Bachelor Thesis

July 14, 2019

Effects of source code properties on variability of software microbenchmarks written in Go

Mikael Basmaci

of Istanbul, Turkey (15-721-244)

supervised by

Prof. Dr. Harald C. Gall

Christoph Laaber

University of Zurich[UZH]

s.e.a.l.

software evolution & architecture lab

Bachelor Thesis

Effects of source code properties on variability of software microbenchmarks written in Go

Mikael Basmaci

University of Zurich[UZH]

s.e.a.l.
software evolution & architecture lab

Bachelor Thesis

Author: Mikael Basmaci, mikael.basmaci@uzh.ch

URL: <url if available>

Project period: 25.03.2019 - 25.09.2019

Software Evolution & Architecture Lab
Department of Informatics, University of Zurich

Acknowledgements

Here comes the acknowledgements.

Abstract

Here comes the abstract.

Zusammenfassung

Here comes the summary.

Contents

List of Figures

List of Tables

List of Listings

Chapter 1

Introduction

The importance of testing software systems for performance has grown in recent years. Software developers can analyze the performance of parts of their software with software microbenchmarks, which can be defined as unit tests for performance of a software. Executing microbenchmarks for a defined time period, one can sample an average execution time of the benchmark, as well as the data points as the results of multiple iterations. The variability of these results may depend on a lot of factors such as the execution platform, the hardware the benchmarks are executed on, or even the programming language of the software itself. It is important to predict the variability of these results to select the stable benchmarks to be run, because executing performance tests of a software is usually a long, time consuming process. Moreover, by predicting the variability, the stability of benchmarks can be estimated. By understanding the root causes for benchmark variability and being able topredict these, we could support developers to write better benchmarks. One of the ways to predict the variability might be through analyzing source code properties of the software. This can on one hand help the developers identify the cause of slowdowns in a newer version for example, on the other hand help them understand which source code property affects the results in which way.

Chapter 2

Related Work

Here comes the literature research about benchmark variability. While some papers look at the benchmark variability causes, some of them try to predict the performance regressions based on different versions of a software by analyzing different commits and mining source code.

Chapter 3

Methodology

Here comes the methodology we use during this project. This includes the categorization of benchmarks of 228 projects written in Go by looking at the total executions, the mean, coefficient of variation and relative confidence interval width with 95% and 99% confidence intervals. Secondly, we introduce the methodology we used to extract source code properties from in the first place chosen benchmarks. Thirdly, comparing the source code properties with the variabilities of benchmarks by using regression models to achieve the results.

Chapter 4

Results

In this section, we present the results from this project. Hopefully there will be significant, interesting and appliable results that contribute to science.

Chapter 5

Discussion

In this section, we discuss the results that we obtained and how relevant these are.

Chapter 6

Threats to the validity

If needed, there can be this section about the threats to the validity of this bachelor thesis report. Maybe we did something wrong whilst analyzing the data, maybe the variabilities are not realistic?

Chapter 7

Conclusion

We finally conclude with what we have done with this project: How we started, which steps we took and which results we achieved.

Bibliography