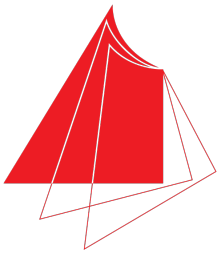


Hochschule Karlsruhe
Technik und Wirtschaft
UNIVERSITY OF APPLIED SCIENCES

Studiengang Informatik (Master)

Last-Test von Web-Seiten mit JMeter

Seminararbeit - Ausarbeitung

Daniel Schäfer (60118)

Betreuer: Prof. Dr. Holger Vogelsang

Sommersemester 2018

Inhaltsverzeichnis

Abbildungsverzeichnis

1 Einführung

Mit dieser Ausarbeitung zur Seminararbeit wird das Performance und Last-Test Programm Apache JMeter ausführlich unter die Lupe genommen. Es werden seine Funktionen und Möglichkeiten erläutert und kurz Alternativen dazu erwähnt. Zusammenfassend werden Vor- und Nachteile aufgelistet, sowie ein Fazit gegeben.

1.1 Motivation

Man erlebt es immer wieder, dass Webseiten unter zu hoher Last in die Knie gezwungen werden. Sei es nun Amazon, Netflix und Konsorten durch DDoS-Angriffe [3], oder auch hochschulinterne Seiten, die durch gegebene Anmeldefristen eine hohe Anzahl von gleichzeitigen Benutzern zu bewältigen haben.

Gerade letzteres Szenario lässt sich im voraus gut vermeiden, da man die grobe Anzahl der Studierenden kennt und somit präventiv die Webseite auf die eingehende Last testen kann. Dabei werden sehr viele nebenläufige Anfragen an einen Anwendung gestartet und die Response Zeiten ausgewertet. Diese Art von Tests nennt man Last oder Performance-Tests und wird in die Klasse der Systemtests kategorisiert [7].

Das frei verfügbare, unter Public License Key stehende Java Programm JMeter bietet diese und weitere Funktionalität um die Leistung einer Webseite bis ins kleinste Detail zu analysieren, testen, auswerten und visualisieren.

1.2 Manuelle vs automatisierte Tests

Bevor es an das eigentliche Thema JMeter geht, gibt es einen kleinen Exkurs in die sogenannten Systemtests. Diese wurden im vorherigen Abschnitt kurz erwähnt und haben die Besonderheit, dass sie sowohl manuell, als auch automatisiert ausgeführt werden können.

1.2.1 Manuelles Testen

Beim manuellen Testen werden bestimmte Aktionen und Request von mehreren Benutzern nach bestimmten Vorgaben gestartet und die resultierenden Ergebnisse protokolliert. Es liegt auf der Hand, dass diese Art des testens ziemlich zeit -und resourcenaufwändig ist. Zusätzlich muss das Personal, sprich die Tester organisiert, betreut und verpflegt werden.

Falls es sich dabei um die Programmierer selbst handelt, stehen diese im Zeitraum auch nicht für andere Tätigkeiten zur Verfügung. Der Faktor Mensch spielt natürlich bei dieser Art der Tests eine Rolle, wodurch nicht ausgeschlossen werden kann, dass Fehler während der Testphasen gemacht und diese entsprechend nicht erfasst werden. [4, S. 11]

Trotz dieser Nachteile sollte man auf zusätzliche manuelle Tests[1] einer Anwendung nie verzichten, da eine „reale" Person über den Tellerrand schauen kann und es dadurch möglich ist, diverse Bugs ausfindig zu machen, die mit dem eigentlichen Test unter umständen gar nichts zu tun haben.

1.2.2 Automatisiertes Testen

Automatisiertes Testen funktionieren immer dann recht gut, wenn häufige Wiederholungen auftreten. Beispielsweise bei Regressiontests. Aber auch bei Lasttests ist die Testautomatisierung ein gängiges Verfahren. Die Tests laufen in erster Linie mit Software, was zur Folge hat, dass sie ziemlich statisch sind und dadurch beispielsweise keine Ästhetik geprüft werden kann, wie dies etwa bei menschlichen Testpersonen der Fall wäre.

Jedoch haben automatisierte Tests den großen Vorteil, dass sie kosteneffizienter sind. Man benötigt lediglich ein Budget für etwaige Lizenzgebühren der Software oder für den Support. Des weiteren ist man durch Testsoftware in der Lage, sehr viele Benutzer gleichzeitig zu erzeugen und parallel auf ein System zugreifen zu lassen. Mit Testskripten sogar rund um die Uhr [6]. Dadurch werden Systeme bis zur ihren Grenzen und darüber hinaus getestet.

