

Compiler Construction

Term Paper

19BCE248 - Dhruvil Shah | 19BCE286 -Jurin Vachhani

Title: Compiler Optimization Techniques

Abstract:

Without altering the program's meaning, a compiler converts code written in one language to another. Additionally, it is assumed that a compiler will make the target code effective and space and time optimal. Basic error detection and translation methods are part of the compilation process. The accuracy of these assessments is heavily constrained by complicated target architectures and unforeseen optimization interactions, which results in performance loss as a result of suboptimal optimization choices. There are hundreds of optimizations available in contemporary compilers, and this number will only increase in the future. The fact that each optimization would attempt to target particular code constructs and boost their efficiency by implementing particular templates is the main reason for the increase in the number of optimizations that the compiler could perform. A growing number of optimizations that are being applied to a certain code may not have been intended for the code that is now being compiled as the optimizations target increasingly specific code constructs. The iterative compilation strategy, which encourages investigating several optimization alternatives and choosing the optimal one a posteriori, can help prevent this performance loss. The newest optimization techniques can be used by compiler suites to produce code 20–30% faster than typical compilers, which has a variety of advantages for developers. Without having to rely on faster and more expensive processors, improved optimization would also enable developers to reduce costs and improve products. In this paper, we discuss various methods for improving compiler optimization, such as caching techniques, dynamic optimization methods, and compiler optimization for dynamic languages like JavaScript. Additionally covered are machine learning approaches that can be used to improve compiler optimization.

Motivation:

As we know that not every developer can write more optimized code as the current trend in the IT industry is based on new innovation rather than focussing on how to optimize the things that are already done. So the responsibility of the compiler becomes more exhaustive as it needs to provide the best possible way of compiling the code with limited resources in limited time. Thus considering these as a vast field of study, we are keen to deep dive into such things to make some learning out of it.

Some Major techniques to be included:

- Cache optimization
- Dynamic optimization
- Optimization for dynamic languages
- Machine learning optimization techniques