Testing Strategies for Microservices- Manual to Automated Testing

Hareesh Kumar Rapolu

hareeshkumar.rapolu@gmail.com

Abstract

The research paper has critically evaluated the significance of microservices. It is a very important part of the software development process since it can help to bring in new updates to an existing application or create a new one. The developers regularly test the microservices using different kinds of manual and automated testing. The research paper has also analysed why the testing of microservices is essential. In the final portion of the study, both the manual and automated methods of microservice testing have been thoroughly presented.

Keywords: Microservices, API, updates, microservice architecture, manual, automation

I. INTRODUCTION

Microservices can be described as an essential part of developing a software. It is a particular style of software development where larger applications are broken down into smaller, independent parts. These small and independent services communicate and transmit data over different application processing interfaces (APIs). These microservices are regularly tested in order to ensure that they perform their delegated tasks in a proper manner. This research paper will explore the fundamental concept of microservices. The study will further critically analyse the different reasons why microservices are monitored and tested. In the final portion of the research paper, it will identify the various types of microservice testing methods such as manual to automated testing.

II. EXPLORING MICROSERVICES

In the context of software development, microservices form an integral part. It is a programming architecture that helps different developers to create innovative, flexible and scalable applications that can be used for different purposes. The different microservices that are part of a single application execute some specific functions. Therefore, the microservices help the application to perform in the most efficient manner[1]. It is extremely important to note that a microservice connects with another in order to communicate with them. This communication is maintained with the help of an API. It allows the developers to write different kinds of services on technology with different languages. Therefore, it is evident that the flexible nature of each microservice helps developers to build a proper application without much hindrances[2]. In this context, it needs to be understood that the term microservice does not refer to its physical size, but rather to the singular function that is performed by an individual microservice. These microservices are properly monitored by the developers and this data regularly. The proper utilisation of the microservices can help developers to speed up the process of creating a new application or integrating some new features to an already existing application. Since the microservices can be updated and independently published by the developers, it allows them to bring more frequent software updates. The developers can calculate the throughput by dividing the total number of requests by the amount of time that is taken. The average time between two consecutive microservice failures can also be calculated by dividing the total amount of time by the number of failures. This is very important for testing the reliability of each microservice.

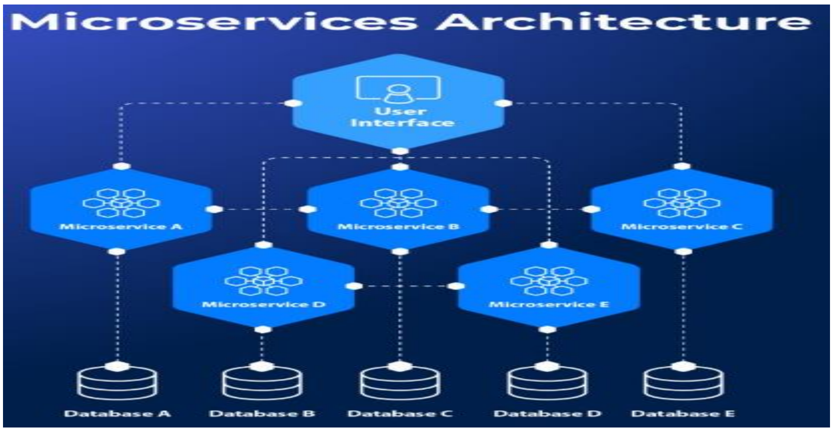

Figure 1: Microservices architecture

III. UNDERSTANDING THE NEED FOR MICROSERVICE TESTING

The software developers implement different ways to test the microservices. This evaluation is mainly done in order to ascertain that an individual microservice is functioning properly within a microservices infrastructure[3]. In this methodology, the developers treat each microservice as an isolated unit. This provides them enough scope and opportunities to assess their performance on an individual level. If a microservice is properly optimised, it can be seamlessly integrated with other services. There are some key reasons why a microservice is tested using a number of methods. These reasons can be further analysed in order to gain a complete understanding.

Rapid development

This is one of the major reasons why microservices are regularly tested. If a developer is able to locate and fix problems within a certain service, it can improve the overall software development cycles. The updates for such applications can also be deployed more efficiently[4]. Additionally, regular monitoring and testing of the microservices help to identify problems prematurely. It can substantially reduce the overall cost that is required to manage the microservices and build new applications.

Enhanced fault isolation

As mentioned earlier, each microservice is treated as an independent entity by the developer. Hence, if a bug is found in a microservice, it will not hamper the entire architecture[5].

Scalability

That developers are able to upscale or downscale the microservices on the basis of their demands. These demands can be properly identified if the microservices are regularly tested.