2 Apache JMeter

Wen man sich nun für die Testautomation entschieden hat, steht man vor der Wahl einer entsprechenden Software. Diese befinden sich in einer Preisspanne von kostenlos bis hin zu einem fünfstelligen Bereich. Die Wahl hängt letzten endes von den eigenen Anforderungen ab [4, S. 15]. Da sich die Arbeit auf JMeter bezieht richten wir unser Augenmerk auf dise spezielle Software.

Wie man in Abbildung 1 erkennen kann, erscheint Apache JMeter mit

[1] Das sogenannte User Acceptance Testing (UAT) ist gerade im finalen Entwicklungsstadium einer Anwendung unabdingbar. [9]

seinen „Metal-Look-and-Feel“ [8] Widgets wie das Relikt aus einer längst vergangenen Zeit. Man sollte sich jedoch von der veralteten und trägen Swing-basierten GUI nicht täuschen lassen, denn hinter der Oberfläche steckt ein mächtiges Werkzeug, mit dem man alle möglichen Arten von Tests erstellen und ausführen kann. [5]

Tatsächlich stammt JMeter aus einer Zeit als das Web und Application Server noch in den Kinderschuhen steckten. Ursprünglich wurde es entwickelt um den Application Server Tomcat zu testen. Seitdem wurde das Programm fortlaufend weiter entwickelt, so dass man heute in der Lage ist diverse Tests zu realisieren. Von verteilten Tests in der Cloud, über das Testen von dynamischen Webseiten bis hin zu Javas JUnit Tests. [2]

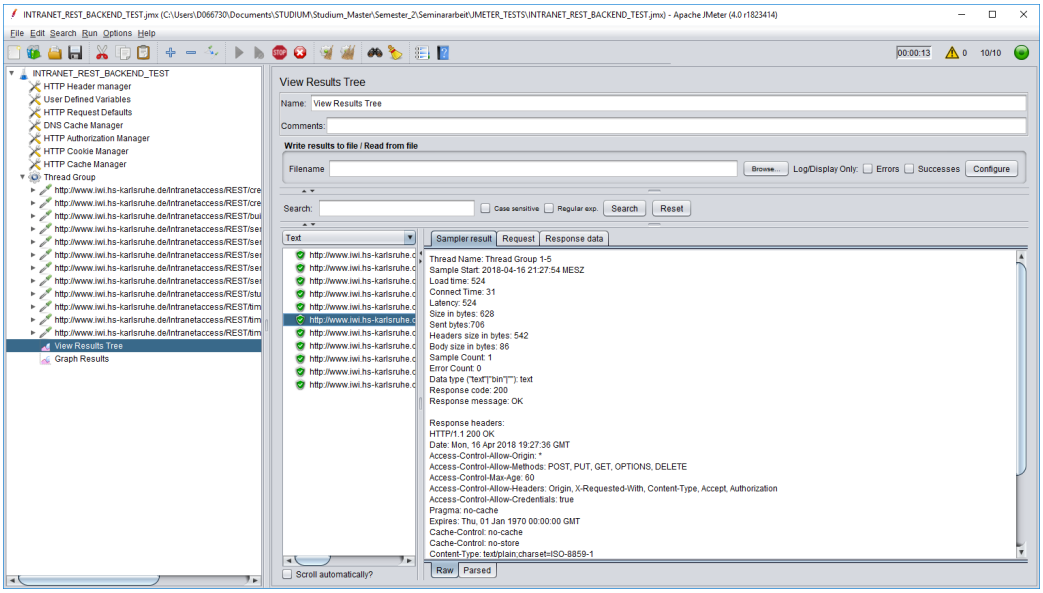


Abbildung 1: Apache JMeter GUI

Apache JMeter wurde in Java geschrieben und kann durch seine plattformunabhängigkeit unter jedem beliebigen Betriebssystem, welches eine Java Runtime Environment (JRE) installiert hat, verwendet werden. Neben der GUI kann man Apache JMeter auch mit der Kommandozeile bedienen. Dies ist gerade bei speicherintensiven Lasttests empfehlenswert, da die GUI schnell an ihre Grenzen kommt und sich die Testergebnisse nicht mehr genau erfassen lassen. Mehr zum Thema Kommandozeile im Abschnitt JMeter Command Line.

Apache JMeter bietet zusätzlich eine gut dokumentierte API an, mit der es möglich ist das Programm selbstständig zu erweitern.

