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

William S. Moses  Tao B. Schardl  Charles E. Leiserson

EECSCon
April 18, 2017
What Is a Compiler?

Programs -> Compiler -> Physical Machine
A Good Compiler Does Wonders

No Optimization

Optimization

Minutes vs Hours!
Compilers Don’t Understand Parallel Code

What’s that?

cilk_for (int i = 0; i < n; ++i) {
    out[i] = in[i] / norm(in, n);
}

#pragma omp parallel for
for (int i = 0; i < n; ++i) {
    out[i] = in[i] / norm(in, n);
}
Example: Normalizing a Vector

```c
for (int i = 0; i < n; ++i)
    out[i] = in[i] / norm(in, n);
```
Test: random vector, n = 64M.
Example: Normalizing a Vector

```c
for (int i = 0; i < n; ++i)
    out[i] = in[i] / norm(in, n);
```

Test: random vector, n = 64M.

Running time: 0.312 s
Idea: Run in Parallel!
Example: Normalizing a Vector in Parallel

cilk_for (int i = 0; i < n; ++i) 
    out[i] = in[i] / norm(in, n);

Test: random vector, n = 64M.

A parallel loop replaces the original serial loop.

Serial running time: 0.3 seconds
Example: Normalizing a Vector in Parallel

```
cilk_for (int i = 0; i < n; ++i)
    out[i] = in[i] / norm(in, n);
```

A parallel loop replaces the original serial loop.

Test: random vector, n = 64M.

Serial running time: 0.3 seconds

18-core running time: 3 minutes
What do we do now?
Tapir: Task-based Asymmetric Parallel IR

💡 Let’s teach the compiler about parallelism!

This animal is called a Tapir
Parallel IR: A Bad Idea?

From “[LLVMdev] LLVM Parallel IR,” 2015:

- “[I]ntroducing [parallelism] into a so far ‘sequential’ IR will cause severe breakage and headaches.”
- “[P]arallelism is invasive by nature and would have to influence most optimizations.”
- “[It] is not an easy problem.”
What is Tapir

- Allows the compiler to represent parallelism as a natural *extension to serial code*
- Existing optimization passes work on parallel code *without modification*
- Don’t need to change the entire 4-million-line compiler!
Tapir modifies only 0.1% of an existing compiler!
Example: Normalizing a Vector with Tapir

```cilk_for
(int i = 0; i < n; ++i)
    out[i] = in[i] / norm(in, n);
```

A parallel loop replaces the original serial loop.

Test: random vector, n = 64M.

Serial running time: 0.3 seconds
Example: Normalizing a Vector with Tapir

cilk_for (int i = 0; i < n; ++i)
    out[i] = in[i] / norm(in, n);

Test: random vector, n = 64M.

A parallel loop replaces the original serial loop.

Serial running time: 0.3 seconds
18-core running time: 0.081s
Benchmarking

- 20 benchmarks
- More efficient on 17
- Gains of 10 to 25% for a third
- Less than 2% difference when less efficient
Status of Tapir

❖ Available on GitHub!

```
git clone --recursive https://github.com/wsmoses/Tapir-Meta.git
```

❖ Ongoing development (bug fixes, new optimizations, etc).

❖ Last fall, alpha version was used by MIT’s 6.172

❖ Submitted to 2017 Conference on Principles and Practice of Parallel Programming

❖ 29 papers accepted of 132 submissions!
Status of Tapir

- Available on GitHub!
  git clone --recursive https://github.com/wsmoses/Tapir-Meta.git

- Ongoing development (bug fixes, new optimizations, etc).

- Last fall, alpha version was used by MIT’s 6.172

- Submitted to 2017 Conference on Principles and Practice of Parallel Programming
  - 29 papers accepted of 132 submissions!
  - Won Best Paper!
Status of Tapir

- Available on GitHub!
git clone -recursive https://github.com/wsmoses/Tapir-Meta.git

- Ongoing development (bug fixes, new optimizations, etc).

- Last fall, alpha version was used by MIT's 6.172

- Submitted to 2017 Conference on Principles and Practice of Parallel Programming
  - 29 papers accepted of 132 submissions!
  - Won Best Paper!
Any Questions?

Disclaimer: Not actually a Tapir