Automatic Synthesis for Co-Optimized Kernel Generation and Scheduling

Kiran Kumar Rajan Babu, Pavan Hegde

{krajanba, pavanh}@andrew.cmu.edu

https://github.com/PavanHegde/CMU_15745_Project

Overview

Frameworks for machine learning perform control flow analysis and subsequently schedule primitive kernel operations, such as a matrix multiplication and convolution, to run on hardware either sequentially or concurrently. Each operator is typically optimized in isolation. As a result, running two or more operators concurrently may yield suboptimal performance on account of physical resources on chip. For example, a kernel *mm()* for matrix multiplication may be optimized assuming it has the full hardware resource. When scheduled to run alongside another operator *X()*, *mm()* and *X()* both perform worse.

The goal of our project is to implement a flexible kernel generation pass as an extension to existing TVM frameworks. This is the first step in creating a co-optimized scheme. Later works may then run exhaustive or non-exhaustive searches across the design space to optimize performance when taking both scheduling and the kernel itself into consideration.

Preliminary metrics include performance of generated kernels and tentatively overhead of our pass extensions at compile-time (based on what tracking is available within the TVM framework).

For now, our goal is to implement automated kernel generation for a small handful of kernels (like matrix multiplication) with a limited set of hyper parameters. Including basic scheduling and exhaustive search falls under stretch goals.

Logistics

Table 1 includes a tentative schedule. Since TVM is a new framework to both Kiran and Pavan, as of now there is no clear, optimal division of labor.

Table 1: Tentative Project Schedule

Week 1 (April 5th-9th)	Familiarizing with TVM
Week 2 (April 12th-16th)	Start Work on Kernel Generation Extensions
Week 3 (April 19th-23rd)	Work on Kernel Generation Extensions
Week 4 (April 26th-30th)	Slack/Verification/Evaluation
Week 5 (April - May 5th)	Presentations

Literature Search

- https://tvm.apache.org/docs/tutorials/index.html#tensor-expression-and-schedules
- https://tvm.apache.org/docs/tutorials/index.html#optimize-tensor-operators

Resources

This project will leverage the Apache TVM compiler infrastructure, an end-end Machine Learning compiler framework for CPUs, GPUs and accelerators. Required hardware includes any PC for running the framework, and tentatively a GPU for testing the compiled code.

Getting Started

Thus far we have discussed the project idea and expected requirements and resources with Pratik Fegade, a PhD student with expertise in the field, and have started looking into the TVM framework documentation. Neither Kiran nor Pavan have prior experience with the TVM framework so preliminary efforts include going through the TVM tutorials and familiarization with the framework. Simultaneously, we shall be working with Pratik to solidify an approach for kernel generation in-line with research and industry expectations.