Review of Microservices: A Performance Tester’s Dream or Nightmare

Author: Simon Eismann (University of Wuerzburg)

Cor-Paul Bezemer (University of Alberta)

Weiyi Shang (Concordia University Montreal)

Dušan Okanović (Novatec Consulting GmbH)

Reviewer: Ratun Rahman (Islamic University of Technology: 170042011)

Summary: Microservices have become hugely popular in recent years. Mainly, because they come with a couple of benefits that are super useful in the era of containerization and cloud computing. You can develop and deploy each microservice on a different platform, using different programming languages and developer tools. So it is really useful to handle large-scale softwares. As microservices getting popular day by day, people also need to know the headaches that they brings especially if you don’t anticipate them. In this paper, it is mostly described about the most important benefits and challenges that comes while performing microservice testing. This paper also shows us the different directions of performance testing for more stable performance and reliability. Performance testing is in general a testing practice performed to determine how a system performs in terms of responsiveness and stability under a particular workload. Microservices has some characteristic which can be described as ‘a performance tester’s dream’. Microservices are containerization and so tests can be deployed’ executed and reproduced easily. They are also build independent and does not need different dependencies as well. They can be implemented onto different platform with very little knowledge or sometimes without any knowledge of implementation at all. They also fit into fast release cycle of DevOps which reduce the gap between performance testing and DevOps. Despite having such advantages, it also bring some ‘nightmares’ along with it. Microservices execution on environment are not stable even when provisioned instances numbers are fixed. Also there is significant differences in response time in ters of different scenarios that will have negative impact on user-perceived performance. It is also very hard to detect small changes when performing regression testing using microservices. Enough repetitions of performance tests should be ensured to deal with this problem which will require more resources. This article is made using a single system under test (TeaStore: one of the few available reference microservice applications for performance testing) and a single cloud environment (GKE: one of the popular engines for deploying microservices). So in conclusion, the question of ‘should I use microservices?’ completely depend on the architecture plan for that particular project case, scale and requirements. Your own context should play a huge part in deciding if you want to go down the microservices path. People think about microservices as only about technology but in order to establish, you have to change your organization, team, structure.

Strength: This paper not only perfectly described the positive and negative impacts on microservices but also proposed new, more stable performance metrics to improve qualities in microservice-based softwares.

Weakness: This paper had not described clearly what a performance tester is, his roles and responsibilities.