IPDPS 2025 Tutorials

To be held on June 3rd and 4th – the first two days of the conference – in parallel with the 20 IPDPS 2025 Workshops

The IPDPS 2025 format will be organized to hold 20 workshops on the first two days of the five-day conference. To enrich the offerings for those two days, the conference will also conduct 6 tutorials, open to all attendees. This will help to complement the one-day workshops, enabling attendees on the first two days of the conference to fill their day with workshops and tutorials covering a host of topics of interest to our community.

IPDPS 2025 **Tutorial 1 (half day)**

Title: Hands-On QubiC: Exploring an Open-Source Quantum Control System with Mid-Circuit Measurement and Feedforward Capabilities

Abstract: The NISQ (Noisy Intermediate-Scale Quantum) era of quantum computing demands flexible, cost-efficient classical control systems to meet the challenges of emerging quantum technologies. Developed at Lawrence Berkeley National Laboratory, QubiC is an open-source quantum control system designed to support mid-circuit measurement and feedforward for superconducting quantum computing. This tutorial introduces QubiC with an in-depth exploration of its architecture, functionalities, and capabilities in quantum control and measurement. We highlight the advanced control requirements for superconducting qubits and demonstrate how QubiC addresses these challenges through features such as fast parameter updates and real-time decision-making. Participants will gain hands-on experience with our simulator, performing pulse- and gate-level experiments to understand the system's practical applications. By the end of the session, attendees will have a comprehensive understanding of QubiC's role in advancing quantum control systems and the tools to experiment with its powerful capabilities.

Speakers: Abhi Rajagopala, Neelay Fruitwala, Anastasiia Butko, Yilun Xu, Gang Huang, and Kasra Nowrouzi, Lawrence Berkeley National Laboratory, USA

IPDPS 2025 **Tutorial 2 (half day)**

Title: Leveraging the IRON AI Engine API to program the Ryzen™ AI NPU

Abstract: The NPU of AMD Ryzen™ AI devices includes an AI Engine array comprised of a set of VLIW vector processors, data movement accelerators (DMAs) and adaptable interconnect. This tutorial is targeted at performance engineers who are looking to develop designs targeting the NPU with open source design tools. We provide a close-to-metal Python API: Interface Representation for hands-ON (IRON) AIE-array programming. IRON is an open access toolkit enabling performance engineers to build fast and efficient, often specialized, designs through a set of Python language bindings around the mlir-aie dialect. Participants will first get insight into the AI Engine compute and data movement capabilities. Through small design examples expressed in the IRON API and executed on an Ryzen™ AI device, participants will leverage AI Engine features for optimizing performance of increasingly complex designs. The labs will be done on Ryzen™ AI-enabled mini-PCs, giving participants the ability to execute their own designs on real hardware.

Speakers: Kristof Denolf and Josef Melber, AMD

IPDPS 2025 **Tutorial 3**
Part 1 on Tues (half-day) and Part 2 on Wed (half day)

Title: The Structural Simulation Toolkit (SST)

Abstract: As the science of high-performance computing (HPC) evolves, there is a growing need to understand and quantify the performance and compositional value of emerging technologies. Modeling and simulation techniques are well positioned to serve this purpose. The Structural Simulation Toolkit (SST) is a parallel discrete event-driven simulation framework that provides tools to enable co-design of HPC systems -- from application to architecture. Tutorial participants will be introduced to key facets of conducting reproducible simulations of HPC architectures and infrastructures. This tutorial will be split into two self-contained sections: one focused on node-level modeling and one on system-level modeling. Attendees will not need to attend one to benefit from the other. As novel compute architectures gain prominence in the evolving landscape of high-performance computing, tools that enable accurate modeling and simulation of these technologies are crucial for advancing both research and design. The balar GPU component, integrated into SST, provides scalable, trace- and execution-driven simulations of GPU systems by leveraging GPGPU-Sim. Similarly, Tactical Computing Labs' (TCL) Rev RISC-V component and the native SST vanadis CPU simulator extend SST's capabilities, allowing for detailed exploration of emerging CPU architectures. Together, these tools empower researchers to model and evaluate both CPU- and GPU-centric systems with unprecedented flexibility and accuracy. Participants will gain insights into the integration and use of balar for GPU performance modeling, as well as the Rev RISC-V component for node-level simulations. Unlike device-level simulation, the scale of modern distributed systems makes full detailed simulation of relevant distributed applications impractical. SST provides two workload modeling environments that allow researchers to create motifs/ skeletons that can be used to drive network simulations and as scaffolding to incorporate more detailed simulations. The Ember environment is the mainstay for workload modeling within SST and has underpinned many large-scale system studies using SST. Mercury is a new environment within SST, based on SST/macro, which allows an approach more closely resembling direct execution. In this section of the tutorial we will examine the use of both workload modeling environments in the context of system-scale network simulations utilizing the Merlin network components.

