Profile driven loop transformations

Richard Guenther
Your affiliation
your-address@example.com

Abstract

Today scientific computing applications are developed using modern principles of software design. Among others, this leaves specialization and optimization of loop kernels to the compiler. In particular, loops which run for a known low number of iterations in one of the dimensions, such as loops handling boundary condition computation, usually produce inferior code using F95 array expressions or C++ template library utility functions. The same holds true for strides of (multidimensional) arrays which usually are the same for all arrays that participate in a loop kernel, but still are not known so at compile time.

GCC has developed a rich infrastructure for both loop analysis and transformation as well as supporting profile guided optimizations. Using this infrastructure we will present the results of developing loop optimizations that rely on the use of loop versioning and profiling of access function evolution. The goal is to reduce the numbers of induction variables to consider during induction variable optimization and improve the generated code by requiring a less overall number of registers. This is done by providing loop specializations for both the above mentioned cases. We will present the implementation of the instrumentation and the optimizing phase discussing current limitations of the framework GCC provides. A case study involving the `tramp3d` benchmark to gather statistical data for the transformations will be presented as well as performance results for applying the transformation on a selected set of loop kernels.