JVM COMPILER BACKEND FOR BALLERINA INTERMEDIATE REPRESENTATION

Thangarajah Kishanthan
179329D

Degree of Master of Science

Department of Computer Science and Engineering
University of Moratuwa
Sri Lanka

May 2019

JVM COMPILER BACKEND FOR BALLERINA INTERMEDIATE REPRESENTATION

Thangarajah Kishanthan
179329D

Thesis submitted in partial fulfillment of the requirements for the degree Master of Science

Department of Computer Science and Engineering
University of Moratuwa
Sri Lanka

May 2019

DECLARATION

I declare that this is my own work and this thesis does not incorporate without acknowledgement any material previously submitted for a Degree or Diploma in any other University or institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

Also, I hereby grant to University of Moratuwa the non-exclusive right to reproduce and distribute my thesis/dissertation, in whole or in part in print, electronic or other medium. I retain the right to use this content in whole or part in future works (such as articles or books).

Signature: Date:

(T. Kishanthan)

The above candidate has carried out research for the Masters thesis under my supervision.

Signature: Date:

(Dr. Indika Perera)

ABSTRACT

Ballerina is an open source, strongly typed language for writing microservices and network applications with main focus on solving enterprise integration requirements. The ballerina compiler converts the ballerina source to set of ballerina byte code which is then executed by the ballerina virtual machine (BVM). The BVM does not perform well for most of the CPU bound operations due to its current design. This project focus on compiling ballerina source to JVM byte code and will be executed by the JVM directly, which will solve the performance bottleneck at BVM. This project also proposes a new compiler architecture, in which, the ballerina source code is transformed to an intermediate representation which is a low level representation of the ballerina program and it is used for generating the target JVM byte code. The performance of JVM based compiler backend implementation against the current BVM was compared for certain algorithms and programs. From the evaluation of the test results, it is found that the JVM target outperforms the ballerina runtime by factor of 100 in certain scenarios. With this promising results, the proposed new compiler architecture based on ballerina intermediate representation and the JVM compiler backend can potentially be used as the replacement for current ballerina compiler and runtime.

ACKNOWLEDGEMENT

I would like to express my profound gratitude to my supervisor Dr. Indika Perera, for his invaluable support throughout by advising and guiding me through the correct directions. His expertise and continuous guidance are one of the key reasons for the success of this research.

I should also thank Dr. Sanjiva Weerawarana (Founder and Chairman of WSO2) and Mr. Sameera Jayasoma (Senior Director at WSO2), who encouraged me to complete this research project. I am mostly thankful for my wife, my parents for their immense support and patience at all the time. At last but not least, I'm thankful to all my colleagues at WSO2, who helped me a lot in various ways throughout this research.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

Abbreviation	Description
BVM	Ballerina Virtual Machine
AST	Abstract Syntax Tree
IR	Intermediate Representation
BIR	Ballerina Intermediate Representation
JVM	Java Virtual Machine
LLVM	Low Level Virtual Machine
CFG	Control Flow Graph