Implementation of Additional Optimizations for VPO on the Power Platform

Chungsoo Lim, Gary A. Smith, Won So
{clim, gasmith, wso}@ncsu.edu
http://www4.ncsu.edu/~gasmith/csc791a_proj

Spring 2005 CSC791A Project Proposal

Introduction

The Very Portable Optimizer is a compiler tool for machine-independent program optimization. Optimizations are performed at the generalized Register Transfer List level so that any machine platform can be targeted. During this process, VPO utilizes a checking routine called "gatekeeper" to ensure that optimizations will translate to instructions for a given machine platform. VPO may make any necessary optimization changes; however, each post-optimization RTL must map to at least one machine instruction. VPO is a member of the Zephyr Compiler Infrastructure toolset [Appel98] developed at the University of Virginia.

VPO is generally utilized on the Sun SPARC platform; however, it can target any machine for which a back-end recognizer and asm emitter has been written. This project will explore adding or repairing optimizations for an existing Power-targeted VPO implementation [IBMPwr4, IBMRdBk].

Existing Optimization State

We performed experiments in order to discover which optimizations function correctly and incorrectly. The items in the existing compiler test suite were divided amongst our team members. Each team member executed tests to discover which optimizations are functional in the compiler. The following table summarizes our findings.

Opt	Current Status	Comments
L	OK	Control-flow Optimizations
O	Seg Faults, Output Differences	Local Register Allocation
G	OK	Global Linking
V	Compile Does not Finish	Evaluation Order Determination
C	Basic Block not Examined Message, Not Enough Link Available, No Output Differences	CSE
M	Depends on O or R -VOM, -VRM, All Errors Identical to -VO case	Code Motion
N	OK	Fix Up Control Flow
S	Various Seg Faults, Memory Faults, and Differences	Strength Reduction
B	Only Appears in vpo.c Comments	Induction Variable Elimination
R	Irreducible Flow Graph in Setup, Signal	Global Register Allocation

Table 1: Current Status of Power-VPO Optimizations

Our Preferred Optimizations

We would prefer to implement the following optimizations. Each optimization will be implemented by an individual team member. Team members will assist each other when technical difficulties are encountered. In particular, we are interested in data flow optimizations.

Optimization	Comments
Live Variable Analysis	
CSE	Numerous Warnings, See Table 1
Evaluation Order Determination	

Table 2: Optimizations We Would Prefer

Plan for Enabling Optimizations

1. Check source code status of assigned optimizations.
    a. Optimization might have existing source code.
    b. Optimization may be functional, but contain errors.
    c. Optimization may not be implemented.
2. Eliminate code bugs from step one or implement functions as appropriate. Divide workload among team members.
3. Verify that optimization is functional by using existing test suite.
4. Compare RTL before and optimization to ensure that desired functionality is present.

Disclaimer

This paper and associated software changes are intended for **informational purposes only**. Therefore, any use of the information presented in this student work is at your own risk. Chungsoo Lim, Gary A. Smith, and Won So provide **no warranties of any kind** surrounding the use of this material.

References

[Appel98] A. Appel, J. Davidson, and N. Ramsey. "The Zephyr Compiler Infrastructure." *Proceedings of Supercomputing '98*, 1998.

[IBMPwr4] IBM Corporation. Power4 System Microarchitecture White Paper. http://www-1.ibm.com/servers/eserver/pseries/hardware/whitepapers/power4.html

[IBMRdBk] IBM Corporation. The POWER4 Processor Introduction and Tuning Guide. http://www.redbooks.ibm.com/redbooks/pdfs/sg247041.pdf