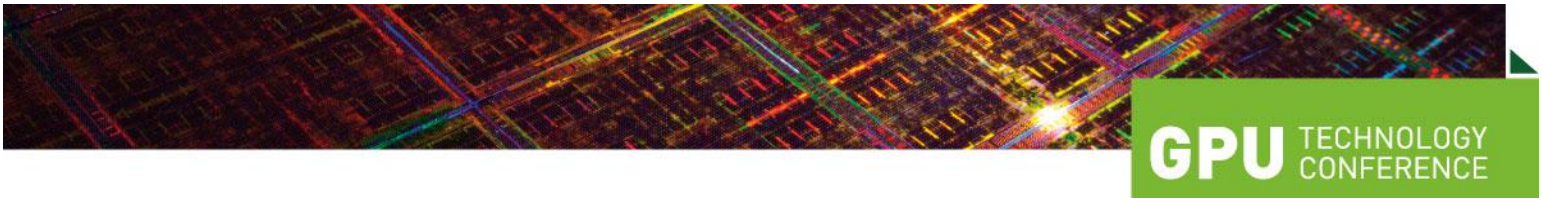


GPU Technology Conference, May 14-17, 2012
McEnery Convention Center, San Jose, California
www.gputechconf.com

Sessions on Parallel Programming Languages & Compilers (subject to change)

IMPORTANT: Visit http://www.gputechconf.com/page/sessions.html for the most up-to-date schedule.

TUTORIALS

S0517A - Programming GPUs with OpenACC (Part 1 of 3)
Mark Harris (NVIDIA), Duncan Poole (NVIDIA), Cliff Woolley NVIDIA)
Day: Monday, 05/14 | **Time:** 10:30 am - 12:00 pm **Topic Areas:** Parallel Programming Languages & Compilers
Session Level: Beginner

OpenACC is a programming standard for parallel computing on accelerators (including GPUs) using directives. It is designed to harness the transformative power of heterogeneous computing systems easily and quickly. In this tutorial you will learn how to add simple compiler hints to your code to expose parallelism to the compiler, allowing it to map computation onto an accelerator. OpenACC directives allow developers to make simple and portable code changes, enabling an easier migration to accelerated computing. This is a 3-part tutorial that will take you from an overview through how to optimize your code. The tutorial starts with an overview of OpenACC programming in which you will learn about applying basic OpenACC directives to your code, with examples. You will also learn more about how GPUs execute parallel programs, and apply this understanding to optimizing more advanced OpenACC examples to gain larger speedups and accelerate applications with various types of parallelism. Lastly, you will see how to use NVIDIA profiling tools to target your optimizations.

S0517B - Programming GPUs with OpenACC (Part 2 of 3)
Mark Harris (NVIDIA), Duncan Poole (NVIDIA), Cliff Woolley NVIDIA)
Day: Monday, 05/14 | **Time:** 1:00 pm - 2:30 pm |
Topic Areas: Parallel Programming Languages & Compilers
Session Level: Beginner

OpenACC is a programming standard for parallel computing on accelerators (including GPUs) using directives. It is designed to harness the transformative power of heterogeneous computing systems easily and quickly. In this tutorial you will learn how to add simple compiler hints to your code to expose parallelism to the compiler, allowing it to map computation onto an accelerator. OpenACC directives allow developers to make simple and portable code changes, enabling an easier migration to accelerated computing. This is a 3-part tutorial that will take you from an overview through how to optimize your code. The tutorial starts with an overview of OpenACC programming in which you will learn about applying basic OpenACC directives to your code, with examples. You will also learn more about how GPUs execute parallel programs, and apply this understanding to optimizing more advanced OpenACC examples to gain larger speedups and accelerate applications with various types of parallelism. Lastly, you will see how to use NVIDIA profiling tools to target your optimizations.

