OpenACC Programming for accelerators

OpenACC Description
OpenACC is a directive-based programming model to create portable accelerated applications. The OpenACC directives allow the programmer to specify regions of code to offload from a host CPU to an attached accelerator. OpenACC provides portability across operating systems, host CPUs and accelerators.

Course Overview
The OpenACC course is on-site 2-day training solution that introduces the attendees to the development environment and programming of accelerated platforms using OpenACC directives. In addition to OpenACC programming theory, this course provides hands-on programming experience in developing general-purpose applications for accelerators. Special attention is focused on presenting the architecture characteristics of modern accelerators, such as Nvidia GPUs.

Topics include a short introduction to GPU Programming with CUDA and the OpenACC programming model. It also focused on an incremental porting methodology in order to port efficiently the application on the accelerator. The lab exercises are based on concrete examples of typical computational kernels and are systematically extended to demonstrate the scope of topics covered. Emphasis is put on developing practical working knowledge of the OpenACC directives.

You Will Learn:
• Nvidia GPU accelerator architecture
• Kernel optimization using OpenACC
• Transfer optimization using OpenACC
• Asynchronous behavior
• OpenACC compiler usage

Course Length: 2 days

Target Audience
The target audience is C, C++ and Fortran developer with little or no knowledge of OpenACC programming, notions of multithreading programming or parallel hardware. This course is designed for performance-oriented application developers targeting heterogeneous computing architectures such as GPUs and other co-processing devices. This course is also a great way for project manager to understand the basics of OpenACC and the impact on applications.

Course Contents:
• CUDA Basics
  o Introduction to GPU computing
  o Nvidia GPU architectures
  o CUDA programming model and API
• OpenACC
  o OpenACC overview and compilers
  o OpenACC programming model
  o Managing data with OpenACC
  o OpenACC loop constructs
  o Asynchronous with OpenACC
  o OpenACC runtime API
• OpenACC 2.0 new features
  o Routine
  o Atomic operations
  o Nested parallelism
  o Device type clause

OpenACC Hands-on Programming Experience
Hands-on examples are meant to provide a deep understanding of parallel execution as well as advanced OpenACC concepts. The hands-on course portion starts with simple application code examples to reinforce the semantics of the OpenACC directives. Course sessions include coverage of sample applications to various common algorithms. Learning outcomes include having participants able to develop efficient OpenACC application, understand hardware resource limits/constraints, detect performance bottlenecks, and tune code solutions to specific hardware resources.

Programming activities explored include:
- Performance impact of device memory transfers and efficient/inefficient memory access patterns
- Performance impact of gang/worker/vector configuration
- Asynchronous behavior
- ...

Recommended Prerequisites:
Hands-on will be in C so previous programming experience with C on Linux OS is required. Familiarity with parallel programming concepts such as task parallelism and domain decomposition is a plus.

Course Material:
Mindshare will supply electronic version of the presentation slides including the lab descriptions and source code referenced in the examples and lab exercises. The hands-on will be executed on the customer’s machine (customer’s in-house cluster, etc.).