2.1 Historisches

Ursprünglich wurde Apache JMeter von Stefano Mazzocchi, seiner Zeit Entwickler bei der Apache Software Foundation, programmiert um die Performance von Apache Tomcat (damals noch Apache JServ) zu testen. Kurz danach wurde das Programm neu entworfen und mit einer verbesserten GUI und zusätzlichen Möglichkeiten von Funktionstests ausgestattet. Im Jahr 2011 wurde JMeter zu einem sogenannten Top Level Apache project, was eine offizielle Homepage und ein Projekt Management Committee mit sich brachte [1].

Mitlerweile ist Apache JMeter ein wichtiger Bestandteil von Performance Tests in sehr vielen Firmen geworden. Großkonzerne wie SAP oder 1&1 verwenden regelmäßig JMeter um die Verfügbarkeit ihrer Produkte unter hoher Last zu prüfen.

2.2 Was kann JMeter

Apache JMeter dient in einer Client/Server Landschaft als Client und kann dadurch Anfragen an bestimmte Anwendungen absetzen. Dadurch erhält man unter anderem die Responsezeit, Responsemesage oder aber auch den Speicherverbrauch.

Anfragen können sowohl an statische als auch an dynamische Resourcen erfolgen. Darunter fallen unter anderem satische Dateien, Servlets, FTP Server, Datenbanken, Java Objekte und Skripte. Um diese Anfragen in einer entsprechend großen Anzahl abzufeuern, bietet das Programm die Simulation von vielen gleichzeitigen Benutzern an. Diese Threads lassen sich in einzelne Thread Gruppen unterteilen.

Nach den Tests bietet JMeter ein HTML Dashboard an, in dem sehr viele Informationen über die Anfragen und Ergebnisse in Tabellen und Statistiken grafisch aufbereitet dargestellt werden. Abbildung 2 zeigt einen Ausschnitt dieser Möglichkeit, die im Abschnitt xxx genauer untersucht wird.

skript erstellen und aufzeichnen In Java geschriebene JUnit Tests importieren und ausführen.!!

