CCA/EBT: Code Comprehension Assistance Tool for Evidence-Based Performance Tuning

M. Hashimoto\textsuperscript{*}, M. Terai\textsuperscript{*}, T. Maeda\textsuperscript{\dagger}, and K. Minami\textsuperscript{*}
\textsuperscript{*}Software Technology and AI Research Lab, Chiba Institute of Technology
\textsuperscript{\dagger}RIKEN Advanced Institute for Computational Science

INTRODUCTION

Application performance tuning is still quite an art, despite advances in auto-tuning systems [1-9].

EBT (evidence-based performance tuning) [5] aims at helping performance engineers gain and share evidence of performance improvement to make better decisions.

Long-term goal is to construct a database of facts, or factbase, extracted from performance tuning histories of computational kernels such that we can search the database for promising optimization patterns that fit a given computational kernel.

OBJECTIVES

• Locating computational kernels
• Predicting location of computational kernels
• Assisting in the manual inspection of source code
• Identifying optimization patterns applied to computational kernels
• Constructing database of positive/negative examples of optimization patterns

TECHNICAL HIGHLIGHTS

Loop Kernel Prediction based on Machine Learning

1) Features were extracted from 175,963 loops from 1000 computation-intensive applications hosted on GitHub [6].
2) 100 were randomly sampled and then manually classified by experienced performance engineers.
3) By using the classification results as training data and C-SVC in LIB-SVM [3] from scikit-learn, we constructed a predictive model.
4) The model achieved 20% classification accuracy of 81% [6].

Dedicated Fortran Parser

Specifications: FORTRAN77, Fortran90, Fortran95, Fortran2003, Fortran2008

Diactes: IBM, PGI, Intel

Directives: Cpp, OpenMP, OpenACC, OCL(Fujitsu), XLF(IBM), DIR/DEC(Intel)

Schemes for Statically Estimating Volume of Memory Traffic

E0 Data is shared in cache only among syntactically identical array references.
E1 The data referenced by the array references that differ only by the first dimension are located in the same cache block. (ex. a(i, n) and a(j, n))
E2 The data referenced by the array references that differ only by the first dimension and by additions/subtractions of constants at the second dimension are located in the same cache block. (ex. a(i, n) and a(j, n + 1))

Topic Analysis for Source Code

• Helping performance engineers understand an application
• Analyzing comments and variable names occurring in the source code
• Examining the topic or research field of the application
• Constructing a topic model with latent semantic indexing [50] [4]
• Based on 168 papers of scientific applications from several research fields
  - Quantum chemistry, astrophysics, climate science, ...

RELATED WORK

Commercial and open-source Fortran analysis tools include the following:
• FORCHECK [2] — A Fortran source code analyzer and programming aid,
• Photran [7] — An IDE and refactoring tool for Fortran, and
• CamFort [8] — Light-weight verification and transformation tools for Fortran. CCA/EBT is capable of predicting loop kernels and of parsing 1000 applications in a fully automated way.

REFERENCES

2. Forcheck b.v. \texttt{http://www.forcheck.nl/}

ACKNOWLEDGMENTS

This work was supported in part by JSPS KAKENHI Grant Number JP26540031.