

Contents

1	Background	1
1.1	Introduction	1
1.2	Compiler Optimizations	4
1.2.1	A Note on Terminology and Metrics	4
1.2.2	Compiler Optimization Benefits and Challenges	5
1.2.3	Compiler Optimization Problems	5
1.3	Machine Learning Models	8
1.3.1	Supervised Learning	8
1.3.2	Unsupervised Learning	13
1.3.3	Other Machine Learning Methods	14
1.4	Conclusions	16
	References	16
2	Design Space Exploration of Compiler Passes: A Co-Exploration Approach for the Embedded Domain	23
2.1	VLIW	23
2.2	Background	25
2.3	Methodology for Compiler Analysis of Customized VLIW Architectures	26
2.3.1	Custom VLIW Architecture Selection	28
2.3.2	Compiler Transformation Statistical Effect Analysis	30
2.4	Conclusions and Future Work	38
	References	38
3	Selecting the Best Compiler Optimizations: A Bayesian Network Approach	41
3.1	Introduction	41
3.2	Previous Work	43
3.3	Proposed Methodology	44

- 3.3.1 Applying Program Characterization 46
- 3.3.2 Dimension-Reduction Techniques 47
- 3.3.3 Bayesian Networks 49
- 3.4 Experimental Evaluation 52
 - 3.4.1 Benchmark Suites 52
 - 3.4.2 Compiler Transformations 54
 - 3.4.3 Bayesian Network Results 55
 - 3.4.4 Comparison Results 58
 - 3.4.5 A Practical Usage Assessment 63
 - 3.4.6 Comparison with State-of-the-Art Techniques 65
- 3.5 Conclusions 68
- References 68
- 4 The Phase-Ordering Problem: An Intermediate Speedup Prediction Approach 71**
 - 4.1 Introduction 71
 - 4.2 Related Work 73
 - 4.3 The Proposed Methodology 74
 - 4.3.1 Compiler Phase-Ordering Problem 76
 - 4.3.2 Application Characterization 76
 - 4.3.3 Intermediate Speedup Prediction 77
 - 4.4 Experimental Evaluation 79
 - 4.5 Conclusions 82
 - References 82
- 5 The Phase-Ordering Problem: A Complete Sequence Prediction Approach 85**
 - 5.1 Intermediate Versus Full-Sequence Speedup Prediction 85
 - 5.2 Related Work 87
 - 5.3 The Proposed Methodology 88
 - 5.3.1 Application Characterization 90
 - 5.3.2 Constructing Compiler Sub-sequences 90
 - 5.3.3 The Proposed Mapper 93
 - 5.3.4 Predictive Modeling 94
 - 5.3.5 Recommender Systems Heuristic 97
 - 5.4 Experimental Results 98
 - 5.4.1 Analysis of Longer Sequence Length 101
 - 5.4.2 MiCOMP Prediction Accuracy 102
 - 5.4.3 MiCOMP Versus the Ranking Approach 104
 - 5.5 Comparative Results 105
 - 5.5.1 Comparison with Standard Optimization Levels 105
 - 5.5.2 Comparisons with State-of-the-Art Models 107
 - 5.5.3 Comparison with Random Iterative Compilation 109

5.6 Conclusion	111
References	112
6 Concluding Remarks	115
6.1 Main Contributions	115
6.2 Open Issues and Future Directions.	116
Index	117



<http://www.springer.com/978-3-319-71488-2>

Automatic Tuning of Compilers Using Machine Learning

Ashouri, A.H.; Palermo, G.; Cavazos, J.; Silvano, C.

2018, XVII, 118 p. 23 illus., 6 illus. in color., Softcover

ISBN: 978-3-319-71488-2