

## Transformations for High-Performance Computing

Jonice Oliveira\*

Department of Computer Science, Federal University of Rio de Janeiro, Rio de Janeiro, Brazil.

### ABSTRACT

In the last three decades a large number of compiler transformations for optimizing programs have been implemented. Most optimizations for uniprocessors reduce the number of instructions executed by the program using transformations based on the analysis of scalar quantities and data-flow techniques. In contrast, optimizations for high-performance superscalar, vector, and parallel processors maximize parallelism and memory locality with transformations that rely on tracking the properties of arrays using loop dependence analysis.

This survey is a comprehensive overview of the important high-level program restructuring techniques for imperative languages such as C and Fortran. Transformations for both sequential and various types of parallel architectures are covered in depth. We describe the purpose of each transformation, explain how to determine if it is legal, and give an example of its application.

**Keywords:** Transformation; Data-flow techniques; Internet; Optimizations

### INTRODUCTION

Programmers wishing to enhance the performance of their code can use this survey to improve their understanding of the optimizations that compilers can perform, or as a reference for techniques to be applied manually. Students can obtain an overview of optimizing compiler technology. Compiler writers can use this survey as a reference for most of the important optimizations developed to date, and as bibliographic reference for the details of each optimization. Readers are expected to be familiar with modern computer architecture and basic program compilation techniques. Cloud computing empowered Internet of Things (IoT) technology has conceptualized the ideology of Industry 4.0. Inspired by this, the food industry 4.0 presents a unique concept for determining food quality in real-time. Conspicuously, the current research provides an IoT-based smart framework for evaluating the food-quality parameters in restaurants and food outlets. IoT technology is primarily utilized to gather data that can explicitly affect food quality within a food serving environment. Such data is analyzed using the Bayesian Modeling Technique on the Fog-Cloud platform to derive a unanimous metric in terms of Probability of Food Grade (PoFG). Also, Food Grade Assessment Scale (FGAS) is quantified to assess real-time food-oriented parameters in the ambient environment of food-outlets and restaurants.

### TRANSFORMATION ISSUES

Decide upon a part of the program to optimize and a particular transformation to apply to it. Verify that the transformation either does not change the meaning of the program or changes it in a restricted way that is acceptable to the user. Transform the program. In this survey we concentrate on the last step: transformation of the program. On determinism can be introduced by language constructs or by calls to system or library routines that return information about the state external to the program (such as Unix `time( )` or `read()`). In some cases it is straightforward to cause executions to be identical, for instance by entering the same inputs, In other cases there may be support for determinism in the programming environment for instance, a compilation option that forces select statements to evaluate their guards in a deterministic order. As a last resort, the programmer may have to replace nondeterministic system calls temporarily with deterministic operations in order to compare two versions.

### CONCLUSION

Optimizing compilers have become an essential component of modern high-performance computer systems. In addition to translating the input program into machine language, they analyze it and apply various transformations to it to reduce its

**Correspondence to:** Jonice Oliveira, Department of Computer Science, Federal University of Rio de Janeiro, Rio de Janeiro, Brazil, E-mail: joniliveira@gmail.com

**Received:** January 28, 2021; **Accepted:** February 11, 2021; **Published:** February 18, 2021

**Citation:** Oliveira J (2021) Transformations for High-Performance Computing. J Inform Tech Softw Eng. S2:002.

**Copyright:** © 2021 Oliveira J. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

running time or its size. As optimizing compilers become more effective, programmers can become less concerned about the details of the underlying machine architecture and can employ

higher-level, more succinct, and more intuitive programming constructs and program organizations.