Open Source LLVM-VPO Compiler

Ryan Baird[1], Brandon Davis[2], Dr. Gang-Ryung Uh[1], Dr. David Whalley[2]
Boise State University[1] and Florida State University[2]

Project Overview

Media applications are becoming more complex; mobile devices are being built with low-power multi-issue mobile processors to manage increasing data and instruction flow. We're developing an Open Source Optimizing compiler from the **LLVM** [2] and **Zephyr** [3] Compiler frameworks so researchers can more easily implement code transformations for low-power Processors.

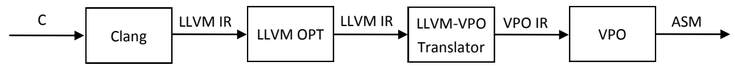
Motivation

VPO w/ LCC	LLVM w/ Clang
• Preferred for machine level Optimizations	• Preferred for high level Optimizations
• No Support for new language standards	• Support for new language standards
• Difficult to implement code-expander for new architectures	• Difficult to add support for new architectures

Project

Previous Attempt	Current Version
• LLVM Module	• LLVM Target
• Easy custom code	• Uses LLC with Optimization
• Performance Loss	• Performance Loss

Compilation Process:



VPO IR Example

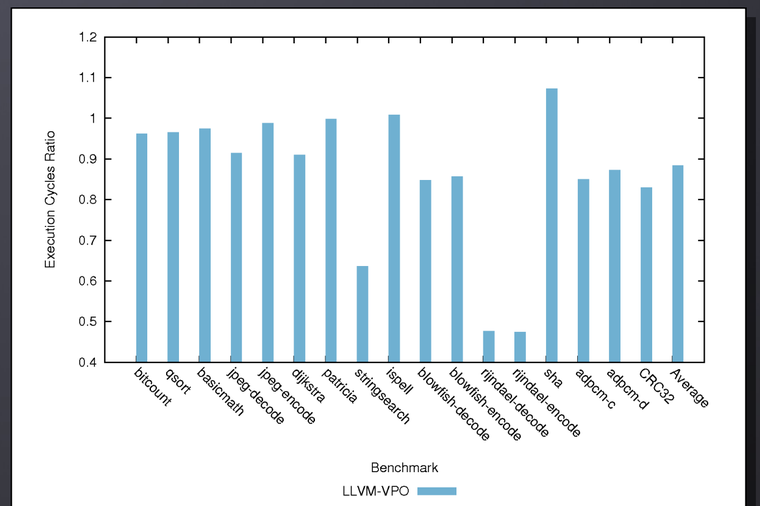
File Header:

Mrfdhsu m registerSize=4 m globalSize = 4 m localSize=4	Architecture Information
-.globl main dmain GLO[1] 0 0	Global Declarations

Function Definitions:

-main: fmain	Function Definition
Dl0.0_a LOC[0] 2 0 4 0 0	Parameter/Local Definitions
+r[25]=GLO[2] ur[25] A[4] +ST=r[25] …	Function Calls
+r[2]=0 ur[2] +PC=RT	Return
s=r[2]; * -.end main	Function End

Current Results



The figure above compares the ratio in execution cycles of our VPO-LLVM compiler to VPO with LCC. On the MiBench Test Suite. Numbers less than 1 indicate performance improvement. The average performance improvement is above 10%.

Future Goals

- Build a version for ARM
- Build a version for Static Pipeline Architecture [1]
- Fix known performance issues
- Improve Implementation
  - Use Pseudo-Registers
  - Generate Dead Register Lists
  - Simplify target change process
- Handle More Test Cases

Open Source

The current version of the project is available for SVN checkout. For more information about downloading this project, visit http://cs.boisestate.edu/~uh/LLVMVPO.htm

Project funded by: Google Faculty Research Awards, Korea SMBA grant 0004537, and KEIT grant 10041725.

1. I. Finlayson, B. Davis, P. Gavin, G. Uh, D. Whalley, M. Sjalander, and G. Tyson: Improving Processor Efficiency by Statically Pipelining Instructions. In ACM LCTES, 2013.
2. C. Lattner and V. Adve: LLVM: A Compilation Framework for Lifelong Program Analysis & Transformation. In Proceedings of the International Symposium on Code Generation and Optimization, March, 2004.
3. M. Benitez and J. Davidson: A Portable Global Optimizer and Linker. In Proceedings of the SIGPLAN'88 Conference on Programming Language Design and Implementation, June, 1988.