University of Victoria | Graduate Studies

Notice of the Final Oral Examination
for the Degree of Doctor of Philosophy

of

IVAN NEIL BURROUGHS

MSc (University of Victoria, 2008)
BSc (University of Victoria, 2006)

"Register Allocation and Spilling Using the Expected Distance Heuristic"

Department of Computer Science

Thursday, January 28, 2016
2:30 P.M.
David Turpin Building
Room A136

Supervisory Committee:
Dr. Nigel Horspool, Department of Computer Science, University of Victoria (Supervisor)
Dr. Yvonne Coady, Department of Computer Science, UVic (Member)
Dr. Paul Lalonde, Software Engineer, Google Inc. (Additional Member)
Dr. Amirali Baniasadi, Department of Electrical and Computer Engineering, UVic (Outside Member)

External Examiner:
Dr. Jingling Xue, School of Computer Science and Engineering, University of New South Wales

Chair of Oral Examination:
Dr. Keivan Ahmadi, Department of Mechanical Engineering, UVic

Dr. David Capson, Dean, Faculty of Graduate Studies

Abstract

The primary goal of the register allocation phase in a compiler is to minimize register spills to memory. Spills, in the form of store and load instructions, affect execution time as the processor must wait for the slower memory system to respond. Deciding which registers to spill can benefit from execution frequency information yet when this information is available it is not fully utilized by modern register allocators.

We present a register allocator that fully exploits profiling information to minimize the runtime costs of spill instructions. We use the Furthest Next Use heuristic, informed by branch probability information to decide which virtual register to spill when required. We extend this heuristic, which under the right conditions can lead to the minimum number of spills, to the control flow graph by computing Expected Distance to next use.

The furthest next use heuristic, when applied to the control flow graph, only partially determines the best placement of spill instructions. We present an algorithm for optimizing spill instruction placement in the graph that uses block frequency information to minimize execution costs. Our algorithm quickly finds the best placements for spill instructions using a novel method for solving placement problems.

We evaluate our allocator using both static and dynamic profiling information for the SPEC CINT2000 benchmark and compare it to the LLVM allocator. Targeting the ARMv7 architecture, we find average reductions in numbers of store and load instructions of 36% and 50%, respectively, using static profiling and 52% and 52% using dynamic profiling. We have also seen an overall improvement in benchmark speed.