Research team accelerates multi-physics simulations with El Capitan predecessor systems

April 24 2024, by Jeremy Thomas

A 2D MARBL simulation of the N210808 “Burning Plasma” shot performed at the National Ignition Facility at the onset of ignition. This calculation consists of 19 million high-order quadrature points and ran on El Capitan predecessor system rzAdams (on AMD MI300A GPUs). Credit: Rob Rieben.

Researchers at Lawrence Livermore National Laboratory (LLNL) have achieved a milestone in accelerating and adding features to complex

multi-physics simulations run on Graphics Processing Units (GPUs), a development that could advance high-performance computing and engineering.

As LLNL readies for El Capitan, the National Nuclear Security Administration's first exascale supercomputer, the team's efforts have centered around the development of MARBL, a next-generation multi-physics code, for GPUs. El Capitan is based on AMD's cutting-edge MI300A Accelerated Processing Units (APUs), which combines Central Processing Units (CPUs) with GPUs and high-bandwidth memory into a single package, allowing for more efficient resource sharing.

El Capitan's heterogeneous (CPU/GPU) computing architecture, along with expectations that most future supercomputers will be heterogeneous, made it imperative that multi-physics codes like MARBL—which targets mission-relevant high-energy-density (HED) physics like those involved in inertial confinement fusion (ICF) experiments and stockpile stewardship applications—could perform efficiently across a wide variety of architectures, researchers said.

In a recent paper published by the *Journal of Fluids Engineering*, by harnessing the power of GPUs, specifically AMD's MI250X GPUs in El Capitan's early access machines, the researchers successfully extended MARBL's capabilities to include additional physics crucial for HED physics and fusion modeling.

"The big focus of this paper was supporting multi-physics—specifically multi-group radiation diffusion and thermonuclear burn, which are involved in fusion reactions—and the coupling of all of that with the higher-order finite-element moving mesh for simulating fluid motion," principal investigator Rob Rieben said.

"To get performance on the GPU, there is a lot you have to do in terms

of programming, optimizing kernels, balancing memory, and turning your code into a GPU-parallel code, and we were able to accomplish that."

Rieben's team has been dedicated to engineering the scalable, GPU-accelerated multi-physics application MARBL for simulating HED physics experimental platforms since 2015, focusing on the simultaneous advancement of software abstractions and algorithmic developments to enable GPU performance.

The work described in the recent paper is essential for delivering on programmatic tasks that rely heavily on large-scale computational science to answer tough national security questions, said co-author Alejandro Campos, who added that the team faced two main challenges in extending MARBL's capabilities: verifying that additional physics modules were accurately implemented and ensuring that those new modules could perform efficiently when running on the next generation of GPU-based machines.

Researchers said the team addressed those challenges through techniques such as new algorithms for solving linear systems with preconditioners, which have historically been optimized for CPUs. A breakthrough from LLNL's Center for Applied Scientific Computing (CASC) led to a new type of preconditioner suited for GPUs, which was integrated into the code and scaled up for production use.

Preconditioners for linear solvers have been challenging to port to GPUs in a performant way, Rieben said. "CASC proposed a new type of preconditioner needed for solving diffusion equations that is specifically designed to provide high performance for high-order methods on GPUs, which enable us to run large 3D multi-physics simulations on GPU machines like El Capitan.

"Our job was to put their method into a production code, scale it up, and show that it works, not just on benchmarks, but on the actual problems that we care about. We took that hot-off-the-presses research, worked with the researchers in CASC, and got it into our code and did all the necessary tuning to make that perform well on multiple GPU systems," Rieben said.

In the paper, the team compared traditional distributed CPU approaches to the rapid computing enabled by GPU architectures and focused on developing software that could effectively utilize the Single Instruction/Multiple Data paradigm of GPU hardware. The multi-physics nature of the simulations introduced bottlenecks that added complexity to the task, which could degrade overall performance and scalability if not properly addressed, the team reported.

Researchers said the team's use of performance portability abstraction layers, such as the LLNL-developed RAJA Portability Suite and the MFEM finite element discretization library, was instrumental in enabling MARBL's single source code to target multiple GPU/CPU architectures.

"In this paper, we focus on the AMD GPUs because we could leverage other open-source performance portability libraries developed here like RAJA," co-author Tom Stitt said. "While there were some AMD-specific changes that needed to be made, there weren't that many, and they didn't take that much time, so to start our performance portability strategy, that's a win."

