Abstract

The front end of a compiler is generally responsible for creating an intermediate representation of the source program whereas the back end of the compiler constructs the desired target program from the intermediate representation and the information in the symbol table. Before the intermediate code is passed to the back end of the compiler, it is necessary to improve the intermediate code so that better target code will result. The code optimization phase in a compiler attempts to improve the target code without changing its output or without side-effects.

Today, most of the compiler research is done in the optimization phase. There are many classical techniques (e.g. Eliminating common sub-expressions, Dead-Code elimination, Constant Folding etc.) that have been used in code optimization. However, the increasing size and complexity of software products and the use of these products in embedded, web-based and mobile systems results in the demand for more optimized versions of the source code. This research paper discusses the challenges involved in code optimization for such systems and some recently developed techniques in code optimization.
New trends and Challenges in Source Code Optimization

References

5. Kenneth Hoste Lieven Eeckhout, ELIS Department, Ghent University,” COLE: Compiler Optimization Level Exploration
7. Stan Yi- Huang Liao, “Code Generation and Optimization for Embedded Digital Signal Processors”, Massachusetts Institute of Technology
12. Qingfeng Zhuge, Bin Xiao, Edwin H.-M. Sha,“ Performance optimization of Multiple Memory Architectures for DSP”
15. Keith D. Cooper 1 and L. Taylor Simpson, “Live Range Splitting in a Graph Coloring Register Allocator”, Rice University, Houston, Texas, USA,
16. Preston Briggs, Keith D. Coope, Linda Torczon,, “Aggressive Live Range Splitting”, Houston University, Texas, USA
17. Michael Burke, Linda Torczon, “Inter-procedural Optimization: Eliminating Unnecessary Recompilation”, IBM Research, Rice University

Index Terms

Computer Science  Information Sciences

Keywords
Optimization, Reverse Inlining, Cross Linking, Address-Code, Leaf Function