S0517C - Programming GPUs with OpenACC (Part 3 of 3)
Mark Harris (NVIDIA), Duncan Poole (NVIDIA), Cliff Woolley NVIDIA)
Day: Monday, 05/14 | **Time:** 2:30 pm - 4:00 pm
Topic Areas: Parallel Programming Languages & Compilers
Session Level: Beginner

OpenACC is a programming standard for parallel computing on accelerators (including GPUs) using directives. It is designed to harness the transformative power of heterogeneous computing systems easily and quickly. In this tutorial you will learn how to add simple compiler hints to your code to expose parallelism to the compiler, allowing it to map computation onto an accelerator. OpenACC directives allow developers to make simple and portable code changes, enabling an easier migration to accelerated computing. This is a 3-part tutorial that will take you from an overview through how to optimize your code. The tutorial starts with an overview of OpenACC programming in which you will learn about applying basic OpenACC directives to your code, with examples. You will also learn more about how GPUs execute parallel programs, and apply this understanding to optimizing more advanced OpenACC examples to gain larger speedups and accelerate applications with various types of parallelism. Lastly, you will see how to use NVIDIA profiling tools to target your optimizations.

S0522 - Introduction to CUDA Fortran
Gregory Ruetsch (NVIDIA), Massimiliano Fatica, (NVIDIA)
Topic Areas: Parallel Programming Languages & Compilers
Session Level: Beginner

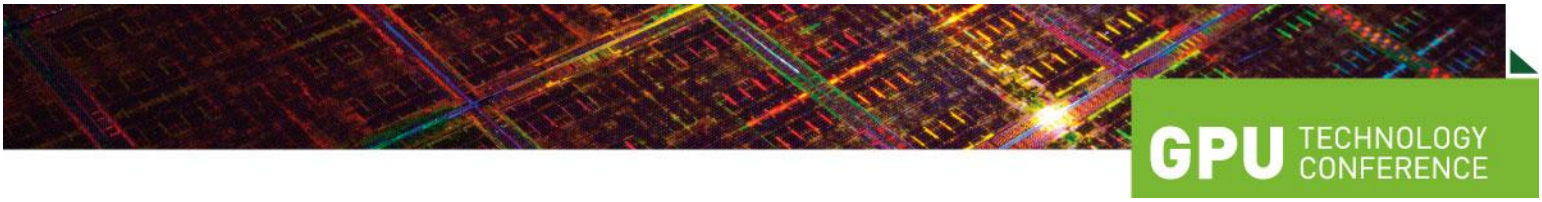
This tutorial will cover various aspects of writing code in CUDA Fortran, which is the Fortran interface to the CUDA architecture. Topics covered will include a basic introduction to parallel programming concepts using CUDA, performance measurements and metrics, optimization, and multi-GPU programming via CUDA 4.0's peer-to-peer capability and MPI. Several case studies will be presented as well.

SESSIONS

S0300 - Jet: A Domain-Specific Approach to Parallelism for Film Fluid Simulation
Dan Bailey (Double Negative)
Day: Tuesday, 05/15 | **Time:** 10:00 am - 10:25 am
Topic Areas: Parallel Programming Languages & Compilers; Digital Content Creation & Film; Computational Fluid Dynamics
Session Level: Intermediate

Discover how a domain-specific language can not only provide fast parallel performance but a simpler user experience in an environment that highly values flexibility. This talk will present the Jet language and heterogeneous compiler built on the LLVM compiler framework that enables efficient generation of X86 machine code or NVIDIA PTX for stencil computation on structured grids. We show that moving target-specific optimizations upstream into the compiler can greatly improve the ability to manipulate the logic of the solver and thus lower the barrier-to-entry for artists and developers without compromising on performance.

S0418 - High Productivity Computational Finance on GPUs
Aamir Mohammad (Aon Benfield Securities), Peter Phillips (Aon Benfield Securities)
Day: Tuesday, 05/15 | **Time:** 2:00 pm - 2:50 pm
Topic Areas: Finance; Application Design & Porting Techniques; Parallel Programming Languages & Compilers
Session Level: Beginner



Learn how Aon Benfield helps clients use GPUs to develop and accelerate Monte Carlo derivatives pricing models. We will present our PathWise software tools used by actuaries and quants in order to rapidly develop and deploy production quality, GPU grid enabled, Monte Carlo models, using only high-level languages and tools without requiring any knowledge of CUDA or C/C++. We will describe our approaching of using Code Generation, Visual Programming, Domain Specific Languages and scripting languages to create a High Productivity Computing software stack for financial services applications.

