Empirical Evaluation of Test Suite Reduction

Adrian Regenfuß

December 15, 2020

Contents

Abstract

As a response to ever-growing test suites with long runtimes, several different approaches have been developed to reduce the time for test suite execution to give useful results. One of these approaches is test suite reduction: selecting a sample of tests that maximizes coverage on the tested source code. This paper attempts to replicate the findings in Cruciani et al. 2019, which borrows techniques from big data to handle very large test suites. We use independently generated testing data from open source projects, and find that TODO.

1 Introduction

To prevent the introduction (or re-introduction) of bugs, software developers often test their software after making changes to it. This is known as regression testing, and takes up a significant portion of development cost. However, the resulting test suites can grow quite significantly in size and execution time, which hinders development speed and increases costs.

This work attempts to determine how different test suite reduction strategies compare to each other in terms of time performance, fault detection loss and magnitude of reduction.

2 Related Work

2.1 Handling Large Test Suites

2.1.1 Test Case Selection

2.1.2 Test Case Priorization

2.1.3 Test Suite Reduction

2.2 Algorithms for Reduction

2.2.1 Greedy Selection

2.2.2 Clustering

2.2.3 Searching

2.2.4 Hybrids

3 Approach

3.1 Replicating "Scalable Approaches for Test Suite Reduction"

3.2 Algorithms Used

3.2.1 FAST

FAST++

FAST-all

References

Emilio Cruciani, Breno Miranda, Roberto Verdecchia, and Antonia Bertolino. Scalable approaches for test suite reduction. In *2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE)*, pages 419–429. IEEE, 2019.