

Abstract

Im Rahmen dieser Arbeit wird die Arbeit von Herren Tomak "Performance-Testing-Infrastruktur" (2020) [1] erweitert. In der genannten Arbeit wurde die Leistung der Fehlerverwaltung in einem Legacy-System, das von der Firma GS Electronics [2] entwickelt wurde, gemessen. Der wichtigste Bestandteil der Arbeit war die Nutzung von dem Load-Testing-Tool Locust [3] zur Evaluation und Loggen der Antwortzeiten des Systems. Ergebnisse der Arbeit sowie der Source-Code sind in einem open-source Repository [4] bereitgestellt. Locust ist ein einfach zu benutzendes Load-Testing-tool, das komplett in Python geschrieben ist und ermöglicht, eigene Tests in Python zu schreiben. Die Erweiterung der genannten Arbeit umfasst die Implementierung eines Web-Services, das Benutzern/Benutzerinnen die Möglichkeit anbietet, Load-Testing ihres gewählten zu testenden Systems nach dem Festlegen der Testspezifikationen und dem Hochladen der entsprechenden Test-Datei durchzuführen, wobei Locust bevorzugt wird. Außerdem muss der zu implementierende Web-Service einfach zu installieren und zu benutzen sein. Um dieses Ziel zu realisieren, wird der Kern des Web-Services als eine Openfaas-Funktion implementiert. Openfaas [5] ist ein Open-Source-Projekt (2016), das von Alex Ellis mithilfe mehrerer Mitwirkenden entwickelt wurde. Das Ziel des Projekts ist es, Serverless-Funktionen in einer der derzeit unterstützten Programmiersprachen zu schreiben und unabhängig vom Betriebssystem benutzen zu können, damit sich Entwickler/Entwicklerinnen nur auf die Entwicklung neuer bzw. bereits existierender Projekte konzentrieren, ohne ihre Arbeitsumgebung jedem einzelnen Projekt anpassen zu müssen. Die derzeit unterstützten Programmiersprachen [6] sind unter anderem Python, Java, C Sharp, Go und vieles mehr. In unserer Arbeit wird eine Openfaas-Funktion geschrieben, die das Ausführen, Loggen und Analysieren von Locust-Tests übernimmt und eine Benutzeroberfläche anbietet, wo Statistiken und Analysen aller laufenden bzw. fertig ausgeführten Tests in Echtzeit dargestellt werden, mit der Möglichkeit in Sicht, Testergebnisse zum Herunterladen bereitzustellen. Unser Ziel ist, das Konzept von Serverless-Funktionen in einem wichtigen Aspekt der Software-Entwicklung — Performance Testing — zu realisieren, und Benutzergruppen die einfache und effiziente Möglichkeit anzubieten, ihre eigenen Tests plattformunabhängig durchführen und in einer benutzerfreundlichen Benutzeroberfläche dargestellt bekommen zu können, ohne dass eigene Arbeitsumgebungen angepasst bzw. belastet werden.

References

1. J. Tomak, S. Gorlatch: Measuring Performance of Fault Management in a Legacy System: An Alarm System Study (2020).
<https://www.springerprofessional.de/en/measuring-performance-of-fault-management-in-a-legacy-system-an-/18809536>
2. GS. <https://www.gselectronic.com/>
3. Byström, C., Heyman, J., Hamrén, J., Heyman, H.: Locust. <https://docs.locust.io/en/stable/what-is-locust.html>
4. J. Tomak: Performance Testing Infrastructure (2020).
https://bitbucket.org/JT1337/locust_scripts/src/master/
5. A. Ellis: OpenFaaS Serverless Functions. <https://github.com/openfaas/faas>
6. Openfaas supported languages. <https://github.com/openfaas/templates>