Compiler Driven Code Comments and Refactoring - DTU Orbit (23/04/2019)

Compiler Driven Code Comments and Refactoring
Helping programmers write parallel software is an urgent problem given the popularity of multi-core architectures. Engineering compilers which automatically parallelize and vectorize code has turned out to be very challenging. Consequently, compilers are very selective with respect to the coding patterns they can optimize. We present an interactive approach and a tool set which leverages advanced compiler analysis and optimizations while retaining programmer control over the source code and its transformation. This allows optimization even when programmers refrain from enabling optimizations to preserve accurate debug information or to avoid bugs in the compiler. It also allows the source code to carry optimizations from one compiler to another. Secondly, our tool-set provides feedback on why optimizations do not apply to a code fragment and suggests workarounds which can be applied automatically. We demonstrate the ability of our tool to transform code, and suggest code refactoring that increase its amenability to optimization. The preliminary results shows that, with our tool-set, automatic loop parallelization with the GNU C compiler, gcc, yields 8.6x best-case speedup over the sequential version. The transformations and suggestions are based on the loop parallelization and matrix reorganization capabilities in the latest production release of gcc.

General information
Publication status: Published
Organisations: Embedded Systems Engineering, Department of Informatics and Mathematical Modeling, IBM Research Laboratory
Contributors: Larsen, P., Ladelsky, R., Karlsson, S., Zaks, A.
Publication date: 2011

Host publication information
Title of host publication: Proceedings of Forth Workshop on Programmability Issues for Heterogeneous Multicores
Source: orbit
Source-ID: 271153
Research output: Chapter in Book/Report/Conference proceeding › Article in proceedings – Annual report year: 2011 › Research › peer-review