A Static Analysis-based Cross-Architecture Performance Prediction Using Machine Learning

Submitted by: Sharanya Kamath (16CO140)

Summary:

Porting code from CPU to GPU is costly and time-consuming. The paper aims to address this problem using machine learning in a supervised setting, using solely the single-threaded source code of the program, without having to run or profile the code. The proposed algorithm relies solely on program properties collected using static analysis of the CPU source code and predicts whether the potential speed-up is above or below a given threshold.

Dataset: Datapoints were collected from widely known GPU benchmark suites. This made the data more biased towards the GPU suited code, so the authors added some ill-suited codes for GPU. They further augmented the data by adding or subtracting code from already existing codes.

Model: A Random Forest algorithm is used to construct the speed-up classifier. The RF model is an ensemble of 1000 decision trees, where each tree is constructed using a random subset of features and training data-points:

- Memory Coalescing
- Branch Divergence
- Kernel Size
- Available Parallelism
- Instruction Intensities

Results: The preliminary results show that this algorithm can achieve 94% accuracy in binary classification, in average, across different thresholds. Leave one out cross validation was used to evaluate the performance of the classifier. According to the confusion matrix it achieves 94% accuracy, 98% recall, 93% precision. The authors have also shown that the technique is also robust to the choice of the speed up cut-off and the GPU platform. Multi-class classification can also be done by discretizing the speedup value into more than 2 categories. But the accuracy drops for more classes as the amount of data associated per class reduces.

Questions:

1. Can the problem be treated as a regression problem instead of classification?
2. Can this problem be approached in a statistical manner instead of applying heuristic methods?