Project Title: Scalability analysis of microservice architecture on a particular software application

Submitter: Tural Mehtiyev

Reviewer: Aytan Gurbanova

Summary

My understanding is that Tural wants to write an application in microservices architecture to conduct a comprehensive analysis of the scalability of the microservices architecture. The goal is to assess the system's ability to manage greater loads without sacrificing performance. The study will explore a variety of scalability characteristics, including performance, load scalability, resource usage, capacity, and fault tolerance.

According to the report, he aims to achieve that goal by considering metrics such as response time, throughput, resource usage, error rates, latency, system utilization, service time, scalability ratio, queue length, and availability.

He also mentions how the issue is addressed today by stating load testing, code reviews, and architectural. As for limitations, time and necessary skills for load tests, trouble estimating future loads, and difficulties detecting bottlenecks in complex systems are mentioned.

Strong Parts

Overall, the proposal is well chosen and offers a lot to learn throughout the process. Both microservices and software scalability are hot topics in the current tech industry. Additionally, he did a good job of outlining the process. The report provides necessary details for understanding the proposal. The mentioned attributes and metrics make sense for evaluating scalability. I also believe that the listed possible risks are valid for the project, and knowing them beforehand will be helpful in the development process, as well as including ways to prevent them from occurring.

To be Improved

Scalability testing of an experimental application, in my opinion, is not an easy task. Providing data load to the application can be a lot of work, especially because real-world applications with real-world transactions are more effective at uncovering scalability concerns. However, observing such issues in an experimental application may be difficult. Furthermore, while the project outlines what will be done, it does not go into detail on how it will be done. If performed by the same person who wrote the code, code review, which is described as part of the analysis, may not be effective. I think a second perspective is always good idea for code review. Lastly, to test scalability, it would be beneficial to establish benchmark metrics or best practices. Since this is only one software application, determining when good performance is achieved can be problematic, as well as subjective. The assessment can be effected from various factors, one example being insufficient testing.