

Challenges and Opportunities in C/C++ Source-to-Source Compilation

Location: Room **Servanty**

Scheduled time: **16:00 – 16:45**

Speaker: João Bispo

João Bispo is an assistant professor at the Faculty of Engineering, University of Porto (FEUP) and coordinator of the SPeCS research group in the Computing Systems Lab. In 2012 received the Ph.D. degree from Instituto Superior Técnico (IST), Lisbon, with a thesis about automatic runtime migration of loops found in assembly traces to customized hardware.

During the past few years his work has focused on source-to-source compilation and on delivering efficient computation on specialized targets, such as GPUs and FPGAs. In 2012 he worked on multi-target compilation of MATLAB to C/OpenCL, and in 2015 he was the University of Porto team leader in the European H2020 project ANTAREX, which focused on energy efficiency in an HPC context. During this project he was responsible for developing Clava, a C/C++ source-to-source compiler, which was used to implement and explore several code analyses and optimizations.

Abstract

Contributing to the lower-levels of the compilation stack (e.g., IRs, compilation passes, and targeting specific backends) via the traditional compiler development flow such as GCC or Clang/LLVM is encumbered by a steep learning curve. We consider that this area is ripe for more high-level approaches, and that one such approach can be source-to-source compilation, as a complement to traditional compilation approaches.

Source-to-source compilation is a technology that is widely used and quite mature in certain programming environments, such as JavaScript, but that faces a low adoption rate in others. In the particular case of the C and C++ some of the identified factors include the high complexity of the languages, increased difficulty in building and maintaining C/C++ parsers, or limitations on using source code as an intermediate representation. Additionally, new

technologies such as MLIR have appeared as competitors to source-to-source compilers at this level.

In this paper we present what has been identified as the main challenges regarding source-to-source compilation of C and C++, as well as what we consider to be opportunities and possible directions forward.

We also present Clava, as a use-case of a C/C++ source-to-source compiler that adopts some of these ideas to raise the abstraction level of compiler research on complex compiled languages such as C or C++, and how it has been used in works such as automatic parallelization of for loops, high-level synthesis optimisation, or hardware/software partitioning with run-time decisions, and automatic insertion of custom RISC-V instructions.