

To conclude, we would like to leave the reader with three interesting considerations regarding the nature of asymmetry in parallelism, the future of parallel optimizations, and extensions of Tapir-like systems to other models of parallel programming.

How the paper fits into broader context

Reasoning about logically parallel tasks asymmetrically based on serial semantics can sometimes simplify the understanding of a parallel program's behavior. When a task is spawned to execute in parallel with another, it is natural to reason about the logically parallel tasks as symmetric, because their instructions can execute in any relative order. For parallel programs with serial semantics, however, it is always valid to execute the program on a single processor, which asymmetrically executes one parallel task to completion before starting the other. Serial semantics encourage an asymmetric representation of parallel control flow that is similar enough to its serial elision that most common analyses and transformations for serial programs work on parallel constructs with little or no modification. In particular, serial semantics enables common optimizations on parallel code that can be invalid under other models of parallelism [71].

Relation to other results / models

One of the great benefits of Tapir is that its strategy for representing parallelism makes it easy to write optimization passes specifically for parallel code. Section 4 briefly mentioned some parallel optimization passes we implemented, including parallel-loop scheduling and unnecessary-sync elimination. In addition to helping close the performance gap between serial and parallel versions of code, we hope that the introduction of Tapir will encourage the development and implementation of many more parallel-optimization passes.

Implications of the work

Finally, Tapir allows fork-join parallel programs to benefit from both serial and parallel optimizations. Moving forwards, it is natural to wonder whether other models of parallelism, such as pipeline parallelism [13, 33, 49] or datagraph computations [43–45, 51, 52, 62, 64], can take advantage of the Tapir approach.

Future work / Open questions