

## Invited Talk

# Tapir: Embedding Fork-Join Parallelism into LLVM's Intermediate Representation

Tao B. Schardl

MIT Computer Science and Artificial Intelligence Laboratory  
32 Vassar Street  
Cambridge, MA 02139

**Abstract.** This talk explores how fork-join parallelism, as supported by dynamic-multithreading concurrency platforms such as Cilk and OpenMP, can be embedded into a compiler's intermediate representation (IR). Mainstream compilers typically treat parallel linguistic constructs as syntactic sugar for function calls into a parallel runtime. These calls prevent the compiler from performing optimizations across parallel control flow. Remediating this situation, however, is generally thought to require an extensive reworking of compiler analyses and code transformations to handle parallel semantics.

Tapir is a compiler IR that represents logically parallel tasks asymmetrically in the program's control flow graph. Tapir allows the compiler to optimize across parallel control flow with only minor changes to its existing analyses and code transformations. To prototype Tapir in the LLVM compiler, for example, we added or modified approximately 5000 lines of LLVM's approximately 3-million-line codebase. Tapir enables many traditional compiler optimizations for serial code, including loop-invariant-code motion, common-subexpression elimination, and tail-recursion elimination, to optimize across parallel control flow, as well as purely parallel optimizations.

This work was conducted in collaboration with William S. Moses and Charles E. Leiserson.