

Michael Kaiser, Martin Seifert: Last- und Performance-Test mit JMeter

JMeter ist ein wichtiges OpenSource-Tool um zu testen, wie sich eine Web-Anwendung unter Last verhält. Dabei können Schwächen der Anwendung aufgedeckt werden wie z.B., ob sie Thread-Save ist oder ob sich Threads gegenseitig beeinflussen. Dies kann i.d.R. allerdings nicht in JUnit-Tests geprüft werden, weshalb der Beschuss der Anwendung aus JMeter heraus immer sehr wichtige weitere Erkenntnisse liefern kann. Ausserdem liefert JMeter erste Aussagen darüber, wie es mit der Performance einer Anwendung bestellt ist. Durch die Simulation einer Vielzahl von Benutzern kann ermittelt werden, wie sich die Antwortzeiten unter diesen Bedingungen entwickeln. Auf diese Weise können in sehr zentralen Feldern Tests im Vorfeld der Inbetriebnahme stattfinden, die potentiell gefährliche Probleme bereits frühzeitig sichtbar machen und einer Lösung zuführen zu können.

Inhalt dieses Abschnitts der Testtage sollte sein, zunächst die Grundlagen dieser Art des Testens darzustellen. Im praktischen Teil soll zunächst JMeter installiert werden. Anschließend werden die einzelnen Schritte des Lasttestens anhand der zur Verfügung gestellten Anwendung durchexerziert. Dazu gehört zunächst die Konfiguration von JMeter. Danach werden geeignete Lasttestfälle aufgezeichnet. Diese Testfälle in Rohform werden dann in darauf folgenden Schritten weiter verfeinert. Hier werden zunächst einzelne spezifische Einträge der aufgezeichneten Tests gelöscht oder durch Variablen ersetzt, so dass sie dann in den folgenden Testläufen ein passendes Verhalten des JMeters als eine Menge von parallel arbeitenden Benutzern erlauben. Es werden die zur Verfügung stehenden Mittel zur Prüfung der Ergebnisse der Anfragen in die Beispiel-Testfälle eingefügt. Für den Betrieb der Lasttests sind dann außerdem noch die Parameter für den Grad der Parallelität, der Abstände zwischen den Thread-Startzeitpunkten und den Wiederholungen sowie Startzeitpunkt und Dauer einzustellen. Schließlich wird noch darauf eingegangen, welche Ergebnisse ein Lasttestlauf produziert, wie diese Ergebnisse ausgewertet werden und wie mit diesen Erkenntnissen Verbesserungen vorgenommen werden können. Dazu kann es evtl. nötig sein, die Beispielanwendung mit geeigneten Fehlern zu bestücken.

Über die Vortragenden

Michael Kaiser hat an der Universität Stuttgart sein Diplom in Informatik gemacht. Er arbeitet seit fast 10 Jahren als Softwareentwickler. Als Senior Software-Ingenieur, Consultant und Mitgründer der stuttIT ist er bei der VersIT in einem JEE-Projekt eingesetzt. Dabei liegt ein wichtiger Schwerpunkt seiner Tätigkeit im Bereich des Testens und der QS.

Martin Seifert ist bei der VersIT als Softwareentwickler beschäftigt und befasst sich seit 1998 hauptsächlich mit Qualitätssicherung und Testautomatisierung. Er ist Autor Artikel zum Thema Testen und Projektplanung.