

## Project Preproposal

Project URL: <http://www-2.cs.cmu.edu/~jeffpang/compilers/>

## 1 Research Direction

**Motivation:** This proposal is motivated by the recognition that although profiling information can be highly beneficial to many compiler optimizations, it is relatively expensive to collect accurately and inexpensively enough to include instrumentation in production code, often limiting profiles to unrealistic or unpredictable workloads (e.g., Ball and Larus cited that efficient path profiling incurred an overhead of 31%). Anderson, *et al.* showed that *sampling* in a real production system can be used to obtain relative instruction counts and other profiling information with negligible overhead. Nonetheless, their statistics were primarily designed to be used by humans rather than compilers.

Given the high cost of human labor (when compared to automatic optimization) and the advent dynamic optimization systems that could benefit from more accurate online profiles, we ask the question: can we derive *enough* accurate information (e.g., path profiles) from *estimations* (e.g., sampled instruction counts) for compilers to exploit? We are encouraged by the success of statistical approximation in similar domains (e.g., network tomography, data streaming, page rank, etc.).

**Problem Definition:** The problem addressed in the project would be how to approximate path profiles from sampled instruction (node) profiles or edge profiles. More specifically, given the program control flow graph (CFG)  $G = (E, V)$  and the approximate counts of number of accesses on  $V$  or  $E$  (and possibly some additional information that is easy to collect), can we identify the hot paths?

**Initial Ideas:**

- **Graphical Model:** If we have the assessed counts from the edges on the control flow graph (CFG), we can easily compute the probability on each edge. Essentially, the CFG augmented with the edge assess probability can be viewed as a directed graphical model of the entire program.
- **Random walk:** Based on the model above, the program path can be generated by random walk from the entry node to the exit node. The path profile can be obtained by repeating the random walk process many times and recording the paths.

The transformation of the program CFG into a probabilistic graphical model enables a lot of possibilities in applying the data mining and machine learning techniques.

## 2 Project Goals

First, we would design a technique to estimate path profiles on sampled edge or instruction profiles exploiting domain specific knowledge, such as features of CFGs or other information that can be obtained in the sampling process (e.g., register values). We would implement our technique to generate estimated path profiles. Second, we would evaluate our technique by comparing the accuracy of our estimated path profiles to those generated by an exact path profiler, such as one based on Ball and Larus's technique. We would need to obtain sampled and exact path profiles for this step.