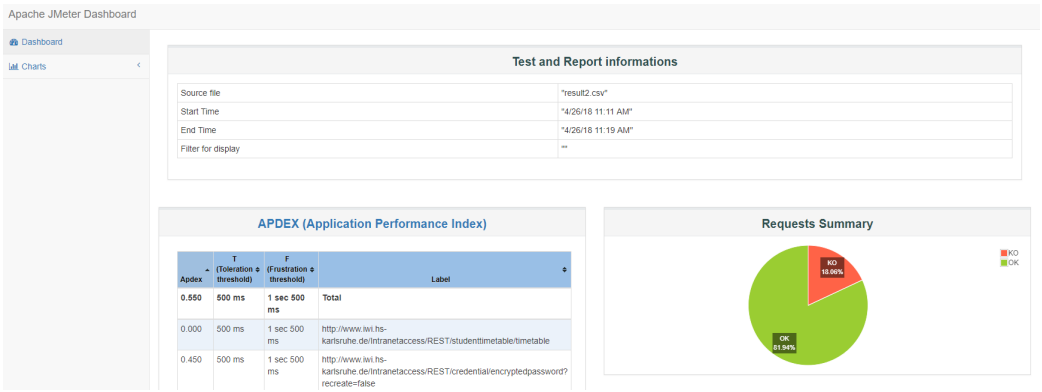


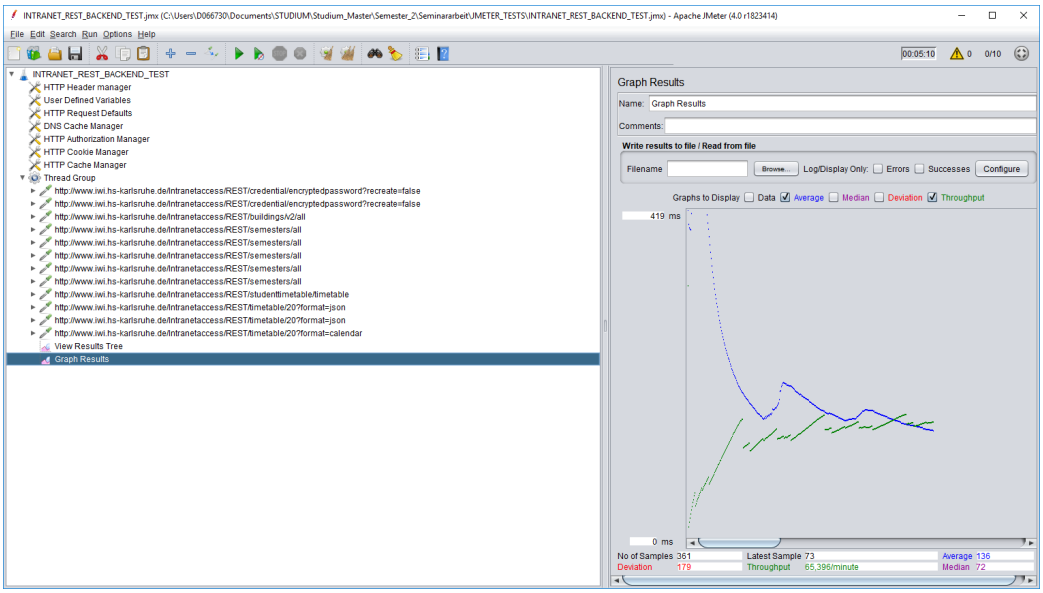
Abbildung 2: HTML Dashboard Report

2.3 Installation

2.3.1 Setting up the environment

2.3.2 Running JMeter

2.4 JMeter GUI



2.5 JMeter Command Line

GUI verbraucht viel Speicher. Nicht empfohlen für große Lastentests Command Line ist geeignet für die Integration in andere Systeme wie Jenkins/-Continious Integration

2.5.1 Ausführen des Console Clients

Zuerst sollte ein Testscript verfügbar sein. Man kann hier ein vorhandenes File verwenden oder in JMeter selbst ein neues *jmx-File erzeugen. Minimale Voraussetzungen sind eine Thread Group sowie ein Sampler wie etwa ein HTTP Request mit entsprechender URI.

JMeter GUI beenden und Command Line starten. Danch in das /bin Verzeichnis von JMeter navigieren. Dort startet man den Test via dem Befehl:

```
1  jmeter -n -t [jmx file] -l [results file]
```

In der folgenden Tabelle sieht man einige häufig verwendete Parameter und ihre Bedeutung. Um eine Übersicht aller Parameter zu erhalten, kann man in der Console den jmeter -? ausführen.

Parametername	Bedeutung
-n	non gui mode
-t	location of jmeter script - combined with -n
-l	location of result file
-?	alle Parameter an die zur Auswahl stehen
-h	Beispielbefehle, die man verwenden kann
-L	Log level - Wann soll etwas geloggt werden
-e	Um html Reports zu erzeugen
-o	Location of the output folder - combined with -e or -g
-g	html report wird aus einem csv file erzeugt

Tabelle 1: Befehle der JMeter Command Line

3 Der Testplan

Was ist überhaupt ein Testplan

3.1 Elemente eines Testplans

Welche Elemente kann ein Testplan enthalten
1 Testplan anlegen

3.1.1 Thread-Groups

Ähnlich wie JUnit. Hier möglichkeiten setup teardown neben den eigentlichen gruppen Gleichzeitige Benutzer usw.

3.1.2 Sampler

Sind die eigentlichen Tests

3.1.3 Listener

Elemente die Informationen über die Performance Tests enthalten Zum abfragen bzw. ergebnisse visualisieren
werden verwendet um ergebnisse und metriken der tests anzuzeigen

3.1.4 http request

3.1.5 Assertions

Mit jmeter lassen sich assertions testen. Dazu wird der response von einem request untersucht assert 200. assert 500 etc.

4 Lasttests von Webseiten

5 Funktionale Tests

6 Auswertung HTML Dashboard

APDEX index. selbst bearbeiten in /bin/user.properties file.

