Automatic Synthesis for Co-Optimized Kernel Generation and Scheduling

Kiran Kumar Rajan Babu, Pavan Hegde

{krajanba, pavanh}@andrew.cmu.edu

Major Changes

On the whole, the goal of our project has not changed, however the scope has narrowed significantly. At this point, we would like to get basic code generation for a concurrent-operator kernel working for a multicore CPU. On account of problems with TVM environment setup, lack of hardware resources (i.e. GPUs), and most importantly time, work on a multicore CPU seems like the most feasible goal.

What Have You Accomplished So Far?

To date, we have completed (finally after many many environment bugs… surprise) setting up the TVM Hetero environment: a fork of the main Apache TVM. We were successfully able to run a few optimization passes via the opt_gemm.py script.

From here, we will try to run the generated code from our implementation and compare it with the existing baseline code that the tvm generates in runtime/ resource utilization / number of lines of code.

Meeting your Milestone

In retrospect, our original goal was somewhat ambitious considering the time constraints. We are behind schedule, however we fully expect to have a working example for a basic case of the pass in time. Pratik has been providing excellent feedback. We will try

Revised Schedule

Week 1 (April 5th-9th)	Familiarizing with TVM
Week 2 (April 12th-16th)	TVM and Environment Setup

Week 3 (April 19th-23rd)	Environment Setup and Work on Kernel Generation Extensions
Week 4 (April 26th-30th)	Work on Kernel Generation Extensions
Week 5 (April - May 5th)	Presentations

Resources Needed

At this stage, we do not believe we need extra software or hardware resources beyond our current TVM Hetero (branch/fork off TVM provided by Pratik) framework installation however we may benefit from standardized benchmarks (should they exist for this problem space).