

Abstract

Im Rahmen dieser Arbeit wird das Projekt von **(Tomak 2020)** ¹ ergänzt, in dem das Open-Source Load-Testing-Tool **Locust** ² dazu verwendet wurde, die Performance der Alarmempfangssoftware der Firma **GS electronic GmbH** ³ zu messen. Jedoch ist die Verwendung von dem Load-Testing-Tool in dem Projekt nicht intuitiv, da alles über die Kommandozeile läuft. Dafür ist die Entscheidung getroffen, das Projekt um einfache Bedienheit und Benutzerfreundlichkeit zu erweitern.

Die Erweiterung umfasst die Implementierung eines Web-Services, der dem Benutzer eine Funktion anbietet, mit der er eine Testspezifikation für ein spezifisches Load-Testing-Tool hochladen kann, wobei Locust bevorzugt wird. Der Web-Service führt dann das Load-Testing-Tool aus und testet somit das in der Testspezifikation angegebene SUT (System-under-Test).

Außerdem muss der Web-Service einfach zu installieren und zu benutzen sein. Um dieses Ziel zu erreichen, wird der Kern des Web-Services als eine **OpenFaaS-Funktion** ⁴ implementiert, die auf einer **Multipass** ⁵ virtuellen Maschine läuft.

Die OpenFaaS-Funktion übernimmt das Ausführen und Analysieren von Locust-Tests und bietet eine Benutzeroberfläche an, wo Statistiken und Analysen aller laufenden bzw. fertig ausgeführten Tests in Echtzeit dargestellt werden.

Anschließend, kann der Benutzer die aufgezeichneten Performance-Metriken und Analysen herunterladen.

Ziel unserer Arbeit ist es, das Konzept von Serverless-Funktionen in einem wichtigen Aspekt der Software-Entwicklung — Performance Testing — zu realisieren, und somit

Testern die einfache und effiziente Möglichkeit anzubieten, ihre eigenen Tests plattformunabhängig über eine Benutzeroberfläche durchzuführen, ohne dass eigene Arbeitsumgebungen angepasst bzw. belastet werden.

Am Ende soll der Benutzer den Web-Service mit wenigen Befehlen installieren und starten.

Referenzen

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. Byström, C., Heyman, J., Hamren, J., Heyman, H.: Locust. <https://docs.locust.io/en/stable/what-is-locust.html>
3. GS. <https://www.gselectronic.com/>
4. A. Ellis: OpenFaaS Serverless Functions, Made simple. <https://www.openfaas.com/>
5. Multipass orchestrates virtual Ubuntu instances. (2019, Dezember).
<https://multipass.run/>