A Predictive Modeling Framework For Compiler Phase-ordering Problem

Amir Hossein Ashouri, Andrea Bignoli, Gianluca Palermo and Cristina Silvano
Politecnico Di Milano, ITALY
name.family-name@polimi.it

ABSTRACT

Today's compilers offer a huge number of transformation optimizations to choose among and this choice can significantly impact on the performance of the code being optimized. Not only the selection of compiler optimizations represent a hard problem to be solved [2, 3], but also finding the best ordering can add further complexity, making it a long standing problem in the compilation research [1, 5]. Classic predictive modelings simply can not cope with the enormous complexity of the optimizations within a sequence. This paper proposes a novel autotuning framework *i)* to dynamically explore and characterize the applications and ii) to predict the best compiler optimizations to be applied in order to maximize the performance. The framework also presents a mapping technique capable of transforming any representation of compiler optimizations vector in the phase-ordering space including those having repetitions into a new binary representation that is classified under the problem of selection of compiler optimization. This way the compiler researchers can benefit from exploiting fix-feature vectors for the predictive modelings. Experimental results using the latest LLVM compiler framework and cBench [4] suite have shown effectiveness of the mapping technique by utilizing a number of predictive modelings. We show statistical analysis over the distribution of the data and the quality comparison with respect to *Random Iterative Compilation* model and *standard optimization* levels -O2 and -O3.

Categories and Subject Descriptors

D.3.4 [**Software**]: Programming Languages—*Compilers*

Keywords

Predictive Modeling, Autotuning, Compilers, Phase-ordering, Machine Learning

1. ACKNOWLEDGMENTS

This work is partially funded by the European Union's Horizon 2020 research and innovation programme under grant agreement ANTAREX-671623.

CGO'16, March 12-16, Barcelone, Spain.

2. REFERENCES

[1] Ashouri, A. H., Bignoli, A., Palermo, G., and Silvano, C. Predictive modeling methodology for compiler phase-ordering. In *Proceedings of 7th Workshop on Parallel Programming and Run-Time Management Techniques for Many-core Architectures and 5th Workshop on Design Tools and Architectures for Multicore Embedded Computing Platforms* (2016), ACM.

[2] Ashouri, A. H., Mariani, G., Palermo, G., and Silvano, C. A bayesian network approach for compiler auto-tuning for embedded processors. In *Embedded Systems for Real-time Multimedia (ESTIMedia), 2014 IEEE 12th Symposium on* (2014), IEEE, pp. 90–97.

[3] Ashouri, A. H., Zaccaria, V., Xydis, S., Palermo, G., and Silvano, C. A framework for compiler level statistical analysis over customized vliw architecture. In *Very Large Scale Integration (VLSI-SoC), 2013 IFIP/IEEE 21st International Conference on* (2013), IEEE, pp. 124–129.

[4] Fursin, G. Collective benchmark (cbench), a collection of open-source programs with multiple datasets assembled by the community to enable realistic benchmarking and research on program and architecture optimization, 2010.

[5] Kulkarni, S., and Cavazos, J. Mitigating the compiler optimization phase-ordering problem using machine learning. *ACM SIGPLAN Notices 47*, 10 (2012), 147–162.