Speakers: Joe Kenny, Scott Hemmert, Clay Hughes, and Gwen Voskuilen, Sandia National Laboratories, USA; John Leidel, Tactical Computing Laboratories, USA

IPDPS 2025 **Tutorial 4 (half day)**

Title: Strategies for Large-Scale Data Analysis with the National Science Data Fabric (NSDF)

Abstract: This tutorial offers an intensive half-day session on the advanced uses of the National Science Data Fabric (NSDF) services and effective strategies for comprehensive scientific data analysis. Designed for researchers, students, developers, and scientists, it provides insights into managing and analyzing datasets exceeding 100TB. Participants engage in creating modular workflows, leveraging data storage and streaming, and deploying visualization and analysis dashboards for scientific purposes, with practical exercises highlighting NSDF's contributions to the HPDC conference's themes. The session covers NSDF's features, common challenges in data analysis, and intermediate exercises using NSDF for Earth science data, including handling high-resolution datasets. Attendees leave equipped with a deeper knowledge of integrating NSDF into research workflows, advancing data accessibility and promoting collaborative scientific discovery.

Speakers: Michela Taufer, University of Tennessee Knoxville; Valerio Pascucci, University of Utah; Jack Marquez, University of Tennessee Knoxville; Amy Gooch and Aashish Panta, University of Utah

IPDPS 2025 **Tutorial 5 (half day)**

Title: Productive, performance portable programming of HPC applications

Abstract: High-performance computing (HPC) is often perceived as a matter of making demanding applications running as fast as possible on large-scale systems, regardless the required programming effort. Not true: emerging large-scale applications are complex to develop and maintain, making programmability of HPC systems a major concern.
In this tutorial, we will overview programming techniques to facilitate the development of parallel applications, able to seamlessly scale from laptops, for fast prototyping, to large HPC systems for deployment. We will introduce two state-of-the-art productivity libraries, which orthogonally support high performance software development. We will discuss the Kokkos ecosystem, providing a hardware agnostic infrastructure for performance portable programming. We will then deep dive into the SHAD library, a collection of distributed containers and algorithms for scaling out on large HPC systems, with familiar interfaces and limited programming effort.

Speakers: Vito Giovanni Castellana (PNNL); Beatrice Branchini and Ian Di Dio Lavore (Politecnico di Milano); and Kokkos team (Sandia National Laboratories)

IPDPS 2025 **Tutorial 6 (half day)**

Title: Warp: A Python Multi-GPU Programming Model with CUDA Performance and Flexibility

Abstract: The rise of artificial intelligence has firmly established Python as the primary programming language for AI, and a similar trend is emerging in scientific applications. However, designing efficient and scalable applications in Python presents unique challenges, driving the development of innovative frameworks. For example, tools like JAX offer a tensor-based programming model inspired by NumPy, which can be mapped to multi-GPU execution. While these abstractions provide significant capabilities, they often obscure low-level details, limiting the user's ability to write optimal code. Tensor-based frameworks often struggle with efficient conditional logic, loops, and memory management. NVIDIA Warp introduces a new solution: a Python multi-GPU programming model with a CUDA-like interface. This approach empowers developers to write high-performance, fine-grained parallel code in Python with CUDA-level efficiency. Warp leverages just-in-time (JIT) compilation not only to directly generate and compile CUDA code but also to provide a differentiable programming interface, which is essential for optimization and AI workflows. In this tutorial, we will introduce the Warp multi-GPU programming model and demonstrate how to write high-performance, multi-GPU applications in Python. Beginning with simple Warp kernel development and profiling, we will progress to more advanced features such as automatic differentiability and multi-GPU programming.

Speakers: Lukasz Wawrzyniak, NVIDIA, USA and Massimiliano Meneghin, Autodesk, ITA