Stitt added that getting MARBL to perform on LLNL's current CPU/GPU flagship Sierra took about six years of employee time versus about four months to achieve performance on the El Capitan early-access systems at an 18-fold productivity boost.

"If we had to invest that six years of time again for this new platform,

we wouldn't have succeeded; we'd still be working on it," Stitt said. "Our code successes show that the RAJA Portability Suite is a very viable option for writing codes that will work across CPU and GPU architectures and across different GPU vendors."

In addition to RAJA, Umpire—a programming interface that helped alleviate memory constraints on Sierra—also has helped improve codes for El Capitan, Stitt said. Since El Capitan will have eight times more memory per node than Sierra, researchers will be able to fit much bigger problems on a single node and take advantage of the parallelism that the AMD APUs can provide, researchers said.

"The MI300As are the next evolution in AMD GPU processors, and thus, we are very excited to carry out our simulations with those resources," co-author Alejandro Campos said. "We've relied on various libraries developed at LLNL, such as MFEM, RAJA, Umpire, and others, to abstract away some of the work that went into performance portability, and thus, we hope the transition for MARBL to the newer processors will be as straightforward as possible."

Co-author Aaron Skinner said prior methods to run MARBL on CPU-based machines proved challenging due to differences in architecture. Recognizing these limitations, Skinner worked with other CASC researchers to develop code and algorithmic enhancements suited for GPUs, an effort that has successfully benefitted multiple physics modules.

"We've known for a while that we need matrix-free methods to gain performance on GPUs, but our best linear solvers don't lend themselves easily to that formalism, if at all," Skinner said.

"With CASC, we've spent a lot of time implementing and optimizing those matrix-free methods, which have really paid off because the same

linear solvers can be used across many different types of modules, including radiation diffusion, thermal conduction, and alpha-particle diffusion. Our approach uses a combination of code optimizations and algorithmic restructuring to gain performance in our linear solvers, which tend to make up the bulk of the computational workload."

Researchers said the successful GPU acceleration for MARBL represents a leap forward for high-performance computing and could have significant implications, not just for El Capitan but for computational science overall.

Improving performance portability will improve flexibility while advancing GPU acceleration could lead to more efficient and accurate simulations for real-world scientific problems in high energy density physics—including fusion energy driven by lasers or pulsed power—and codes for aerospace and automotive engineering, materials science, climate, biological applications, and other complex phenomena.

"Performance portability of codes like MARBL will allow for simulations that provide answers much more quickly or simulations that were previously too expensive to carry out even on the largest supercomputers, as it allows for seamless utilization of different GPU hardware without the need for extensive hardware-specific porting," Campos said.

In the paper, the team conducted scaling studies on key physics benchmark problems to demonstrate the success of their approach on various computing architectures, showing the potential of GPU acceleration for high-order finite element multi-physics simulations and highlighting the versatility and adaptability of their performance portability approach.

"The fact that we have a single source code that can target multiple

GPUs from different vendors, that's a really big deal," Rieben said. "At the DOE labs, one of our principles has been that we can't afford to be locked into a specific vendor. That's baked into how we develop our software, so this is a big win for us. It's a big multiplier in terms of being able to run the code on as many platforms as we possibly can."

Researchers said they were able to run problems with MARBL on El Capitan's early access machines, in which the integrated CPU/GPUs share a single memory space at about twice the speed of Sierra and aim to reach a factor of five times or greater on El Capitan's advanced MI300 APUs, and a 15- to 20-fold increase over the Lab's current fastest Commodity Technology Systems.

Rieben said faster computation through GPUs directly correlates with scientific discovery, as researchers learn from running numerous simulations rather than just one. Rapid iteration at high resolution enables users to turn around problems quickly, boosting productivity. Additionally, the increased computational power LLNL will get with El Capitan will allow for larger-scale simulations that were previously unattainable and raise the standard for simulation complexity.

"The ability to rapidly iterate at full fidelity and high resolution in 3D is crucial for efficient discovery," Rieben said. "That's an immediate benefit; people can turn problems around that much faster. So, that speed increase directly translates into a productivity boost for the user."

"The other thing it lets you do is, of course, scale, so now you can consider things at a scale that you wouldn't have considered before. What was once considered cutting-edge will become more commonplace over time."

More information: Thomas Stitt et al, Performance Portable Graphics Processing Unit Acceleration of a High-Order Finite Element Multiphysics Application, *Journal of Fluids Engineering* (2024). DOI: 10.1115/1.4064493

Provided by Lawrence Livermore National Laboratory