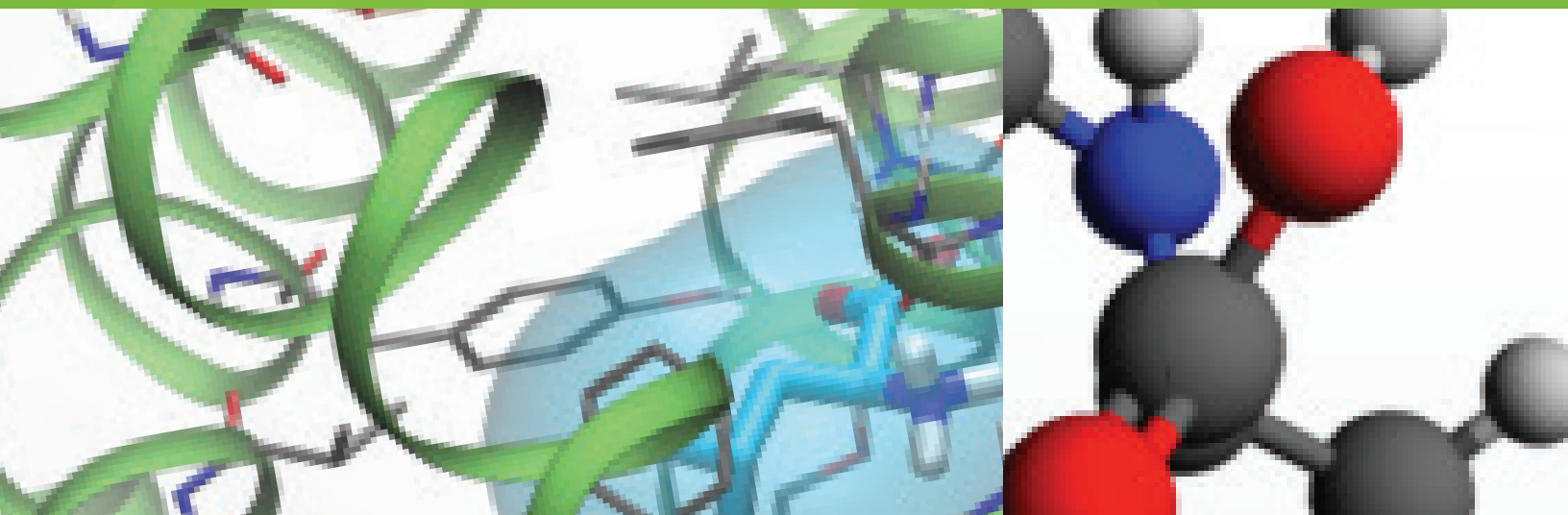


SUCCESS STORY | AARHUS UNIVERSITY

ACCELERATING QUANTUM CHEMISTRY

Quantum chemist leverages OpenACC to
compute high-accuracy molecular energies.



NVIDIA.

A quantum chemist without any formal education in computer science achieves 12x speed-up modifying less than 100 lines of code in one week of programming effort.

SUMMARY

CHALLENGE

Simulate molecular structures of unprecedented size and scale that would require computational capabilities of GPU-powered supercomputers by accelerating LSDalton, a quantum chemistry application

SOLUTION

Used the OpenACC Toolkit, which includes the PGI Accelerator Compilers with OpenACC to optimize and accelerate LSDalton on the Oak Ridge Titan supercomputer.

RESULT

Achieved up to 12x speedups on GPU compared to optimized CPU implementation after modifying fewer than 100 lines of code with one week of programming effort, enabling the team to solve larger scientific problems than previously possible.

Janus Juul Eriksen, a Ph.D. fellow at Aarhus University in Denmark, is part of the research team developing the quantum chemistry code LSDalton, a massively parallel and linear-scaling program for the accurate determination of energies and other molecular properties of large molecular systems. The team needed higher performance for LSDalton to improve accuracy for simulating larger molecular systems for accelerating discoveries in materials and quantum chemistry. With insatiable need for compute cycles, the team sought access on one of world's largest supercomputers, Oak Ridge National Laboratory's Titan system, and was awarded allocation through the INCITE program.

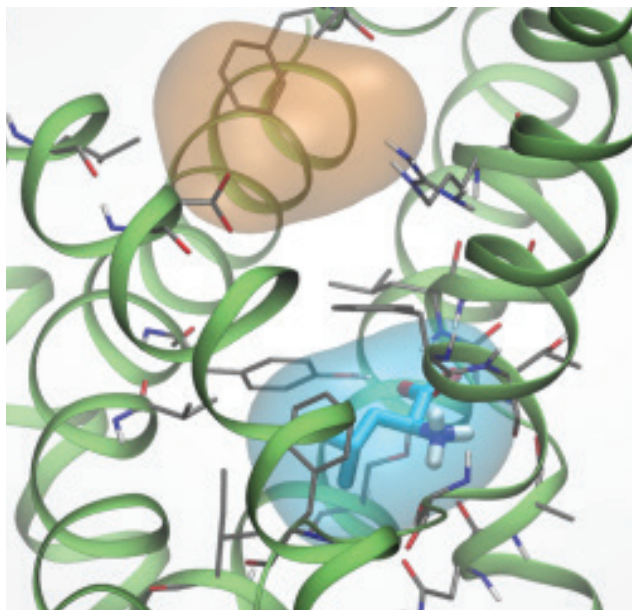
"The application of quantum mechanics to molecular systems and phenomena has become an integral tool to all of chemical, biological, and general material sciences," said Eriksen.