S0313 - Understanding and using Atomic Memory Operations
Lars Nyland (NVIDIA), Stephen Jones (NVIDIA)
Day: Tuesday, 05/15 | **Time:** 2:00 pm - 2:50 pm
Topic Areas: Algorithms & Numerical Techniques; Parallel Programming Languages & Compilers
Session Level: Advanced

Atomic memory operations provide powerful communication and coordination capabilities for parallel programs, including the well-known operations compare-and-swap and fetch-and-add. The atomic operations enable the creation of parallel algorithms and data structures that would otherwise be very difficult (or impossible) to express without them - for example: shared parallel data structures, parallel data aggregation, and control primitives such as semaphores and mutexes. In this talk we will use examples to describe atomic operations, explain how they work, and discuss performance considerations and pitfalls when using them.

S0515 - Multi-GPU Programming
Paulius Micikevicius (NVIDIA)
Day: Tuesday, 05/15 | **Time:** 2:00 pm - 3:30 pm
Topic Areas: Parallel Programming Languages & Compilers
Session Level: Advanced

CUDA releases starting with 4.0 include a number of features that facilitate multi-GPU programming and computing. In this session we will review the features useful for programming for multiple GPUs, both within a single node and across network. We will cover peer-to-peer GPU communication, communication patterns for various GPU topologies, as well as streams in the context of multiple GPUs. Concepts will be illustrated with a case study of 3D forward wave modeling, common in seismic computing.

S0407 - A High Level Programming Environment for Accelerated Computing
Luiz DeRose (Cray Inc.)
Day: Tuesday, 05/15 | **Time:** 3:00 pm - 3:50 pm
Topic Areas: Development Tools & Libraries; Parallel Programming Languages & Compilers
Session Level: Intermediate

One of the critical hurdles for the widespread adoption of accelerated computing in HPC is programming difficulty. Users need a simple programming model that is portable and is not significantly different from the approaches used on current multi-core x86 processors. In this talk I will present Cray's strategy to accelerator programming, which is based on a high level programming environment with tightly coupled compilers, libraries, and tools. Ease of use is possible with compiler making it feasible for users to write applications in Fortran, C, C++, tools to help users port and optimize for accelerators, and auto-tuned scientific libraries.

S0089 - Accelerator Directives, OpenACC and OpenMP4ACC
James Beyer (Cray Inc.), David Oehmke (Cray Inc.)
Day: Tuesday, 05/15 | **Time:** 4:00 pm - 4:50 pm
Topic Areas: Parallel Programming Languages & Compilers; Supercomputing
Session Level: Intermediate

Rather than require the programmer to rewrite code for accelerators several directive sets have been created and proposed to support non-cache coherent and cache coherent accelerators. This talk will present the OpenACC specification and its implementation for Cray developers, as well as touch on a similar proposal being evaluated by the OpenMP language committee. The presentation will start by discussing the Memory and Execution model needed to allow a programmer to write codes that will run effectively on both distinct memory systems and unified memory systems. Once a proper background has been set the directives will be examined via usage examples.

S0431 - Evolving Use of GPU for Dassault Systems Simulation Products
Matt Dunbar (Dassault Systemes, SIMULIA)
Day: Wednesday, 05/16 | **Time:** 9:00 am - 9:25 am
Topic Areas: Computational Structural Mechanics; Parallel Programming Languages & Compilers
Session Level: Intermediate

SIMULIA, the Dassault Systems brand for simuliation, has been working with NVidia GPGPU cards to accelerate the computation required in doing large-scale structural finite-element simulations with the widely used Abaqus product line. SIMULIA's initial efforts with GPGPU's have been focused on accelerating particularly costly parts of the code when running both on workstations and clusters. We will look at success in these areas with existing products. Futher SIMULIA is now looking at how evolving programming models like OpenACC open the door to using GPU's as a compute platform more than acceleration for limited parts of an application.

S0235 - Compiling CUDA and Other Languages for GPUs
Vinod Grover (NVIDIA), Yuan Lin (NVIDIA)
Day: Wednesday, 05/16 | **Time:** 10:00 am - 10:50 am
Topic Areas: Parallel Programming Languages & Compilers
Session Level: Advanced

This talk gives an overview of the technology behind NVIDIA's CUDA C and OpenCL C compilers, as well as the GPU architecture as seen from a compiler's perspective. Similarities and differences with compiling to a CPU are also discussed. We provide insights into compiler optimizations affect performance and how other languages could be targeted to GPUs.

