Compiling and optimization for high-performance systems

Roel Jordans

Abstract

The increasingly specialized processors utilized in high-performance systems have resulted in an extremely complex programming environment. The resulting differences in processor architectures, memory system organization, programming model are impacting the programmability of these systems. Expecially their respective interactions are resulting in confusion as, even for experienced programmers, it becomes very difficult to keep track of the inner workings of the compilation process.

In this presentation we will provide you with the basic knowledge required to understand the compilation process for such high-performance systems. We will discuss how the compiler processes your code, how this code is translated into an executable file, and the kind of optimizations that the compiler is able to apply to the program. This presentation will be separated into two parts; the overall compiler organization, and the generic optimization approach. In the first part of this presentation we will introduce the basic organization of the compilation process and illustrate how this has been extended towards heterogeneous architectures. We will introduce the main steps to the compilation process, the internal representation of the compiled program, the binary object file format, and the combination of (many) object files into the final executable during linking. The second part will focus more on the considerations taken by a compiler when optimizing your code. We will take auto-vectorization as an example and demonstrate how, and when, the compiler can apply this optimization, how we can help the compiler to recognize the available optimization oportunities, and how to guide it to enable high-performance execution of the program.

About the presenter

Roel Jordans received both the MSc and PhD degrees in field of Electrical Engineering from Eindhoven University of Technology in 2009 and 2015 respectively. He worked within the PreMaDoNA project on the MAMPS tool flow as a researcher afterwards. His dissertation focussed on the automatic design space exploration of VLIW ASIP instruction-set architecture within the ASAM project. His research interest include compilers and compilation techniques for application specific systems, digital signal processing systens based on customized VLIW architectures, and reliability and fault-tolerant design for space applications. Currently he is employed at the Radboud University Nijmegen where he is active as science DSP architect in the Radboud Radio Lab, working on the software-defined radio astronomy receiver for the Netherlands-China Low-frequency Explorer (NCLE) mission. In parallel he is the primary lecturer of the parallelization, compilation, and platforms course at Eindhoven University of Technology. He also serves as a program committee member for the EUROMICRO Symposium on Digital System Design.