

Project Proposal  
Profiled-based hyperblock formation

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

## 1 Project Description

**Motivation:** Our project focuses on two parts: 1) construct the hyperblocks based on path profiles; 2) estimate the path profiles from edge profiles.

First part is based on the observation that grouping the entire hot path into the same hyperblock enables more optimization opportunities for the compiler. Second part 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.

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?

**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?

**Evaluation:** We plan to evaluate the success of our estimation heuristics by comparing the accuracy of the estimated paths with that obtained with an exact path profiler. To quantify the impact this has on path profiling optimizations, we plan to implement a simple path-profile based super-block formation optimization and compare the quality of resulting schedules on the c6x clustered architecture.

## 2 Logistics

**Schedule:**

Date	Jeff's Plan	Jimeng's Plan
3/27 - 4/2	Path profiling lit search, Learn HALT lib, c2dil interface	Heuristics lit search (in DB lit)
4/3 - 4/9	Implement c2dil interface with HALT profiling lib (path profiling, sampled instr+edge profiles)	Devise and implement initial path estimation heuristic (external to c2dil)
4/10 - 4/16	Study hyperblock formation in c2dil, plan modification for superblock formation, start implementing	Get some workloads; initial evaluation of path estimation heuristics (write glue scripts, etc.)
4/17 - 4/23	Finish superblock formation in c2dil based on some path profile	Refine path estimation heuristics (if needed); more initial evaluation
4/24 - 4/30	Glue together a scheduler (from class) with c2dil; evaluation + writing	Refine 2 or 3 basic experiments; evaluation + writing
5/1 - ???	More evaluation + writing (if time)	Ditto

**Milestone:** We plan to have the following complete by April 12th:

1. An implemented initial path estimation heuristic based on sampled instruction counts. If based on nothing else than our informed conjectures.
2. A version of c2dil instrumented to gather path profiles and be able to “simulate” instruction and edge count sampling (in actuality this would be done using an interrupt handler on a running program, like in Anderson, *et al.*, or architecture specific performance counters).

### Literature Search:

**INSTRUMENTED PATH PROFILING:** These papers focus on using instrumentation to perform path profiling and are mainly for background.

*T. Ball and J. Larus. Efficient Path Profiling. International Symposium on Microarchitecture, 1996, pp. 46-57.*

*R. Joshi, M. Bond, and C. Zilles. Targeted Path Profiling: Lower Overhead Path Profiling for Staged Dynamic Optimization Systems. Proceedings of the International Symposium on Code Generation and Optimization, 2004.*

**RUNTIME PATH PROFILING:** These papers attempt to leverage performance counters (like those on the Itanium) to sample branches taken at runtime and extrapolate profiles from them, and hence are most similar to our project. The first focuses on dynamic optimization and does not try to predict path profiles directly. The second uses sampled branch to extrapolate path profiles; our work will differ in that we plan to leverage known data-mining techniques to discover paths where there is ambiguity (their heuristics are conservative) and we target a clustered architecture (they target the Itanium). Moreover, they did not evaluate the impact their path profiles have on real optimizations, while we plan to.

*H. Chen, W. Hsu, J. Lu, P. Yew, and D. Chen. Dynamic trace selection using performance monitoring hardware sampling. Proceedings of the International Symposium on Code generation and optimization: feedback-directed and runtime optimization, 2003.*

*A. Shye, M. Iyer, T. Moseley, D. Hodgdon, D. Fay, V.J. Reddi, and D.A. Connors. Analysis of Path Profiling Information Generated with Performance Monitoring Hardware. Proceedings of the 9th Workshop on Interaction between Compilers and Computer Architecture, February 2005.*

**SUPERBLOCK FORMATION:** This paper describes the path-profiling optimization we plan to implement.

*W. Hwu, S. Mahlke, W. Chen, P. Chang, N. Warter, R. Bringmann, R. Ouellette, R. Hank, T. Kiyohara, G. Haab, J. Holm, and D. Lavery. The Superblock: An Effective Technique for VLIW and Superscalar Compilation. The Journal of Supercomputing, 1993, pp. 229-248.*

**PATH ESTIMATION:** These papers describe random-walk techniques on graph used in the data-mining community:

S. Brin and L. Page. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30(1-7):107-117, 1998.

T. Haveliwala. Topic-sensitive pagerank. In *Proceedings of the Eleventh International World Wide Web Conference*.

**Resources Needed:** The primary resource we will probably require is a helpful (preferably human) guide when delving into the parts of the c2dil compiler we have not already been familiarized with. Tim has already helped us with finding the HALT profiling library so the only major thing remaining is probably some tips about how the hyperblock formation currently works.

The other thing we need is a set of benchmarks to use as a workload, but we assume that most other groups will require some form of benchmarking for evaluation also so we will probably work on this together (?).

**Getting Started:** We have reviewed the literature related to our project and got started learning the internals of some of the libraries we will be using, like HALT.