S0127 - Petascale Molecular Dynamics Simulations on GPU-Accelerated Supercomputers
James Phillips (University of Illinois)
Day: Wednesday, 05/16 | **Time:** 3:00 pm - 3:25 pm
Topic Areas: Molecular Dynamics; Application Design & Porting Techniques; Parallel Programming Languages & Compilers; Supercomputing
Session Level: Intermediate

The highly parallel molecular dynamics code NAMD was chosen in 2006 as a target application for the NSF petascale supercomputer now known as Blue Waters. NAMD was also one of the first codes to run on a GPU cluster when G80 and CUDA were introduced in 2007. How do the Cray XK6 and modern GPU clusters compare to 300,000 CPU cores for a hundred-million-atom Blue Waters acceptance test? Come learn the opportunities and pitfalls of taking GPU computing to the petascale and the importance of CUDA 4.0 features in combining

multicore host processors and GPUs in a legacy message-driven application.

S0365 - Delite: A Framework for Implementing Heterogeneous Parallel DSLs
HyoukJoong Lee (Stanford University), Kevin J. Brown (Stanford University)
Day: Wednesday, 05/16 | **Time:** 3:00 pm - 3:50 pm
Topic Areas: Parallel Programming Languages & Compilers
Session Level: Intermediate

Domain-specific languages can be a solution for heterogeneous parallel computing since they provide higher productivity and performance. To lower the barrier for DSL development, we implemented the Delite compiler framework and runtime. DSL developers can easily extend the framework to build a new DSL. The framework provides various optimization facilities and automatically generates code for heterogeneous hardware including GPU. The runtime executes the generated code in parallel by scheduling the kernels on target devices and managing the memory allocations and data transfers. This talk will cover the details of Delite with examples from OptiML, a machine learning DSL implemented with the framework.

S0214 - GPU Based Stacking Sequence Optimization For Composite Skins Using GA
Sathya Narayana K. (Infosys Ltd.), Ravikumar G.V.V. (Infosys Ltd., Bangalore)
Day: Wednesday, 05/16 | **Time:** 3:00 pm - 3:25 pm
Topic Areas: Computational Structural Mechanics; Algorithms & Numerical Techniques; Parallel Programming Languages & Compilers; Algorithms & Numerical Techniques
Session Level: Advanced

The goal of this session is to showcase how GPUs can be used to achieve high performance in a Genetic algorithm based optimization. The particular domain applied is stacking sequence optimization of Aircraft wing skins. The concepts illustrated use CUDA but are generic to any other GPU language. It is assumed that the registrants have exposure to optimization in engineering domain.

S0514 - GPU Performance Analysis and Optimization
Paulius Micikevicius (NVIDIA)
Day: Wednesday, 05/16 | **Time:** 3:30 pm - 6:30 pm
Topic Areas: Parallel Programming Languages & Compilers
Session Level: Advanced

This session will present the fundamental performance-optimization concepts and illustrate their practical application in the context of programming for Fermi and Kepler GPUs. The goal is twofold: make the optimization process a methodical sequence of steps, facilitate making performance-aware algorithmic decisions before coding even starts. In order to maximize GPU performance, a code should have sufficient parallelism, access memory in a coalesced pattern, and be amenable to vector execution within warps (groups of 32 threads). We will show how to quantify these requirements for a specific GPU in order to determine performance limiters and their importance for a given code. To address the limiters, we will review hardware operation specifics and related optimization techniques. Optimization process will be illustrated using NVIDIA profiling tools and kernel case studies.

S0286 - Scaling Applications to a Thousand GPUs and Beyond
Alan Gray (The University of Edinburgh), Roberto Ansaloni (Cray Italy)
Day: Wednesday, 05/16 | **Time:** 4:00 pm - 4:50 pm
Topic Areas: Supercomputing; Computational Fluid Dynamics; Parallel Programming Languages & Compilers; Application Design & Porting Techniques
Session Level: Intermediate

Discover how to scale scientific applications to thousands of GPUs in parallel. We will demonstrate our techniques using two codes representative of a wide spectrum of programming methods. The Ludwig lattice Boltzmann package, capable of simulating extremely complex fluid dynamics models, combines C, MPI and CUDA. The Himeno three-dimensional Poisson equation solver benchmark combines Fortran (using the new coarray feature for communication) with prototype OpenMP accelerator directives (a promising new high-productivity GPU programming method). We will present performance results using the cutting-edge massively-parallel Cray XK6 hybrid supercomputer featuring the latest NVIDIA Tesla M2090 GPUs.

