PERFORMANCE COMPARISON OF IREE-COMPILED MLIR OPERATORS AND HAND-TUNED CUDA KERNELS

Vedant Bhasin [*1] **Vibha Arramreddy** [*1]

ABSTRACT

Compare IREE compiled hardware specifc code for tensor operations like SGEMM and Conv with hand tuned CUDA kernels written with CUTLASS

URL: https://vibhaarramreddy.github.io/

1 MILESTONE REPORT

1.1 Project Schedule

The schedule now includes detailed half-week increments, with assigned tasks as follows:

- **December 1 - 4:** Implement convolution operator in CUDA (CUTLASS) and MLIR.
- **December 4 - 8:** Optimize convolution operator for GPU in both CUDA and MLIR, start implementing fused MHSA kernel if feasible.
- **December 8 - 11:** Benchmark convolution operator across input sizes on RTX 2080 and Tesla T4 GPUs, refine performance metrics.
- **December 11 - 15:** Analyze benchmark results, finalize report and graphs for the poster session.

1.2 Work Completed So Far

We have successfully completed a proof of concept with MLIR/IREE, demonstrating the compilation and execution of SGEMM and vector addition kernels. After evaluating the complexity of these operators, we decided to pivot towards implementing a more advanced operator. Currently, the focus is on convolution (conv2d), with a potential reach goal of implementing a fused Multi-Head Self-Attention (MHSA) kernel, depending on feasibility. The project has shifted from exclusively working with MLIR to comparing automatically generated IREE-compiled code with hand-tuned CUDA kernels.

1.3 Goals and Deliverables

1.3.1 Progress on Initial Goals

The original MVP of achieving MLIR-generated CPU matmul code has been completed. Current efforts are focused on GPU benchmarking and optimization, with plans to compare CUDA kernels and MLIR-compiled kernels.

1.3.2 Updated Goals for Poster Session

- Minimum viable product (MVP): Comparison of CUDA kernel and IREE-compiled MLIR convolution operator.
- Target: Benchmarking performance across GPUs (RTX 2080 and Tesla T4) with varying input sizes.
- Reach goal: Implementation and benchmarking of a fused MHSA operator.

1.4 Poster Session Plan

We plan to present graphs and analysis comparing the performance of IREE-compiled operators and hand-tuned CUDA kernels. The focus will be on performance metrics such as execution time, memory usage, and scalability across input sizes.

1.5 Issues and Concerns

The feasibility of implementing a fused MHSA kernel in MLIR is uncertain, particularly due to challenges with operators like softmax and dropout. The convolution operator seems achievable.

[*] Equal contribution [1] Carnegie Mellon University, Pittsburgh, PA, USA. Correspondence to: Vedant Bhasin <vedantbhasin@cmu.edu>, Vibha Arramreddy <varramre@andrew.cmu.edu)>.

Proceedings of the 5^{th} MLSys Conference, Santa Clara, CA, USA, 2024. Copyright 2024 by the author(s).