

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?



A Good Compiler Does Wonders



Compilers Don't Understand Parallel Code



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

#pragma omp parallel for
for (int i = 0; i < n; ++i) {
 out[i] = in[i] / norm(in, n);
}</pre>

Confused Compiler

Example: Normalizing a Vector

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

Test: random vector, n = 64M.

Example: Normalizing a Vector

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

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);</pre>

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);</pre>

Test: random vector, n = 64M.

A parallel loop replaces the original serial loop.

Serial running time: 0.3 seconds 18-core running time: 3 minutes

What do we do now?



Tapir: Task-based Asymmetric Parallel IR

Vertified the compiler about parallelism!



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 <u>extension to serial</u> <u>code</u>
- Existing optimization
 passes work on parallel
 code <u>without modification</u>
- 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);</pre>

Test: random vector, n = 64M.

A parallel loop replaces the original serial loop.

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);</pre>

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 <u>https://github.com/wsmoses/Tapir-Meta.git</u>
- 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 <u>https://github.com/wsmoses/Tapir-Meta.git</u>
- 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

- PROGRAMMENG comments other discussions [3] PROGRAMMENG comments of the optimizes parallel code "better than optimizes para * Available on GitHub! Ongoing development (hother discussions (a) new optimizations comments optimizations comments optimizer (news.mt.)
 Last fall, 2' modified Luve compiler optimizer (news.mt.)
 S- programments open-source of MIT's 6
- MIT says their modified LLVM complier open MIT says their modified LLVM complier submitted 2 months ago by amc22004

N comments and share is accepted of 132 submissions!

Won Best Paper!



Шіт

Any Questions?



Disclaimer: Not actually a Tapir