Fabio Mascarenhas de Queiroz

Optimized Compilation of a Dynamic Language to a Managed Runtime Environment

TESE DE DOUTORADO

DEPARTMENTO DE INFORMÁTICA
Postgraduate Program in Informatics

Rio de Janeiro
September 2009

Pontifícia Universidade Católica
do Rio de Janeiro

Fabio Mascarenhas de Queiroz

Optimized Compilation of a Dynamic Language to a Managed Runtime Environment

TESE DE DOUTORADO

Dissertation presented to the Postgraduate Program in Informatics of the Departamento de Informática, PUC–Rio as partial fulfillment of the requirements for the degree of Doutor em Informática

Advisor: Prof. Roberto Ierusalimschy

Rio de Janeiro
September 2009

Fabio Mascarenhas de Queiroz

Optimized Compilation of a Dynamic Language to a Managed Runtime Environment

Dissertation presented to the Postgraduate Program in Informatics of the Departamento de Informática, PUC–Rio as partial fulfillment of the requirements for the degree of Doutor em Informática. Approved by the following commission:

Prof. Roberto Ierusalimschy
Advisor
Departmento de Informática — PUC–Rio

Prof. Noemi de La Rocque Rodriguez
Departamento de Informática — PUC–Rio

Prof. Edward Hermann Haeusler
Departamento de Informática — PUC–Rio

Prof. Sandro Rigo
Instituto de Computação – UNICAMP

Prof. Claudio Luis de Amorim
COPPE — UFRJ

Prof. José Eugenio Leal
Head of the Science and Engineering Center — PUC–Rio

Rio de Janeiro — September 4, 2009

Fabio Mascarenhas de Queiroz

Fabio Mascarenhas de Queiroz graduated from the Universidade Federal da Bahia (Salvador, Bahia) in Computer Science. He then obtained a Master degree at PUC–Rio in programming languages, and has now finished his Ph. D. at PUC–Rio, also in programming languages.

Bibliographic data

Queiroz, Fabio Mascarenhas de

Optimized Compilation of a Dynamic Language to a Managed Runtime Environment / Fabio Mascarenhas de Queiroz; advisor: Roberto Ierusalimschy. — Rio de Janeiro : PUC–Rio, Departamento de Informática, 2009.

v., 97 f: il. ; 29,7 cm

1. Tese de Doutorado - Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Informática.

Inclui bibliografia.

1. Informática – Dissertação. 2. Linguagens de Programação. 3. Compiladores. 4. Inferência de Tipos. 5. Ambientes de Execução Gerenciada. 6. Desempenho. 7. Linguagens Dinâmicas. 8. Lua. I. Ierusalimschy, Roberto. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Informática. III. Título.

CDD: 004

Acknowledgments

I would like to thank my advisor, Professor Roberto Ierusalimschy, for his advice and insights in crucial moments of this work, and for the many intellectually stimulating, and entertaining, conversations that we had through these almost seven years that I have been his student, both Master's and Ph. D.'s.

I also would like to thank my friends and colleagues at Lablua, Sérgio Medeiros and Hisham Muhammad, for providing a productive and fun work environment.

I also thank my friends, my family, and specially Cristina, for the constant encouragement, prodding, and for enduring me in when I was my most stressed and least social.

Finally, I would like ot thank CNPq and FAPERJ for their financial support in respectively the first and second half of my doctorate, without which this work could not have been done.

Abstract

Queiroz, Fabio Mascarenhas de; Ierusalimschy, Roberto. **Optimized Compilation of a Dynamic Language to a Managed Runtime Environment**. Rio de Janeiro, 2009. 97p. Tese de Doutorado — Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

Managed runtime environments have become popular targets for compilers of high-level programming languages. They provide a high-level type system with enforced runtime safety, as well as facilities such as garbage collection, possibly sandboxed access to services of the underlying platform, multithreading, and a rich library of data structures and algorithms. But managed runtime environments lack a clear performance model, which hinders attempts at optimizing the compilation of any language that does not have a direct mapping to the runtime environments' semantics. This is aggravated if the language is dynamically typed.

We assert that it is possible to build a compiler for a dynamic language that targets a managed runtime environment so that it rivals a compiler that targets machine code directly in efficiency of the code it generates. This dissertation presents such a compiler, describing the optimizations that were needed to build it, and benchmarks that validate these optimizations. Our optimizations do not depend on runtime code generation, only on information that is statically available from the source program. We use a novel type inference analysis to increase the amount of information available.

Keywords

Programming Languages. Compilers. Type Inference. Managed Runtime Environments. Benchmarking. Dynamic Languages. Common Language Runtime. Lua.

Resumo

Queiroz, Fabio Mascarenhas de; Ierusalimschy, Roberto. **Compilação Otimizada de uma Linguagem Dinâmica para um Ambiente de Execução Gerenciada**. Rio de Janeiro, 2009. 97p. Tese de Doutorado — Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

Ambientes de Execução Gerenciada tornaram-se alvos populares para compiladores de linguagens de programação de alto nível. Eles provêem um sistema tipos de alto nível com segurança de memória garantida, assim como facilidades como coleta de lixo, acesso a serviços da plataforma subjacente (possivelmente através de uma sandbox), multithreading, e uma rica biblioteca de estruturas de dados e algorithmos, mas não possuem um modelo de desempenho claro, o que atrapalha tentativas de otimização de qualquer linguagem que não tenha um mapeamento direto na semântica do ambiente de execução, especialmente se a linguagem é dinamicamente tipada.

Nós afirmamos que é possível construir um compilador para uma linguagem dinâmica que tem como alvo um ambiente de execução gerenciada que rivaliza um compilador que tem como alvo linguagem de máquina na eficiência do código que ele gera. Essa tese apresenta um compilador com tal característica, descrevendo as otimizações necessárias para sua construção, e testes de desempenho que validam essas otimizações. Nossas otimizações não dependem de geração de código em tempo de execução, apenas em informação estaticamente disponível no código fonte. Nós usamos uma nova análise de inferência de tipos para aumentar a quantidade de informação disponível.

Palavras–chave

Linguagens de Programação. Compiladores. Inferência de Tipos. Ambientes de Execução Gerenciada. Desempenho. Linguagens Dinâmicas. Lua.

Contents

List of Figures

List of Tables