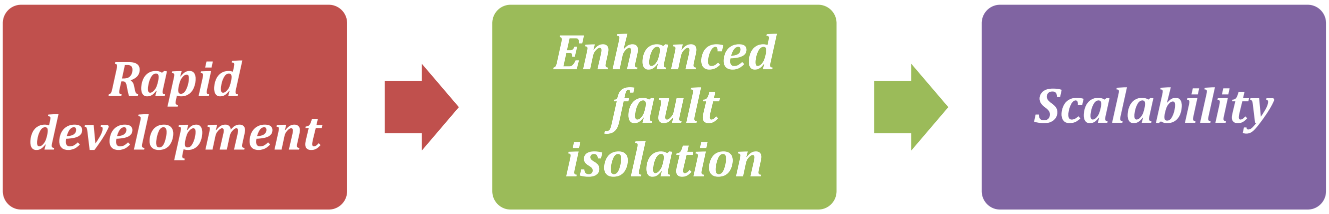

Figure 2: Reasons for testing microservices

IV. ANALYSING THE TWO MAIN KINDS OF MICROSERVICE TESTING

The developers utilise different ways to test the microservices. The manual and automated testing are the most important strategies that are adopted by them. These two primary methodologies can be further explained.

Manual testing

This type of quality assessment technique is used by different software developers to ensure that each individual microservice is functioning in an optimal manner. The manual testing methodology is further helpful in assessing the stability of the performance of each microservice. The developers use different scenarios to trigger the microservice[6]. A number of data is put into each microservice to observe their reaction. However, the manual type of microservice testing has a rather limited scope. This is mainly due to the complexity of the microservices architecture. There are certain challenges that are related to this type of microservice testing. It can be really difficult to properly manage and co-ordinate the interactions between different microservices manually. In addition, it can take a lot of time to manually test each microservice by the developer.

Automated testing

In order to mitigate the challenges of manual testing, developers opt for evaluating the microservices in an automated manner. The developers and testers utilise the different kinds of AI and machine learning technologies to assess the performance of a microservice. This is how the task of testing a complicated micro services environment can be done within a stipulated time period[7]. It also helps to provide accurate results which is absolutely crucial for software development and maintenance.

V.CONCLUSION

From the above discussion, it can be concluded that microservices form a very important element in the software development process. However, it is evident that the developers need to test them regularly in order to ensure that they are performing in the right manner. Both manual and automated testing methods are utilised bydevelopers and testers to evaluate the efficiency and stability of the microservices.

Abbreviations and acronyms

API - Application Processing Interface

Units

Throughput= Transactions per second (TPS)
Rate of microservice failure= percentage (%)
Mean Time Between Failures= Hours, days, seconds

Equations

MTBF= Number of Failures/Total Time
Throughput= Time/Total Number of Requests

REFERENCES

[1] A. Avritzer*et al.*, "Scalability Assessment of Microservice Architecture Deployment Configurations: A Domain-
based Approach Leveraging Operational Profiles and Load Tests," *Sciencedirect.com*, vol. 165, Feb. 2020, doi: https://doi.org/10.1016/j.jss.2020.110564. Available: https://www.sciencedirect.com/science/article/am/pii/S016412122030042X

[2] F. Tapia, M. Á. Mora, W. Fuertes, H. Aules, E. Flores, and T. Toulkeridis, "From Monolithic Systems to Microservices: A Comparative Study of Performance," *Applied sciences*, vol. 10, no. 17, Aug. 2020, doi: https://doi.org/10.3390/app10175797

[3] G. BLINOWSKI, A. OJDOWSKA, and A. PRZYBYŁEK, "Monolithic vs. Microservice Architecture: A Performance and Scalability Evaluation," *IEEE Access*, vol. 10, Feb. 2022, doi: https://doi.org/10. 1109/ACCESS.2022.3152803

[4] I. Ghani, W. M. N. Wan-Kadir, A. Mustafa, and M. I. Babir, "Microservice Testing Approaches: A Systematic Literature Review," *International Journal of Integrated Engineering*, vol. 11, no. 8, Sep. 2019, doi: https://doi.org/10.30880/ijie.2019.11.08.008

[5] J. Bogner, J. Fritzsch, S. Wagner, and A.
Zimmermann, "Industry practices and challenges for the evolvability assurance of microservices," *Empirical Software Engineering*, vol. 26, no. 5, Jul. 2021, doi: https://doi.org/10.1007/s10664-021-09999-9

[6] M. Camilli, A. Janes, and B. Russo, "Automated test-based learning and verification of performance models for microservices systems," *Journal of Systems and Software*, vol. 187, p. 111225, May 2022, doi:https://doi.org/10.1016/j.jss.2022.111225

[7] S. Newman, *Building Microservices*, 1st ed. O'Reilly, 2015. Available:
https://book.northwind.ir/bookfiles/building-microservices/Building.Microservices.pdf