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.

E. Assumptions

- *Assumption 1:* It is assumed that the faults in the dataset are correctly classified according to the existing comprehensive taxonomy.
- *Assumption 2:* Assume that the selected fault localization techniques are suitable for application to REST APIs, despite their original design for conventional software systems. This includes the assumption that these techniques can handle the unique challenges posed by REST APIs, such as dealing with non-code artifacts.

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?

G. Evaluation Dataset

H. Evaluation metrics

II. Background and Motivation

A. Importance of REST APIs

REST APIs (Representational State Transfer APIs) are central to modern web and cloud applications, serving as critical conduits for data exchange and system integration. They leverage standard HTTP methods like GET, POST, PUT, and DELETE to interact with networked resources, making them integral to the architecture of distributed systems. The scalability, reliability, and ease of integration provided by REST APIs facilitate seamless interactions between clients and servers, thereby enhancing the performance and flexibility of complex software ecosystems. Their widespread adoption underscores their importance in today's technology landscape, where rapid communication and data accessibility are paramount.

B. Challenges of Fault Localization in REST APIs

While REST APIs have simplified the development and management of modern applications, they introduce specific challenges that complicate fault localization. The architecture often involves non-code artifacts such as API specifications, configuration files, and database schemas, which are not traditionally addresses by fault localization techniques developed for code-centric applications.

C. Gaps in Current Research

Fault localization is a well-established research area within software engineering, focusing primarily on identifying the locations of faults in conventional software systems to reduce debugging time and enhance system reliability. However, the unique operational dynamics and architectural complexities of REST APIs pose new challenges that are not fully addressed by existing fault localization techniques. Previous studies have developed fault taxonomies for REST APIs which are helpful if the FL techniques can localize faults to understand what categories of faults can be localized. There remains a significant knowledge gap regarding the effectiveness of these techniques in accurately identifying and localizing faults in REST APIs, particularly those involving non-code artifacts.

D. Need for Targeted Research on Fault Localization in REST APIs

This research is motivated by the critical need to evaluate and understand the performance of existing fault localization techniques within the REST API context. By determining how effectively these techniques can identify and localize faults in REST APIs, developers can gain valuable insights that could lead to quicker and more accurate fault diagnosis. This is particularly important as even minor faults can escalate into major disruptions in REST API-dependent systems, affecting user experience and operational efficiency. Addressing this gap will not only contribute to the field by enhancing the robustness and reliability of REST APIs but also by supporting the development of more sophisticated tools and methodologies tailored to the needs of modern software architectures.

E. Research Objectives

The primary objective of this research is to systematically assess the applicability of established fault localization techniques—namely Spectrum-Based Fault Localization (SBFL), Information Retrieval Fault Localization (IRFL), and Program Slicing—to a curated dataset of REST API faults. This dataset will be developed to encompass a broad spectrum of typical faults in REST APIs and will be classified according to an existing comprehensive taxonomy. The insights gained from this study are expected to reveal whether traditional fault localization methods are suitable for the complex environments of REST APIs and may guide future efforts in refining these techniques or developing new approaches specifically designed for REST API ecosystems.

III. Related Work

IV. Approach

V. Evaluation

A. Dataset

B. Metrics

C. Experiment Procedure

D. Results

VI. Discussion and Threats to Validity

VII. Contributions