Bridging the Performance Gap between Manual and Automatic Compilers with Intent-based Compilation

Abstract

In spite of years of research in automatic parallelization, progress has been slow in terms of tools that can consistently generate scalable, portable and efficient code for multiple architectures. Moreover, a substantial difference in efficiency exists between code generated automatically and code generated by an expert programmer. Although the fact that the best sequential algorithm for a problem can be very different from the best parallel algorithm is well known, the feature of algorithm substitution is absent from most tools available today. However, automatically identifying an algorithm used in code is not trivial considering the nuances in programming style, algorithmic representations and expressions. This paper presents a novel Intent Based Compilation approach that uses a rule-based Expert System Engine to identify the intent of the algorithm used in the code based on fine- and coarse-grained features extracted from code. Using this information, the most optimized algorithm for the target architecture is then substituted. Results obtained by using Amoeba, a framework that incorporates this methodology, on codes obtained from the public domain are presented.

References

- Javier Diaz, Camelia Muñoz-Caro, and Alfonso Niño. A survey of parallel
Bridging the Performance Gap between Manual and Automatic Compilers with Intent-based Compilation


Bridging the Performance Gap between Manual and Automatic Compilers with Intent-based Compilation


Index Terms
Computer Science
Information Sciences

Keywords
Intent-based compilation automation code-to-code transformers parallelization
parallel compilers