title:

High-Level Loop Transformations and Polyhedral Compilation

abstract:

High-level loop transformations change the order in which basic computations in a program are executed and are of great importance to achieving parallelism and optimized reuse of memory. Polyhedral compilation is a framework for analyzing and/or constructing such restructuring transformations that is widely used in high-level synthesis tools and in production compilers such as gcc/graphite, LLVM/Polly and the IBM XL compilers. This framework is based on the polyhedral model, a compact abstraction for (parts of) programs that are "sufficiently regular". The key feature of this model is that it is instance based, allowing for a representation and treatment of individual dynamic executions of a statement inside a loop nest and/or individual array elements.

After a brief introduction to the application of high-level loop transformations such as loop distribution, loop fusion and loop tiling to obtain parallelism or locality, this lecture describes the core concepts behind polyhedral compilation and how they relate to loop transformations. These concepts mainly include dependence analysis, schedules and data-layout transformations. The lectures finishes with an overview of some available software and a more detailed discussion of one of these tools, an automatic parallelizer called PPCG that targets GPUs.

bio:

Sven Verdoolaege is a senior researcher working for Polly Labs. In general, his main research interests are in Polyhedral Compilation. He earned his master degrees in Computer Science (1998) and Artificial Intelligence (1999) as well as his PhD on polyhedral loop transformations and counting integer points in polyhedra in Computer Science (2005) from the Katholieke Universiteit Leuven (Belgium). He continued his research on integer point counting, contributing to primal and weighted counting algorithms. As a postdoc, he successively investigated polyhedral process networks, including non-affine extensions, at Leiden University (Netherlands) and equivalence verification at KU Leuven (Belgium). In 2010 he started conducting research in the group of Albert Cohen at INRIA and ENS (France) on topics such as transitive closures and parallelization for GPUs. As a Polly Labs researcher, he continues to collaborate with this group while also diversifying to working with partners from industry. His research has culminated in various widely used software packages such as isl, an integer set and polyhedral compilation library; barvinok, a library for (weighted) counting of integer sets; pet, a polyhedral parser; and PPCG, a polyhedral parallelizer.