Uppsala University

Bachelor's Thesis

Static Multi-Versioning for Efficient Prefetching

Author:
Per Ekemark

Supervisor:
Alexandra Jimborean

Examiner:
Olle Gällmo

Reviewer:
David Black-Schaffer

A thesis submitted in fulfilment of the requirements for the degree of Bachelor of Computer Science

in the

Research Group Name
Department of Information Technology

April 2014

UPPSALA UNIVERSITY

Abstract

Faculty for Science and Technology
Department of Information Technology

Bachelor of Computer Science

Static Multi-Versioning for Efficient Prefetching

by Per Ekemark

The Thesis Abstract is written here (and usually kept to just this page). The page is kept centered vertically so can expand into the blank space above the title too...

Contents

List of Figures

List of Tables

Abbreviations

LAH	**L**ist **A**bbreviations **H**ere

Physical Constants

Speed of Light $c = 2.997\,924\,58 \times 10^{8}\ \mathrm{ms}^{-\mathrm{S}}$ (exact)

Symbols

a	distance	m
P	power	W (Js^{-1})
ω	angular frequency	rads^{-1}

Chapter 1

Introduction

General introduction...

1.1 Background

- Decreased power consumption through DVFS
  - Power equation
  - Breakdown of Dennard scaling
- Decoupled access/execute
  - Finding tasks suitable for optimisation
  - Breaking them down in small enough chunks
  - Creating the access version

1.2 Approach

- Multiple access versions
  The idea is to generate more than one access version. These are to be generated from different rule sets. The generated access versions are evaluated in a runtime system in order to determine which version that produces the best results for a specific task. That version is selected to be used for that task while others may use different access versions if they prove to be better.

Chapter 2

Related Work

- Fix the code [1]
- Towards More Efficient Execution [2]
- Polyhedral model and affine code
  (Not strictly required for this thesis.)

Chapter 3

Methodology

- Environment setup
  - LLVM 3.4
  - SPEC CPU2006
- Access version generation techniques and rules
- Evaluation model
  - Power model
  - Required assumptions

Chapter 4

Experiments

... or Evaluation

Chapter 5

Conclusion

Appendix A

Appendix Title Here

Write your Appendix content here.

Bibliography

[1] A. Jimborean, K. Koukos, V. Spiliopoulos, D. Black-Schaffer, and S. Kaxiras. Fix the code. don't tweak the hardware: A new compiler approach to voltage-frequency scaling. In *GCO'14*, February 2014.

[2] K. Koukos, D. Black-Schaffer, V. Spiliopoulos, and S. Kaxiras. Towards more efficient execution: A decoupled access-execute approach. In *ICS'13*, June 2013.