Invited talk #2:
Title: Python-native domain-specific compilation for low-level systems
Speaker: Tobias Grosser, University of Cambridge

Abstract:

Generating high-performance code for distributed low-level systems historically either required the use of low-level systems languages or the design of ad-hoc code generators. Specialized DSL frameworks sometimes facilitated these tasks and particular brave developers added native compiler support for their favorite features. However, in a perfect world, we would wish for the productivity of Python, the low-level control of assembly code, and the formal foundations of modern compiler technology. In this talk, I discuss how recent innovations in the LLVM community around MLIR, our work on Python-Native compiler development, and ongoing research on abstractions for MPI and RISC-V assembly code provide powerful tools that allow programmers of parallel architectures to combine control and productivity effectively.

Brief: CV:

Tobias Grosser is an Associate Professor at the University of Cambridge and an advocate for open-source-first research. Tobias co-founded the Polyhedral loop optimization framework Polly, the FPL Presburger Library for MLIR, the LLHD/CIRCT hardware-design compiler, and is regularly teaching compiler design using the Python-Native xDSL compiler project designed to lower the barrier of entry into the LLVM ecosystem. Tobias worked as a Google PhD Fellow at ENS Paris, an SNSF Ambizione Fellow at ETH Zurich, and a Reader at the University of Edinburgh.