World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:16, No:12, 2022

A User-Directed Approach to Optimization via Metaprogramming

Authors : Eashan Hatti

Abstract : In software development, programmers often must make a choice between high-level programming and high-performance programs. High-level programming encourages the use of complex, pervasive abstractions. However, the use of these abstractions degrades performance-high performance demands that programs be low-level. In a compiler, the optimizer attempts to let the user have both. The optimizer takes high-level, abstract code as an input and produces low-level, performant code as an output. However, there is a problem with having the optimizer be a built-in part of the compiler. Domain-specific abstractions implemented as libraries are common in high-level languages. As a language's library ecosystem grows, so does the number of abstractions that programmers will use. If these abstractions are to be performant, the optimizer must be extended with new optimizations to target them, or these abstractions must rely on existing general-purpose optimizations. The latter is often not as effective as needed. The former presents too significant of an effort for the compiler developers, as they are the only ones who can extend the language with new optimizations. Thus, the language becomes more high-level, yet the optimizer – and, in turn, program performance – falls behind. Programmers are again confronted with a choice between high-level programming and high-performance programs. To investigate a potential solution to this problem, we developed Peridot, a prototype programming language. Peridot's main contribution is that it enables library developers to easily extend the language with new optimizations themselves. This allows the optimization workload to be taken off the compiler developers' hands and given to a much larger set of people who can specialize in each problem domain. Because of this, optimizations can be much more effective while also being much more numerous. To enable this, Peridot supports metaprogramming designed for implementing program transformations. The language is split into two fragments or "levels", one for metaprogramming, the other for high-level general-purpose programming. The metaprogramming level supports logic programming. Peridot's key idea is that optimizations are simply implemented as metaprograms. The meta level supports several specific features which make it particularly suited to implementing optimizers. For instance, metaprograms can automatically deduce equalities between the programs they are optimizing via unification, deal with variable binding declaratively via higher-order abstract syntax, and avoid the phase-ordering problem via non-determinism. We have found that this design centered around logic programming makes optimizers concise and easy to write compared to their equivalents in functional or imperative languages. Overall, implementing Peridot has shown that its design is a viable solution to the problem of writing code which is both high-level and performant.

Keywords : optimization, metaprogramming, logic programming, abstraction
Conference Title : ICSLEM 2022 : International Conference on Software Language Engineering and Metaprogramming
Conference Location : Auckland, New Zealand
Conference Dates : December 02-03, 2022