurately compute the gravitational field of a system that is rapidly changing in time and is far from a nice, mostly spherical object like the Earth or Sun. "This can be a very expensive and communication-driven operation, and can severely limit effective strong scaling by an order of magnitude or so," said Katz. "One of the things I will be trying next

is to express the gravity solving operation in a way that is much less communication intensive, at the expense of potentially requiring more raw FLOPs (Floating Point Operations per Second).

The team is using the Titan supercomputer at OLCF through the DOE INCITE program, where most of the computing power is on GPUs. “For us to remain competitive in applying for time on this system, we must have a way to effectively use the GPUs,” he said.



The researchers had to decide on a language for programming the GPUs. Both CASTRO and MAESTRO are built on top of the BoxLib library for grid management, which has both C++ and Fortran class hierarchies designed to efficiently manage the construction and refinement of spatial grids that represent data on a computational domain. Jacobs is proficient with OpenMP, MPI and OpenACC, a directive-based accelerator programming model targeted at scientists, engineers and other domain experts who are not full-time software developers. Katz has extensive experience programming in OpenMP and moderate experience in MPI—both of which are fundamentally enconced in BoxLib. However, neither of the researchers has much experience with CUDA.

“CUDA is infeasible because it’s too vendor-specific and hardware-specific,” said Jacobs. “For scientific applications that run on several different supercomputing architectures and need to be usable for many generations of architecture, the cons of something like CUDA outweigh the pros. That’s why we prefer OpenACC.”

“The dominant workload in our systems can often be expressed as independent loops over individual points in space, so much of our parallelism is expressed in OpenMP pragmas that accelerate these loops,” said Katz. “OpenACC pragmas were thus the natural target for us to continue using this approach.”

Solution

The team chose to use the OpenACC compiler from PGI, a company at the forefront of OpenACC development. Katz began with vectorizing one of the principal modules—the “equation of state” module—whose job is to evaluate thermodynamic properties on a point-by-point basis. “Learning how to use OpenACC pragmas effectively and vectorizing the module took about two weeks of effort. Another one or two weeks will be spent modifying the code so that we could implement and use the more communication-friendly gravity solver, then accelerate it on the GPUs.”

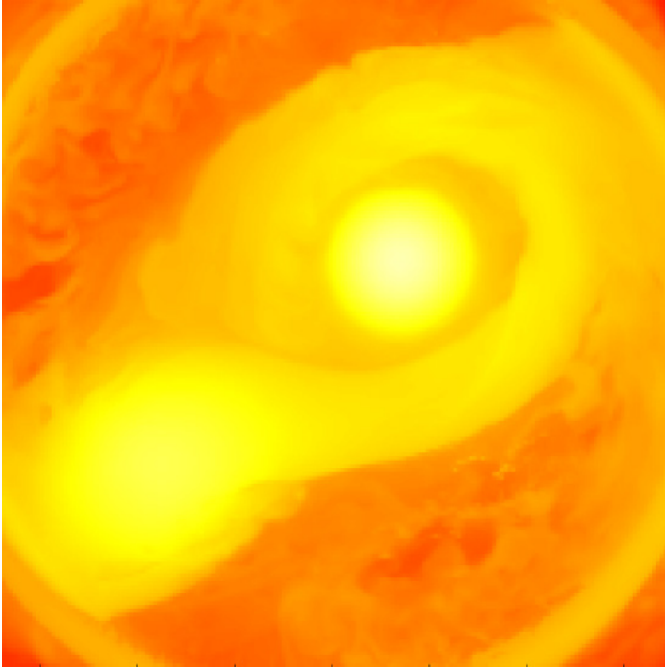
Before accelerating the reactions module, Jacobs started by working with a reduced prototype module. After accelerating it, he found it to be 4.4 times faster than running it on a conventional multi-core computer with 16 cores. Under optimal conditions, applying what was learned with the prototype to the acceleration of MAESTRO’s nuclear reactions modules on the GPUs could result in about a 10% performance improvement in overall execution, compared to running on a multi-core system.

“For scientific applications that run on several different supercomputing architectures and need to be usable for many generations of architecture, the cons of something like CUDA outweigh the pros. That’s why we prefer OpenACC.”

Adam Jacobs, PhD candidates in the Department of Physics and Astronomy at Stony Brook University

Results

Now that they're able to accelerate the microphysics calculations, Katz and Jacobs can run faster, more scientifically interesting simulations.



"If I can successfully implement the gravity method and get the desired performance boost, it may change the fact that I am currently unable to effectively use more than 10-20 thousand cores," said Katz. "If I could boost that by a factor of a few, I might be able to do much higher resolution studies of this system, and zoom in on most interesting regions to determine if they will result in a thermonuclear detonation."

"On the reactions side, accelerated calculations allows us to model larger networks of nuclear reactions for similar computational costs as the simple networks we model now," said Jacobs. "This enables us to do more scientifically accurate and interesting models."

The team has discussed the possibility of putting the entire hydrodynamics solver on the GPUs, in which case the host nodes would be primarily used for communication operations.

"I am currently engaged in a code refactoring effort in CASTRO, which should make it somewhat straightforward to accelerate using OpenACC," said Katz. "Only the first step for GPU acceleration has been made, and the team is working on the second part of the code, with the end goal of accelerating the all of the code on GPUs."