

1. **Record Nr.** TD18045527
Autore D'ELIA, DANIELE CONO
Titolo New techniques for adaptive program optimization [Tesi di dottorato]

Lingua di pubblicazione Inglese

Formato Tesi di dottorato

Livello bibliografico Monografia

Note diritti: info:eu-repo/semantics/openAccess
In relazione con info:eu-repo/semantics/altIdentifier/hdl/11573/876041

Sommario Adaptive optimization technology is a key ingredient in modern runtime systems. This technology aims at improving performance by making optimization decisions on the basis of a program's observed behavior. Application virtual machines indeed face different and perhaps more compelling issues compared to traditional static optimizers, as dynamic language features can force the deferral of most effective optimizations until run time. In this thesis, we present novel ideas to improve adaptive optimization, focusing on two main problems: collecting fine-grained program profiles with low overhead to guide feedback-directed optimization, and supporting continuous optimization and deoptimization by diverting execution across dynamically generated code versions. We present two profiling techniques: the first works at inter-procedural level to collect calling context information for hot code portions, while the second captures cyclic-path profiles within a function's boundaries. Both techniques rely on efficient and elegant data structures, advancing the state of the art of the theory and practice of the performance profiling literature. We then focus our attention on supporting continuous optimization through on-stack replacement (OSR) mechanisms. We devise a new OSR framework encoded entirely at intermediate-representation level, which extends the best OSR practices with the

ability to perform OSR at nearly any program location. Our techniques pave the road to aggressive optimizations and debugging techniques that were not supported by previous approaches. The main technical challenge is how to automatically generate compensation code to fix the program's state across an OSR transition between different code versions. We present a conceptual framework for OSR, distilling its essence to a core calculus with an operational semantics. Using bisimulation techniques, we describe how OSR can be correctly supported in the presence of common compiler optimizations, providing the first soundness results in this context. We implement our ideas in production systems such as Jikes RVM and the LLVM compiler toolchain, and evaluate their performance against a variety of prominent benchmarks. We investigate the end-to-end utility of our techniques in a series of case studies: we illustrate two possible applications of multi-iteration path profiling, and show how our OSR techniques advance the state of the art for MATLAB code optimization and for source-level debugging of optimized code. Part of the results of this thesis have been published in PLDI, OOPSLA, CGO, and Software Practice and Experience.

Localizzazioni e accesso

http://memoria.depositolegale.it/*/http://hdl.handle.net/11573/876041
