Poster: A Novel Variable-centric Fault Localization Technique

Jeongho Kim
Sungkyunkwan University
Suwon, Korea
jeonghodot@skku.edu

Jindae Kim
HKUST
Sai Kung, Hong Kong
jdkim@cse.ust.hk

Eunseok Lee
Sungkyunkwan University
Suwon, Korea
leees@skku.edu

ABSTRACT

Fault localization is one of the most important debugging tasks. Therefore, many techniques have already been developed to improve the efficiency. Among them, the spectrum-based fault localization technique is the most popular, and it has been the subject of 35% of total fault localization-related studies. SFL techniques leverage coverage spectra and localize a fault based on the coverage difference between passed and failed test cases. However, it is difficult to localize faults effectively when coverage differences are not clear. Therefore, we propose a novel variable-centric fault localization technique to improve performance of existing techniques. Proposed technique extracts suspicious variables and uses them to generate a suspicious ranked list. In an evaluation with 120 C faults, the proposed technique outperforms SFL techniques with the same similarity coefficient. The average Exam score of proposed techniques are reduced by 55% compared to SFL techniques.

CCS CONCEPTS

• **Software and its engineering → Software testing and debugging**

KEYWORDS

Software debugging, Software testing, Spectrum-based fault localization, Suspicious variable, Variable-centric

1 INTRODUCTION

Fault localization task is one of the most tedious and time-consuming for developers. To reduce this effort, researchers have proposed various fault localization techniques. Among these techniques, Spectrum-based Fault Localization (SFL) technique is one of the most popular. Many SFL techniques with various similarity coefficients such as Tarantula [3], Jaccard [1], Naish [5], and GP [7] have been proposed to improve localization performance. However, such SFL techniques cannot localize faults effectively if coverage information is not sufficient to distinguish execution of passed and failed test cases [4]. Therefore, our idea to overcome this limitation and improve the fault localization performance of SFL techniques is to leverage additional information about program entities. More precisely, we focus on variables that appear at source code lines and try to find suspicious variables. The contributions of this paper are listed below.

- A novel variable-centric fault localization technique with improved performance.
- A new approach to fault localization based on suspicious variables that can be identified by information originally available to SFL techniques.

2 PROPOSED TECHNIQUE

We propose a novel variable-centric fault localization technique. The proposed technique only requires information about variables in the source code and coverage information. Coverage information can be easily obtained from inputs originally used in SFL techniques. Variable information can also be easily extracted from source code using automation tools. Therefore, the proposed technique is not only very lightweight like SFL technique, but it can also easily be combined with other techniques.

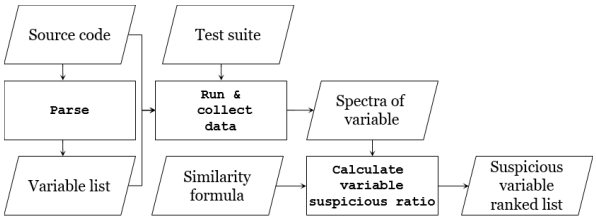

Figure 1: An Overview of Variable-centric Fault Localization Technique.

Figure 1 illustrates the overall process of the proposed technique. First, we parse source code to obtain a list of variables. Then, we convert the coverage of statements to the coverage of the variables. With this converted coverage information, we compute suspicious ratio of each variable, and produce a ranked list of the suspicious variables. Finally, we leverage the list of the suspicious variables

This research was supported by the Next-Generation Information Computing Development Program through the National Research Foundation of Korea(NRF)(No. 2017M3C4A7066206) funded by the Ministry of Education and Science and Technology and Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(No. 2016R1D1A1B03934610).

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

ACM ISBN 978-1-4503-5663-3/18/05.
https://doi.org/10.1145/3183440.3194956

to generate a ranked list of suspicious statements, which is the same output of common SFL techniques. To evaluate the proposed technique, we applied the technique to 120 faults from seven C programs in the Siemens suite using Exam score.

3 EVALUATION

3.1 Benchmark

To evaluate the proposed technique, we employ the Siemens suite for C programs. This project is a popular set of faults that have been used for many fault localization studies [6].

3.2 Field Dependent BLS Measurements and DMM Calculations

One popular method of evaluating the performance of fault localization techniques is computing the Exam score. Exam score indicates the amount of code that needs to be examined until a fault is localized. The formula is as follows [2].

$$EXAM\ score = \frac{Rank\ of\ an\ actual\ fault\ location}{Total\ number\ of\ code\ lines} \times 100$$

4 RESULTS

To verify the validity of this study, we set the following four research questions. However, because of space limitations, we answer only one research question based on one experiment.