S0299 - Exploiting Fault Tolerant Heterogeneous Parallelism with SPM.Python
Minesh B Amin (MBA Sciences)
Day: Wednesday, 05/16 | **Time:** 4:00 pm - 4:25 pm
Topic Areas: Parallel Programming Languages & Compilers
Session Level: Advanced

In this session, we shall review how SPM.Python enables the exploitation of parallelism across servers, cores and GPUs in a fault tolerant manner. We will start off by describing the how/what/why SPM.Python augments the traditional (serial) Python with parallel concepts like parallel task managers and communication primitives. Specifically, the context for and solutions to three formally open technical problems will be described. We will conclude by reviewing examples of how SPM.Python can be used to exploit both coarse and fine grain parallelism using GPUs within and across servers in a fault tolerant manner.

S0367 - Physis: An Implicitly Parallel Framework for Stencil Computations
Naoya Maruyama (Tokyo Institute of Technology)
Day: Wednesday, 05/16 | **Time:** 4:30 pm - 4:55 pm
Topic Areas: Parallel Programming Languages & Compilers; Supercomputing; Development Tools & Libraries; Computational Fluid Dynamics
Session Level: Intermediate

This session presents how to implement finite difference methods in a concise, readable, and portable way, yet achieving good scalability over hundreds of GPUs, using the Physis high-level application framework. Physis extends the standard C language with a small set of custom declarative constructs for expressing stencil computations with multidimensional structured grids, which are automatically translated to CUDA for GPU acceleration and MPI for node-level parallelization with automatic domain-specific optimizations such as overlapped boundary exchanges. We demonstrate the programmability improvement and performance of Physis using hundreds of GPUs on TSUBAME2.0.

S0525 - Copperhead: Data Parallel Python
Bryan Catanzaro (NVIDIA)
Day: Wednesday, 05/16 | **Time:** 4:30 pm - 4:55 pm
Topic Areas: Parallel Programming Languages & Compilers
Session Level: Intermediate

Copperhead is a data parallel language suitable for GPU programming, embedded in Python, which aims to

provide both a productive programming environment as well as excellent computational efficiency. Copperhead programs are written in a small, restricted subset of the Python language, using standard constructs like map and reduce, along with traditional data parallel primitives like scan and sort. Copperhead programs interoperate with existing Python numerical and visualization libraries such as NumPy, SciPy, and Matplotlib. In this talk, we will discuss the Copperhead language, the open-source Copperhead runtime, and selected example programs.

S0298 - Performance Tools for GPU-Powered Scalable Heterogeneous Systems
Allen Malony (University of Oregon)
Day: Wednesday, 05/16 | **Time:** 5:00 pm - 5:50 pm
Topic Areas: Development Tools & Libraries; Parallel Programming Languages & Compilers; Application Design & Porting Techniques
Session Level: Intermediate

Discover the latest parallel performance tool technology for understanding and optimizing parallel computations on scalable heterogeneous platforms. The session will present the TAU performance system and its support of measurement and analysis of heterogeneous platforms composed of clusters of shared-memory nodes with GPUs. In particular, TAU's integration of the CUPTI 4.1+ technology will be described and demonstrated through CUDA SDK examples and the SHOC benchmarks. Attendees will be provided LiveDVDs containing the TAU toolsuite and many pre-installed parallel tool packages. It will also include the last CUDA driver, runtime library, and CUPTI.

S0242 - Harnessing GPU Compute with C++ AMP (Part 1 of 2)
Daniel Moth (Microsoft)
Day: Wednesday, 05/16 | **Time:** 5:00 pm - 5:50 pm
Topic Areas: Parallel Programming Languages & Compilers; Development Tools & Libraries
Session Level: Intermediate

C++ AMP is an open specification for taking advantage of accelerators like the GPU. In this session we will explore the C++ AMP implementation in Microsoft Visual Studio 11. After a quick overview of the technology understanding its goals and its differentiation compared with other approaches, we will dive into the programming model and its modern C++ API. This is a code heavy, interactive, two-part session, where every part of the library will be explained. Demos will include showing off the richest parallel and GPU debugging story on the market, in the upcoming Visual Studio release.