"Besides contributing qualitative information on molecules and their different interactions, modern quantum chemistry may also provide a deeper understanding of molecular processes, which cannot be derived from experimental work alone."

As a result, there is a desire for accurate computer simulations on increasingly larger molecular systems, not only by academia, but also by various industrial research labs.

CHALLENGE

The key challenges are two folds. First, the team wanted to enable LSDalton to scale with size of any molecular system, which grows dramatically whenever higher accuracy is desired while ensuring that code as a whole remains portable. "The code is bound to install and run as easily on a regular workstation or a modest Beowulf cluster, as it



Left: The active site of the Bacterial Leucine Transporter (LeuT) protein, on which the LSDalton team has been doing a number of high-accuracy calculations. The LeuT protein belongs to a family of transporters that are linked to neurological diseases.

will on a modern supercomputer such as Titan. Adding any accelerated code to the source must not interfere with the compilation process when building on architectures where accelerators are absent.” said Eriksen.

Second challenge was time. Eriksen’s Ph.D. studies in Denmark cannot last for longer than a maximum of three years. He had to be able to apply the program to molecular problems in organic chemistry or material sciences, as well as implement the given theoretical method. A chemist by trade, Eriksen has no formal education in computer science, so it was imperative the accelerated code should be relatively easy to write and maintain.

SOLUTION

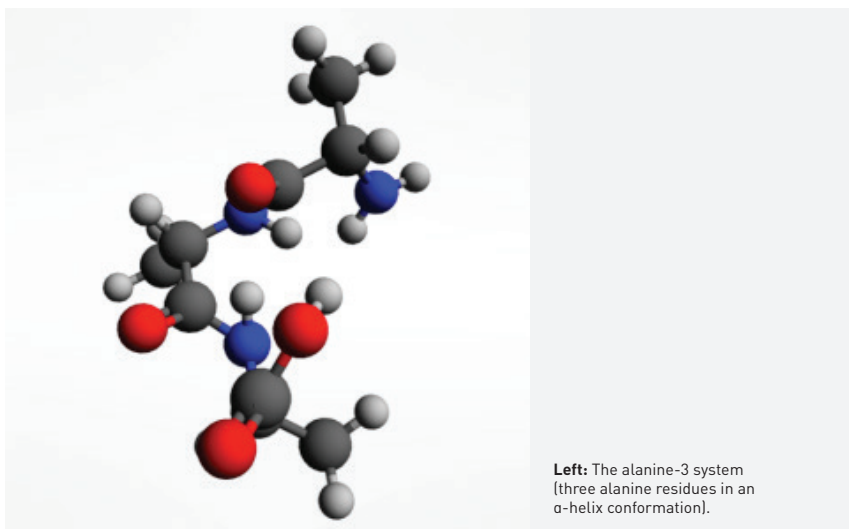
Eriksen and his team chose OpenACC as the platform for developing the code. OpenACC is a high-level approach for researchers and scientists who need to rapidly boost application performance for faster science while maintaining code portability on various systems. With OpenACC, the original source code is kept intact, making the implementation intuitively transparent and leaving most of the work to the compiler.

Eriksen used the new OpenACC Toolkit, which includes a free PGI OpenACC compiler for academic researchers, that helps researchers and scientists accelerate applications.

“OpenACC was much easier to learn than OpenMP or MPI,” said Eriksen. “It makes GPU computing approachable for domain scientists. Our initial OpenACC implementation required only minor efforts, and more importantly, no modifications of our existing CPU implementation.”

“OpenACC was much easier to learn than OpenMP or MPI. Our initial OpenACC implementation required only minor efforts, and more importantly, no modifications of our existing CPU implementation.”

Janus Juul Eriksen, Ph.D.
fellow at Aarhus University
in Denmark



RESULT

Using OpenACC resulted in up to 12x speedups compared to CPU-only code after modifying fewer than 100 lines of code with one week of programming effort. Additionally, with demonstrated success of the GPU-accelerated code on Titan, Eriksen's project was selected to be one of 13 application code projects to join the Center for Accelerated Application Readiness (CAAR) program. This means they will be among the first applications to run on Summit, the new, pre-Exascale supercomputer expected to deliver more than five times the computational performance of the Titan supercomputer.

Eriksen said the main advantage of OpenACC is that any accelerated code making use of the standard is based on original source code, regardless of whether the code is written in Fortran, C, or C++. This makes implementation intuitive and transparent, and easier to maintain and extend. “OpenACC enabled us to accelerate the code with minimal effort.” he said.

“Using OpenACC leaves most of the hard work to the compilers. Future developers are guaranteed their code will execute on future architectures, as well.”

OpenACC

Directives for Accelerators

OpenACC is a directives-based parallel programming model designed for simplifying programming of massively parallel processors. The model provides performance portability across a wide range of platforms, including host-GPU, multi-core, and many-core processors. OpenACC is complementary to and interoperates with existing HPC programming models including OpenMP, MPI, and CUDA.

To learn more and to download the free OpenACC Toolkit visit developer.nvidia.com/openacc

Join the discussion on Stack Overflow bit.ly/openaccforum