4.1 RQ1: How well does proposed technique localize faults?

4.2 RQ2: How well does proposed technique place faulty statements at low ranks?

4.3 RQ3: Is proposed technique effective for every fault?

4.4 RQ4: Can we increase proposed technique performance and applicability by using additional methods?

The main purpose of using an automatic fault localization technique is to help the developers during debugging by reducing the search area for finding a fault. Hence, we evaluated proposed technique by assessing the amount of source code to be examined until a real faulty statement is found.

To answer RQ1, we computed Exam score of SFL and proposed techniques for all 120 faults from the Siemens suite. Figure 2 shows Exam score of SFL and proposed techniques using nine different similarity coefficients. Each group represents Exam score of SFL (left) and proposed techniques (right) computed with a similarity coefficient. The Exam score shown here are average scores of 120 programs in the Siemens suite for each similarity coefficient.

For example, in Tarantula, the SFL Exam score is 14.84, the proposed technique Exam score is 6.53, and the reduced effort is about 56%. This result indicates that the SFL technique should examine 14.84% of the total source code lines to find the fault, while the proposed technique can find faults by reviewing only 6.53% of the total lines. Therefore, our proposed technique can reduce about 56% of the effort for examination compared to SFL techniques. The minimum reduced effort is 57% when we use GP26 as the similarity coefficient. If we use the proposed technique, we do not need to examine at least a half of source code lines which we originally need to check with SFL techniques.

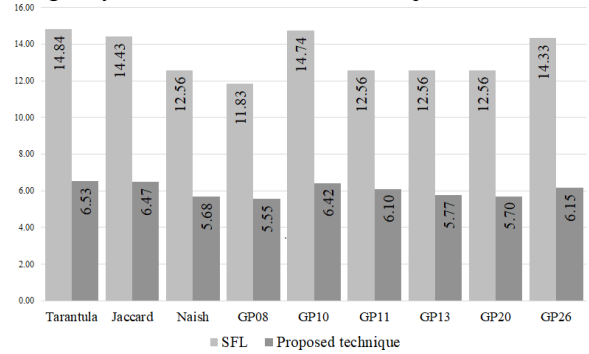

Figure 2: Comparison of Exam score between SFL and Proposed technique.

5 CONCLUSION

We proposed a novel variable-centric fault localization technique to more accurately localize faults. The proposed technique exploits additional information obtained from source code to address problems in existing spectrum-based techniques that rely on coverage information only. In an evaluation, our proposed technique outperforms SFL techniques in locating faults at the top of ranked lists for the majority of faults collected from C programs. We believe that the future of this proposed technique is promising since it is a lightweight technique that can collaborate with other techniques including SFL. A few more computations with variable information obtained by parsing buggy source code would be beneficial. The characteristics of our proposed technique allow the possibility of collaboration with other future techniques. Therefore, the usefulness of this technique is not just limited to the current results.

REFERENCES

[1] Rui Abreu, Peter Zoeteweij, and Arjan J.C. van Gemund. 2006. An Evaluation of Similarity Coefficients for Software Fault Localization. In Proceedings of the Pacific Rim International Symposium on Dependable Computing. IEEE, 39–46.

[2] Ruizhi Gao, W. Eric Wong, Zhenyu Chen, and Yabin Wang. 2017. Effective software fault localization using predicted execution results. Software Quality Journal 25, 1 (March 2017), 131–169.

[3] James A. Jones and Mary Jean Harrold. 2005. Empirical evaluation of the tarantula automatic fault-localization technique. In Automated Software Engineering. ACM, 273–282.

[4] Jeongho Kim, Hohyeon Jeong, and Eunseok Lee. 2017. Failure History Data-based Test case Prioritization for Effective Regression Test. In Proceedings of the Symposium on Applied Computing. ACM, 1409–1415.

[5] Naish Lee, Hua Jie Lee, and Kotagiri Ramamohanarao. 2006. A model for spectra- based software diagnosis. ACM Transactions on Software Engineering and Method- ology, Article 11 (August 2006).

[6] University Of Nebraska-Lincoln. 2005. Software-artifact Infrastructure Reposi- tory. (2005). http://sir.unl.edu/php/index.php

[7] Shin Yoo. 2012. Evolving Human Competitive Spectra-Based Fault Localisation Techniques. Search Based Software Engineering LNCS 7515 (2012), 244–258.