S0244 - Harnessing GPU Compute with C++ AMP (Part 2 of 2)
Daniel Moth (Microsoft)
Day: Thursday, 05/17 | **Time:** 10:00 am - 10:50 am
Topic Areas: Parallel Programming Languages & Compilers; Development Tools & Libraries
Session Level: Intermediate

C++ AMP is an open specification for taking advantage of accelerators like the GPU. In this session we will explore the C++ AMP implementation in Microsoft Visual Studio 11. After a quick overview of the technology understanding its goals and its differentiation compared with other approaches, we will dive into the programming model and its modern C++ API. This is a code heavy, interactive, two-part session, where every part of the library will be explained. Demos will include showing off the richest parallel and GPU debugging story on the market, in the upcoming Visual Studio release.

S0338 - New Features In the CUDA Programming Model
Stephen Jones (NVIDIA)
Day: Thursday, 05/17 | **Time:** 10:00 am - 10:50 am
Topic Areas: Parallel Programming Languages & Compilers
Session Level: Intermediate

The continuing evolution of the GPU brings with it new hardware capabilities and new functionality. Simultaneously, ongoing development of CUDA and its tools, libraries and ecosystem brings new features to the software stack as well. Come and learn from on of CUDA's programming model architects about what's new in the GPU, what's coming in the next release of CUDA, how it works, and how it all fits together.

S0039 - Data-Driven GPGPU Ideology Extension
Alexandr Kosenkov (University of Geneva), Bela Bauer (Microsoft Research)
Day: Thursday, 05/17 | **Time:** 10:00 am - 10:25 am
Topic Areas: Application Design & Porting Techniques; Computational Physics; Parallel Programming Languages & Compilers; Development Tools & Libraries
Session Level: Advanced

In this session we will demonstrate how the GPGPU ideology can be extended so that it can be used on a scale of Infiniband hybrid system. The approach that we are presenting combines delayed execution, scheduling techniques and, most importantly, casts down the CPU multi-core ideology to the streaming multiprocessor's one enforcing full-fledged "GPGPU as a co-processor" way of programming for large-scale MPI hybrid applications. Staying compatible with modern CPU/GPGPU libraries it provides more than a fine grained control over resources - more than you wanted that is.

S0320 - PTask: OS Support for GPU Dataflow Programming
Christopher Rossbach (Microsoft Research Silicon Valley), Jon Currey (Microsoft Research Silicon Valley)
Day: Thursday, 05/17 | **Time:** 2:00 pm - 2:50 pm
Topic Areas: Development Tools & Libraries; General Interest; Parallel Programming Languages & Compilers
Session Level: Advanced

This session considers the PTask API, OS-level abstractions that support GPUs as first-class computing resources, and supports a dataflow programming model. With PTask, the programmer specifies where data goes, rather than how and when it should get there, allowing the system to provide fairness and isolation guarantees, streamline data movement in ways that currently require direct programmer involvement, and enable code portability across diverse GPU-based platforms. Our experience building the PTask APIs shows that PTask can provide important system-wide guarantees and can enable significant performance benefits, for example improving the throughput of hand-tuned CUDA programs by up to 2x.

S0157 - A Study of Persistent Threads Style Programming Model for GPU Computing
Kshitij Gupta (UC Davis), Jeff Stuart (UC Davis)
Day: Thursday, 05/17 | **Time:** 3:00 pm - 3:50 pm
Topic Areas: Parallel Programming Languages & Compilers; Audio, Image and Video Processing
Session Level: Advanced

We present the usefulness of a new style of GPU programming called Persistent Threads, known to be useful on irregular workloads. First, we will begin by formally defining the PT model. We will then categorize use of PT into four use cases, and present micro-benchmark analyses of when this model is useful over traditional kernel formulations. Third, we will show a full speech recognition application that uses all four PT use cases. Finally, we will conclude our talk by suggesting appropriate modifications to GPU hardware, software, and APIs that make PT

kernels both easier to implement and more efficient.

S0218 - ASI Parallel Fortran: A General-Purpose Fortran to GPU Translator
Rainald Lohner (George Mason University)
Day: Thursday, 05/17 | **Time:** 4:30 pm - 4:55 pm
Topic Areas: Development Tools & Libraries; Computational Fluid Dynamics; Computational Physics; Parallel Programming Languages & Compilers
Session Level: Advanced

Over the last 3 years we have developed a general-purpose Fortran to GPU translator: ASI Parallel Fortran does. The talk will detail its purpose, design layout and capabilities, and show how it is used and implemented. The use of ASI Parallel Fortran will be shown for large-scale CFD/CEM codes as well as other general purpose Fortran codes.