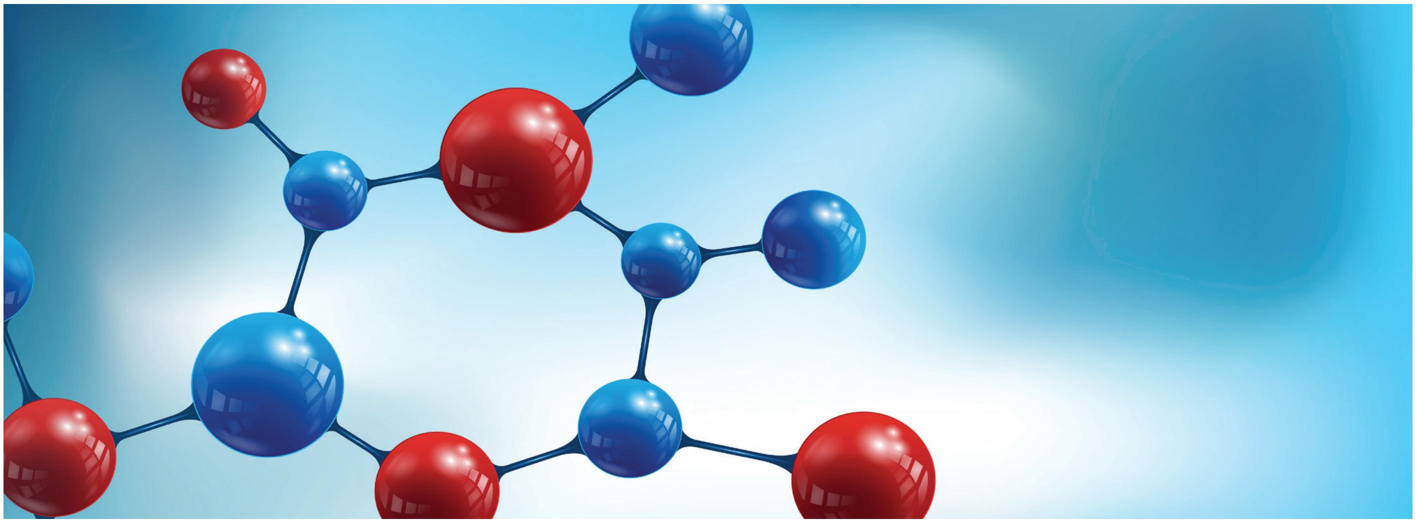
Massively Scaling Computational Electromagnetics Code Using OpenACC



NekCEM, or Nekton for Computational Electromagnetics, is a code designed for highly efficient, accurate predictive modeling of physical systems arising in electromagnetics, photonics, electronics, quantum mechanics, and accelerator physics. It's used in the design of large particle accelerators for producing high-energy photons and in the design of photonic and semiconductor devices for solar energy production.

"NekCEM enables researchers to prototype advanced numerical algorithms for solving the underlying partial differential equations at extreme scales of parallelism," said Dr. Misun Min, a computational scientist at Argonne National Laboratory. "Simulation-based investigation with NekCEM will help research communities understand fundamental physics over range of length scales, with extreme-scale computing capability on the future-generation HPC platforms." Such research, said Min, can significantly reduce the cost and risk of the design and analysis of physical systems.

Challenge

Dr. Min's team needed to process enormous amounts of data with a high degree of accuracy. Frequent performance bottlenecks, excessive time required for data processing, and the need to scale to future GPU-based architectures were the team's main challenges.

"Our principal challenge is rapid time to solution," said Min. "NekCEM strong-scales to a few hundred points per core on the Argonne Leadership Computing Facility (ALCF) Blue Gene/P and Blue Gene/Q. Our biggest challenge on next-generation architectures is to be able to strong- scale but keep low solution times for problems involving a few hundred million grid points."

A second challenge is maintaining code portability. NekCEM is part of a larger code base that has been in development for decades and has hundreds of users. "During this time, many architectures have come and gone, and we cannot port to models for which there is little demand," said Min. "That OpenACC is truly open source was an important factor in our decision to use it for our research."

Solution

The team's efforts included developing an OpenACC implementation of the local gradient and spectral-element curl operator for solving the Maxwell equations.

"The OpenACC implementation covers all solution routines for the Maxwell equation solver in NekCEM, including a highly tuned element-by-element operator evaluation and a GPUDirect gather-scatter kernel to effect nearest-neighbor flux exchanges," Min said. Modifications were designed to make effective use of vectorization, streaming, and data management.

The team had two goals:

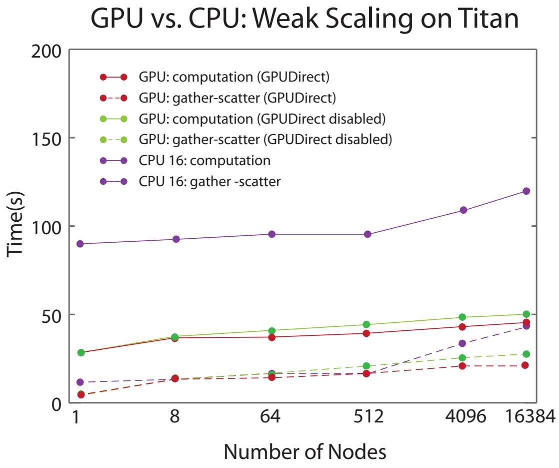
- Develop a high-performance GPU-based operational variant of NekCEM that supports the full functionality of the existing CPU-only code in Fortran/C
- Analyze performance bottlenecks and infer potential scalability for GPU-based architectures of the future

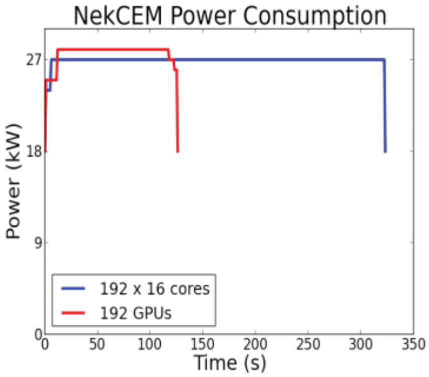
The initial port of the Maxwell solver took about 40 person-days. OpenACC enabled Min's team to port the entire NekCEM application to a GPU. "Because of Amdahl's law, we had to port many routines before the benefits of a hybrid CPU/GPU model came to fruition," said Min. "OpenACC allows us to do this with relative ease."

Results

The NekCEM team tested strong and weak scaling on up to 16,384 GPUs compared with 262,144 CPU cores, achieving more than a 2.5x speedup over a highly tuned CPU-only version of the code on the same number of nodes, for problem sizes up to 6.9 billion grid points.

"The most significant result from our performance studies is faster computation with less energy consumption compared with our CPU-only runs, Min said. "The GPU required only 39 percent of the energy needed for 16 CPUs to do the same computation."





Next Steps

Min plans to extend the OpenACC format for other NekCEM models. "We are currently using the OpenACC variant of NekCEM for electromagnetics computations," she said. "We are also using the NekCEM port as a prototype for porting our fluids and drift-diffusion solvers to OpenACC."

A paper summarizing the results of using OpenACC has been submitted to the International Journal of High Performance Computing Application and can be found here. http://www.mcs.anl.gov/papers/P5341-0415.pdf.