[Extended Abstract] Scenario-based Resilience Evaluation and Improvement of Microservice Architectures: A Case Study

Dominik Kesim
dominik.kesim@gmail.com
University of Stuttgart, Germany

Lion Wagner
st148345@stud.uni-stuttgart.de
University of Stuttgart, Germany

Joakim von Kistowski
joakim.vonkistowski@datev.de
DATEV eG, Germany

Sebastian Frank
sebastian.frank@iste.uni-stuttgart.de
University of Stuttgart, Germany

Alireza Hakamian
mir-alireza.hakamian@iste.uni-stuttgart.de
University of Stuttgart, Germany

André van Hoorn
andre.van-hoorn@iste.uni-stuttgart.de
University of Stuttgart, Germany

Context

Modern distributed systems are expected to be resilient against changes, e.g., concerning workload, failures, deployments, and associated automated architectural reactions such as auto-scaling and other resilience mechanisms. It is difficult to identify the impact and root cause of severe quality-of-service (QoS) degradations or major service outages. Therefore, it is necessary to evaluate system resilience by identifying and assessing scenarios that may cause QoS degradation or bring the whole system down. After the interpretation of the assessment result, e.g., based on architectural analysis and resilience (aka chaos) tests, the software architects suggest improvements through resilience patterns. The Architecture Trade-off Analysis Method (ATAM) [1] is an established technique for evaluating software architectures' quality. The core idea is to characterize quality requirements into quality scenarios, each capturing the stimuli to which the architecture has to respond, the architectural decisions that impact achieving the quality requirement, and a measurable response and quality metric.

Objective

In an industrial setting, we aimed to leverage ATAM to identify resilience scenarios for a business-critical microservice-based system, derive executable chaos experiments from the scenarios, and develop a resilience/chaos test infrastructure that can be used for continuous resilience testing.

Method

We built on our previous work [2] (also presented at last year's SSP) on applying risk assessment techniques (e.g., Fault Tree Analyses) to identify and prioritize resilience scenarios, combining it with ATAM. Finally, we automated the execution of the scenarios for the case study system using the ChaosToolkit [3].

Result

Our scenario-based resilience assessment has the following steps,

1. We set up a one-day workshop, including the system's stakeholders, to systematically identify risks and hazards that cause QoS degradation. We invited stakeholders of the system that each has a different role, including product owners, software architects, quality engineers, and developers. The objective is to identify scenarios that lead to QoS degradation and downtime. We formulated 13 resilience scenarios according to the scenario template suggested in ATAM. According to the template, our resilience scenario consists of five parts, i.e., stimuli, artifact, environment, response, and response measure. Load spike is one type of stimulus in our scenarios that we modeled using LIMBO [4].

2. After collecting all resilience scenarios, we used ChaosToolkit and the LIMBO-based load generator to automate the scenario execution. The automation involves implementing the stimuli, candidate artifact, and response according to the resilience scenario. We assess the response measure by analyzing the QoS metrics measurements, e.g., error rates and response times.

3. After executing each resilience scenario, we applied a suitable resilience pattern that researchers and practitioners proposed in academia and industry. We re-executed each automated resilience scenario to validate if the suggested resilience pattern improved the system's overall resilience by comparing QoS's behavior before and after the resilience pattern.

Conclusion

Our experience of using ChaosToolkit reveals that chaos engineering lacks systematic identification of possible root causes of QoS degradation and outage. Therefore, we re-used the risk assessment methods. Scenario-based resilience assessment helps to precisely specify and later on quantify the behavior of a QoS.

Talk Outline and Further Resources

In this talk, we report on the mentioned case study by outlining our steps in specifying and assessing resilience scenarios quality, lessons learned, and key conclusions. A detailed description of the case study (with a link to artifacts) can be found in [5], and a publication is in preparation.

References

[1] Bass, Len, Paul Clements, and Rick Kazman. "Software Architecture in Practice." (2012).

[2] D. Kesim, A. van Hoorn, S. Frank, and M. Häussler, "Identifying and prioritizing chaos experiments by using established risk analysis techniques," in Proceedings of the 31st International Symposium on Software Reliability Engineering (ISSRE 2020), 2020, accepted.

[3] Chaos toolkit. [Online]. Available: https://github.com/chaostoolkit (2020).

[4] v. Kistowski, Jóakim, Nikolas Herbst, and Samuel Kounev. "LIMBO: a tool for modeling variable load intensities." Proceedings of the 5th ACM/SPEC international conference on Performance engineering. 2014.

[5] D. Kesim, L. Wagner, "Scenario-based Resilience Evaluation and Improvement of Microservice Architectures by Applying Automated Chaos Experimentation." [Online]. Available: https://drive.google.com/file/d/1kvNj37c2dw_uXl6TwKb3d5GDYE9oKEfY/view?usp=sharing (2020).