CS4130/5130 Project 4: Loop Cost

Due: Wednesday, Dec. 7 @ 5:00 PM

We will continue basing our project on the Low Level Virtual Machine (LLVM) compiler infrastructure. In this project you will implement a new analysis pass that calculates loop cost based on the model by McKinley et al. TOPLAS'96. Your implementation can be embedded into DependenceAnalysis.cpp.

1 Project Summary

LLVM has implemented a loop dependence analysis pass (see `$llvm/lib/Analysis/DependenceAnalysis.cpp`) which is based on Goff et al. PLDI'91. Our lecture notes are also based on this paper. Your work will need to use the dependence testing results to compute the cost of each loop. You will also need spatial reuse detection in Project 3.

2 Implementation

You only need to consider perfectly nested loops which only contains one basic block in the loop body. You can assume all loop bounds are known constants and loop steps are also constant. You can assume a 32KB cache with 64-byte cache lines.

2.1 Output

Direct your output to stderr. For each loop nest, print "Loop Nest" followed by a loop nest number that is based on your analysis processing order. Then in this loop nest, for each loop as the presumable innermost loop, print the loop control variable, reference groups and the cost of each group. You need to identify a leader for each reference group. When outputting a reference group, print the leader first.

2.2 Testing

You can use the examples in my lecture notes and the examples in the paper for testing. I will test your code using the attached cost.c.

3 Submission

Submit your source code for this new analysis pass and other source code if you make changes. Write a short description of your implementation. Attach your output for cost.c.