Abstract: Assessing Transition-based Test Selection Algorithms at Google

Claire Leong, Abhayendra Singh
Google Inc.
Mountain View, CA, USA
{claireleong, abhayendra}@google.com

Mike Papadakis, Yves Le Traon
University of Luxembourg
Luxembourg
{michail.papadakis, yves.letraon}@uni.lu

John Micco
Los Gatos, CA, USA
john.micco@gmail.com

I. Reference

This work was presented in the 41st International Conference on Software Engineering: Software Engineering in Practice, *ICSE (SEIP)* 2019, Montreal, QC, Canada, May 25-31, 2019.

II. Introduction

Applying continuous integration in large and rapidly evolving codebases requires substantial (computational) resources to test every committed change. Recent experience reports[1], show that in Google, a code commit happens every second (on average). This commit pace results in more than 150 million test executions per day. Evidently such computational demands involve a high operating costs and require long time (up to nine hours) to integrate the commits[1].

In view of this, the Google continuous integration environment is expected to balance between a small consumption of resources and fast response, i.e., informing developers whether any transition was triggered by their changes. In this context, a transition is "a change in state in the sequence of results across commits for a test, either from Pass to Fail or Fail to Pass"[2]. Detecting transitions (instead of test failures) has been ignored by previous research with the unfortunate result of ignoring Fail to Pass transitions that are important.

Moreover, an additional practical problem is the flaky test problem, i.e., "tests with non-deterministic outcomes"[2]. Interestingly, at Google 84% of test transitions are due to flakiness, and 16% of the tests involve some level of test flakiness[2]. Evidently, such a prevalence of unstable results has a major impact on regression testing and developed heuristics.

Previous research aimed at selecting tests that failed in the recent execution history. Unfortunately, because of the frequency that flaky tests transition make such techniques prioritize the flaky instead of non-flaky tests. This calls into question whether these methods offer practical benefits. We investigate this by developing a mechanism for comparing test selection methods with regard to transition detection.

We use the historical sequence of test results to simulate the performance of test selection methods on real data by accounting for test flakiness (by filtering flaky test executions). Our primary aim was to record the relative trade-offs between skipping tests and triggering transitions. Since in Google, tests run in parallel, we focus on test selection and aim at identifying signals with strong results which could lead to future research into test selection schemes with relatively accurate trade-offs.

We thus, evaluate 5 algorithms using simple heuristics; 2 based on the recent commits that have an impact on a test, i.e., number of affecting commits and number of authors committing affecting code, 2 based on previous test results, i.e., pre-submit out- comes - testing prior to commit - and number of historical transitions, and 1 based on test characteristics, i.e., number of directories shared with modified files.

We show that flaky tests have a significant impact and that algorithms based on the recently invoked commits perform best (outperforming a random baseline). Algorithms based on prior testing results perform worse but still significantly better than the random baseline. Unfortunately, heuristics based on test characteristics perform similarly to random. Interestingly, the distance from the optimal performance range from 0% to 90% indicating room for improvements.

III. Contribution

- The significance and impact of flaky test transitions on regression test selection heuristics (and evaluations) in the continuous integration context.
- The important of evaluating test selection heuristics with non-flaky test transitions.
- Some preliminary result on evaluating regression test selection heuristics at Google continuous integration environment.

IV. Talk Outline

The talk outline will be the following:

- Brief introduction on Google continuous integration environment.
- Introduce the concepts of flaky test transitions.
- Introduce transition-based test selection heuristics.
- Present an empirical evaluation of the heuristics at Google.

[1] A. M. Memon, Z. Gao, B. N. Nguyen, S. Dhanda, E. Nickell, R. Siemborski, and J. Micco, "Taming google-scale continuous testing", in ICSE-SEIP, 2017, pp. 233242

[2] Claire Leong, Abhayendra Singh, Mike Papadakis, Yves Le Traon, John Micco, "Assessing transition-based test selection algorithms at Google" in ICSE-SEIP 2019, pp. 101-110.