MSeer – an Advanced Technique for Locating Multiple Bugs in Parallel

Ruizhi Gao
Department of Computer Science
University of Texas at Dallas
Richardson, USA
gxr116020@utdallas.edu

W. Eric Wong*
Department of Computer Science
University of Texas at Dallas
Richardson, USA
ewong@utdallas.edu

ABSTRACT

In practice, a program may contain multiple bugs. The simultaneous presence of these bugs may deteriorate the effectiveness of existing fault-localization techniques to locate program bugs. While it is acceptable to use all failed and successful tests to identify suspicious code for programs with exactly one bug, it is not appropriate to use the same approach for programs with multiple bugs because the *due-to* relationship between failed tests and underlying bugs cannot be easily identified. One solution is to generate fault-focused clusters by grouping failed tests caused by the same bug into the same clusters. We propose MSeer – an advanced fault localization technique for locating multiple bugs in parallel. Our major contributions include the use of (1) a revised Kendall tau distance to measure the distance between two failed tests, (2) an innovative approach to simultaneously estimate the number of clusters and assign initial medoids to these clusters, and (3) an improved K-medoids clustering algorithm to better identify the due-to relationship between failed tests and their corresponding bugs. Case studies on 840 multiple-bug versions of seven programs suggest that MSeer performs better in terms of effectiveness and efficiency than two other techniques for locating multiple bugs in parallel.

KEYWORDS

Software fault localization, parallel debugging, multiple bugs, clustering, distance metrics

1 INTRODUCTION

Regardless of the effort spent developing a computer program, it may still contain bugs [2]. When the execution of a test case on a program fails, it suggests that the program has bugs. However, the burden of locating and fixing these bugs is on the programmers. To do so, they must first be able to identify exactly where these bugs are. Known as fault localization, this step can be very time consuming and expensive. Many fault localization techniques use all failed and successful test cases to prioritize code and generate a ranking of statements in descending order of their suspiciousness values. Fault localization techniques using this approach have been well reported in literature [3]. Many of these studies assume there is exactly one bug in the program. Since a single-bug assumption may not hold in practice and mixed failed test cases associated with different causative bugs can reduce the effectiveness of a fault localization technique, in this paper we extend existing techniques and propose an advanced fault localization technique, MSeer [1], to debug programs with multiple bugs in parallel. There are four novel aspects of MSeer:

- In MSeer, a failed test case is represented by a suspiciousness ranking of statements generated by a given fault localization technique using the corresponding failed test and all successful tests. The ranking is in descending order of each statement's likelihood of containing bugs.
- MSeer uses a revised Kendall tau distance that gives greater weight to more suspicious statements and smaller weight to less suspicious statements to measure the distance between two failed test cases (i.e., two suspiciousness rankings).
- MSeer applies an innovative approach to estimate the number of clusters and, at the same time, assign initial medoids to these clusters.
- MSeer performs clustering using an improved K-medoids algorithm that uses an innovative approach to help determine appropriate initial medoids without using all possible combinations.

2 CASE STUDIES

Case studies on seven medium- to large-sized programs (*gzip*, *grep*, *make*, *flex*, *ant*, *socat* and *xmail*) written in different languages (C, C++ and Java) with various functionalities were conducted to evaluate the effectiveness and efficiency of MSeer based on several metrics. In total, 840 multiple-bug faulty versions (2-bug, 3-bug, 4-bug, and 5-bug) of the seven programs were used. A cross-comparison between MSeer and two other techniques is conducted. Our results strongly suggest that MSeer performs better in terms of both effectiveness and efficiency.

REFERENCES

[1] R. Gao and W. E. Wong, "MSeer - An Advanced Technique for Locating Multiple Bugs in Parallel," Transactions on Software Engineer, accepted on November 2017, DOI: 10.1109/TSE.2017.2776912
[2] W. E. Wong, X. Li, P. A. Laplante, "Be More Familiar with Our Enemies and Pave the Way Forward: A Review of the Roles Bugs Played in Software Failures," Journal of Systems and Software, vol. 133, pp. 68-94, November 2017
[3] W. E. Wong, R. Gao, Y. Li, R. Abreu and F. Wotawa, "A Survey on Software Fault Localization," *IEEE Transactions on Software Engineering*, vol. 42, issue. 8, pp. 707-740, August 2016

*W. Eric Wong is the corresponding author of this paper

ICSE '18, May 27-June 3, 2018, Gothenburg, Sweden

ACM ISBN 978-1-4503-5638-1/18/05.
https://doi.org/10.1145/3180155.3182552