Analysis of Fault Localization Techniques on REST APIs

Aakash Kulkarni
School of EECS
Oregon State University
Corvallis, United States
kulkaraa@oregonstate.edu

Soon Song Cheok
School of EECS
Oregon State University
Corvallis, United States
cheoks@oregonstate.edu

Shreyes Joshi
School of EECS
Oregon State University
Corvallis, United States
joshish@oregonstate.edu

***Abstract*—**
***Index Terms*—**

I. Introduction

A. Problem statement

REST APIs (Representational State Transfer) have become backbone of the modern web and cloud applications. They facilitate seamless interactions between client and server through stateless communication, enabling services to be scalable, reliable, and easily integrateable. Basically, REST APIs are a set of rules and standards used to enable communication between different software applications over the internet. They are built around the use of standard HTTP methods such as GET, POST, PUT, and DELETE to interact with resources, which are any kind of data or service that can be named on a network. Given their critical role, the effective identification and resolution of faults within REST APIs remain a significant challenge, prompting the need for research on how fault localization techniques work to their unique structure and functionality.

B. Motivation

As systems that rely on REST APIs grow in scale and complexity, minor faults can escalate into major disruptions, impacting user experience and business operations. This research is motivated by the need to evaluate how well current fault localization techniques perform in the unique context of REST APIs. The goal is to determine if these techniques can indeed be applied effectively to REST APIs and, if so, to explore which types of faults are more amenable to being localized. This understanding could potentially allow developers to more efficiently diagnose and address issues, thereby enhancing the stability and performance of REST API-based systems.

C. Relation with software engineering research

Fault localization is a well-established area of research within software engineering, traditionally concentrated on more conventional software systems. REST APIs, however, present distinct challenges that complicate fault localization due to their composition and operational dynamics. Additionally, the architecture of REST APIs often involves diverse artifacts that are not source code, such as configuration files, API specifications, and database schemas.

D. Key Insight or Idea

The primary objective of this research is to evaluate the effectiveness of existing fault localization techniques within the unique context of REST APIs, which feature a mix of code and non-code artifacts. This study will systematically apply established fault localization methods—Spectrum-Based Fault Localization (SBFL), Information Retrieval Fault Localization (IRFL), and Program Slicing—to a self-created dataset of REST API faults. This dataset will be meticulously developed to represent a wide range of faults typical in REST APIs and classified according to an existing comprehensive taxonomy derived from previous research.

This approach will enable us to assess the applicability of these fault localization techniques to REST APIs by determining how effectively they can identify and localize different types of faults. The analysis will provide insights into whether traditional fault localization methods are suited to the complexities of REST APIs, especially considering their unique structural and functional characteristics.

[[What is the high-level approach that you would like to explore to solve the problem? Why you feel you can succeed with that approach in 5 weeks?]]

E. Assumptions

[[What kind of assumptions will you need to make for your choice of solution?]]

F. Research questions

The primary research questions are formulated as follows:

- **RQ1:** What categories of faults (as per the existing taxonomy) are most effectively localized by current techniques?
- **RQ2:** Which fault localization techniques offer the highest accuracy and precision in the context of REST APIs?

[[What research question(s) will you answer?]]

G. Evaluation Dataset

H. Evaluation metrics

[[How will you know that you have solved the problem successfully? In other words, how will you evaluate your solution?]]

II. Background and Motivation

III. Related Work

[[R2Fix [1] and iFixR [2], use information retrieval-based fault localization (IRFL) that ranks suspicious program statements based on their similarity with bug reports.]]

IV. Approach

V. Evaluation

A. Dataset

B. Metrics

C. Experiment Procedure

D. Results

VI. Discussion and Threats to Validity

VII. Contributions

References

[1] C. Liu, J. Yang, L. Tan, and M. Hafiz, "R2Fix: Automatically generating bug fixes from bug reports," in *IEEE International Conference on Software Testing, Verification and Validation*, 2013, pp. 282–291.

[2] A. Koyuncu, K. Liu, T. F. Bissyandé, D. Kim, M. Monperrus, J. Klein, and Y. L. Traon, "iFixR: bug report driven program repair," in *27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE)*, 2019, p. 314–325.