PGI®

SUCCESS STORY

AWE Demonstrates OpenACC Performance Portability

OpenACC
Directives for Accelerators

CHALLENGE
Determine whether achieving performance portability across system architectures with legacy applications is possible.

SOLUTION
Using the OpenACC parallel programming model, port the CloverLeaf mini app to GPUs and then recompile and run the same source code on multi-core CPUs comparing that performance to an OpenMP version optimized to run in parallel on CPUs.

RESULTS
- Established OpenACC as a viable option for porting applications to run optimally on multiple system architectures
- Used an OpenACC GPU version of CloverLeaf with no code changes to run on a CPU with performance equal to an optimized OpenMP version
- Achieved 4x faster performance using the same code on a GPU than on a CPU

Project Overview

"The field of science for our current work is computer science," said Oliver Perks, of AWE Management Limited (AWE). "We believe we have a major challenge exploiting many core technologies with our legacy scientific code base."

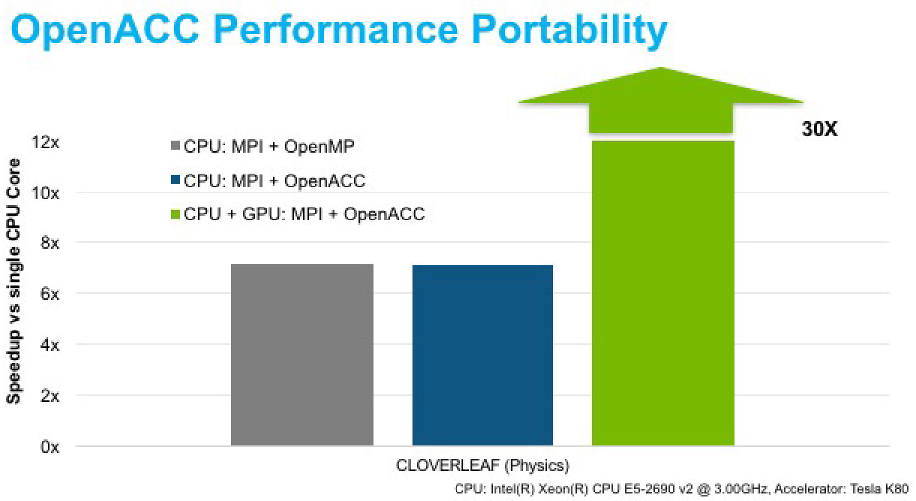
According to Perks, it's nearly impossible to port legacy code in its current form on many core architectures, so the team is using mini-apps to pick out key parts of its science algorithms they believe will need to work effectively at huge scale.

"We want to make the code as performant, portable and as scalable as we can," said Perks. "Once we've decided on the best approach from lessons learned using the mini-apps, we can use the same approach to refactor our legacy code base."

The team, along with its academic partners, has developed a high level of experience with parallel programming languages, including OpenMP, MPI, CUDA, OpenACC, OpenCL, OpenMP4, Kokkos, Raja and DSLs. "We have ported a number of mini-apps to a significant set of these languages and are evaluating them from a performance, portability and maintainability standpoint in a living code that still requires additional physics models and development," he said.

Although the process is time-consuming, it will provide insight as to what the solution will look like and how it will perform. The key metric is how long the applications take to run.

"The goal of our research is to be able to use multi-petaflop systems effectively to deliver scientific results more quickly and with greater fidelity," said Perks.



"We were extremely impressed that we can run OpenACC on a GPU with no code change and get equivalent performance to our CPU OpenMP/MPI implementation. The same code delivers up to 4x faster performance on a GPU. From a performance portability and code future-proofing perspective, this is an excellent result."

— Oliver Perks and Wayne Gaudin of AWE Management Limited (AWE)

Challenge

The team's overarching goal was identifying a high-performance portable programming model for porting legacy codes. Perks noted several challenges to completing the research, many of which depend on the mini-app being employed. They include:

1. Finding the initial concurrency in the algorithms, ideally without changing the numerical results
2. Finding concurrency that works on different core hardware, without major changes
3. Maximizing memory bandwidth
4. Maximizing MPI performance at the interconnect level

Solution

To tackle the challenges, the team chose to use OpenACC, a directive-based accelerator programming model targeted at scientists, engineers and other domain experts who are not full-time software developers.

"We are trying OpenACC because the vast majority of our code developers are applied mathematicians and computational physicists who need a familiar environment such as Fortran, and a relatively simple programming model to extract parallelism," said Perks. "So, while we can experiment with lower level languages, it's unlikely we could retrain our developers in those languages, or rewrite a huge legacy code base in a potentially vendor-specific language that requires detailed knowledge. OpenACC provides a level of abstraction from the hardware, eliminating this requirement."

Results

The results were promising. "We were extremely impressed that we can run a GPU OpenACC code on a CPU with no code change using the PGI Accelerator compilers, and get equivalent performance to our OpenMP/MPI implementation," said Perks. "The same code delivers up to 4x faster performance on a GPU. From a performance portability and code future-proofing perspective, this is an excellent result."

CloverLeaf now has a version that runs equally well on a GPU and CPU with no code changes, and with optimum performance. As a result, Perks's team has met its goals for performance and portability on this mini-app.

"Finding a single programming, vendor-agnostic model that delivers the necessary performance, scalability and maintainability is critical, and we know OpenACC is a viable solution," he said.

According to Perks, the team's next steps will involve looking at using OpenACC in other mini-apps to solve physics problems of interest. "We will also be working to understand how we can use heterogeneous technology such as the OpenPOWER+Tesla nodes effectively, ideally through OpenACC/OMP4," he said.