

Evaluating Automated Grey-Box Testing of Microservice Systems

Master's Thesis

Supervisor: Prof. Dr. Alexander Pretschner Advisor: Lena Gregor Email: {alexander.pretschner, lena.gregor}@tum.de Starting date: flexible

Context

A microservice is a small, independent service that is designed to perform a specific function within a larger software system. In a microservice architecture, an application is composed of a collection of these microservices, each running in its own process and communicating with other services via APIs or other lightweight mechanisms. Microservices are typically designed to be loosely coupled, with well-defined boundaries and minimal dependencies on other components, making them easier to deploy, scale, and maintain. [5]

Microservice systems can be tested on different testing levels: Unit tests regard everything within single microservices. Component tests test a service though its interfaces. Integration tests test the integration of at least two microservices and system tests test the completely deployed system. Giametti et al. [4] have proposed an approach for automated system testing of microservice systems which uses grey-box coverage criteria to optimize the test case generation.

Goal

The goal of this work is to evaluate the automated grey-box testing for microservice architectures approach by Giamattei et al. [4]. In order to evaluate this approach, the provided tool has to be used on the same microservice system as in [4] (or in a comparable MS system) and compared to at least one other approach to automatically generate test cases (f.e. [1, 2, 3]). Based on this, the approach should be evaluated (also in comparison to the other approach(es)) regarding multiple criteria, f.e.:

- · Fault detection/effectiveness (f.e. by fault seeding techniques)
- Applicability to other contexts (other interface types, other technologies, integration testing, component testing)
- Cost/efficiency
- Limitations
- "Ease" of usage

Working Plan

- 1. Familiarize yourself with the approach by [4].
- 2. Set up the tool for generating test cases proposed in [4] in a VM provided by the chair.
- 3. Set up the MS system to run the tool in a VM provided by the chair.
- 4. Set up at least one other approach to generate test cases in a VM provided by the chair.
- 5. Write technical report.
- 6. Create a plan and set up criteria for the evaluation.
- 7. Evaluate the tools based on the criteria mentioned above.
- 8. Write the thesis report.

Deliverables

- The VM(s) with runnable tools and generated test cases/ or GitHub repo including the tools with the correct settings.
- Technical report with comprehensive explanations on how to run and use the tools.
- Results of the evaluation.
- Final thesis written in conformance with TUM guidelines.
- Presentation of the work at the chair after the submission.

Application for Thesis

Please apply for this thesis topic with your CV, grade report, and a short motivation, why you are interested in this topic. Based on these documents, we will invite some students for a personal meeting to see if the topic fits the student.



Fakultät für Informatik Lehrstuhl 4 Software & Systems Engineering Prof. Dr. Alexander Pretschner

Boltzmannstraße 3 85748 Garching bei München

Tel: +49 (89) 289 - 17362 https://www4.in.tum.de Please note that some prior experience in the development of microservice systems and knowledge about software testing is highly desirable for this thesis.

References

- Arcuri, A.: RESTful API Automated Test Case Generation. In: 2017 IEEE International Conference on Software Quality, Reliability and Security (QRS). pp. 9–20 (Jul 2017). https://doi.org/10.1109/QRS.2017.11
- [2] Arcuri, A.: EvoMaster: Evolutionary Multi-context Automated System Test Generation. In: 2018 IEEE 11th International Conference on Software Testing, Verification and Validation (ICST). pp. 394–397 (Apr 2018). https://doi.org/10.1109/ICST.2018.00046
- [3] Atlidakis, V., Godefroid, P., Polishchuk, M.: RESTIer: Stateful REST API Fuzzing. In: 2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE). pp. 748–758 (May 2019). https://doi.org/10.1109/ICSE.2019.00083, iSSN: 1558-1225
- [4] Giamattei, L., Guerriero, A., Pietrantuono, R., Russo, S.: Automated grey-box testing of microservice architectures. In: 2022 IEEE 22nd International Conference on Software Quality, Reliability and Security (QRS). pp. 640–650. IEEE (2022)
- [5] Lewis, J., Fowler, M.: Microservices (Mar 2014), https://martinfowler.com/articles/ microservices.html

Fakultät für Informatik Lehrstuhl 4 Software & Systems Engineering Prof. Dr. Alexander Pretschner

Boltzmannstraße 3 85748 Garching bei München

Tel: +49 (89) 289 - 17362 https://www4.in.tum.de