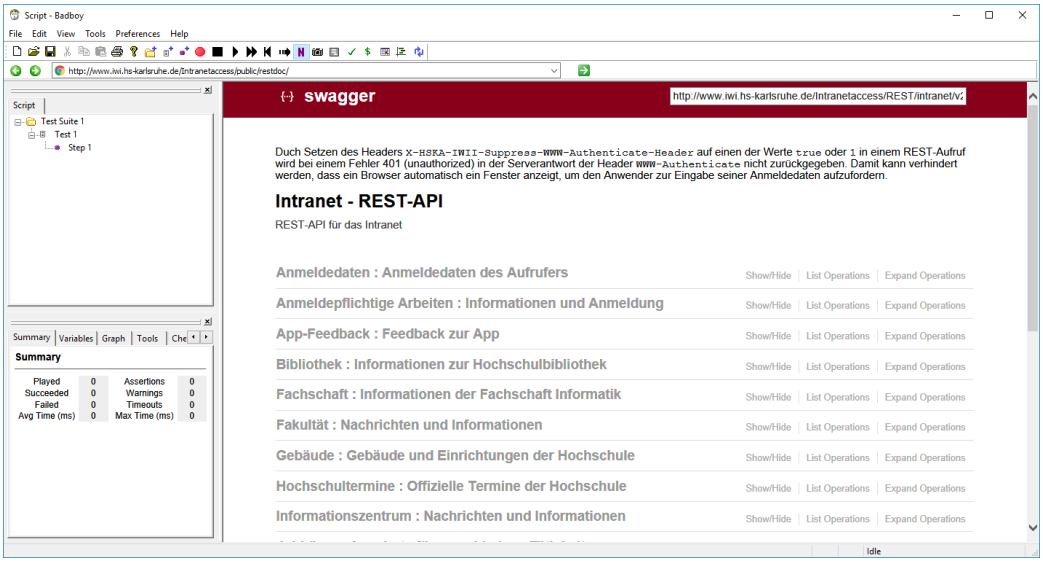
7 Wissenswertes und Besonderheiten

Man kann die Ergebnisse zur Laufzeit in Logfiles schreiben.

7.1 JMeter Plugin Manager

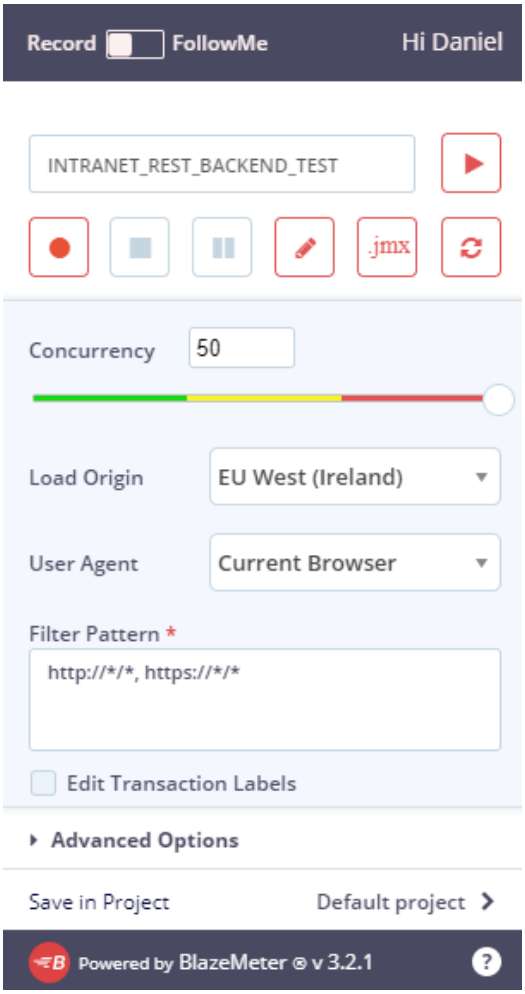
7.2 Aufzeichnen von Aktionen - UI web Test

Build in Solution: Bei non-test-Elements Testscript Recorder <-hiermit kann man ein script aufzeichnen um ganze Webseiten zu untersuchen



Für Windows only Platformabhängig BadBoy: Ansonsten drittanbieter verwenden wie etwa badboy.

Chrome Plugin: Blazemeter Mit Blazemeter ist man in der Lage direkt in Chrome und somit platformunabhängig Aufzeichnungen von Aktionen auf Webseiten aufzunehmen. Kleiner Nachteil: Man muss sich bei Blazemeter registrieren um den Export in .jmx Dateien ausführen zu können.



7.3 Datenbankabfragen - Datenbank testplan?

Auch datenbankabfragen lassen sich damit visualisieren

Falls interersse besteht, wie es funktioniert bitte in die Ausarbeitung schauen. Thread Gruppe erstellen -> Configuration (JDBC Connection Configuration) MySQL jdbc jar in lib folder von jmeter adden DAnn Sampler -> JDBC Request and put in values.. z.b. select * from table

7.4 Virtuelle Compute Unit

In die Cloud um an einer zentralen stelle viele rechner zu bedienen? Eventuell viele Rechner simulierenö

7.5 JMeter change Settings

In /bin folder von JMeter kann man die Datei user.properties öffnen und Werte anpassen.

8 Nachteile von JMeter

Wenn ein Variablenfeld required ist. z.b. bei JDBC Connection Configuration. dann wird einem der fehler nicht direkt im feld angezeigt.

Ziemlich resourcenhungrig. Trotz 16GB Arbeitsspeicher hängt sich das Ding gerne mal auf.

9 Alternativen

9.1 Gatling

https://www.heise.de/developer/artikel/Last-und-Performance-Tests-mit-JMeter-oder-Gatling-3648505.html

10 Fazit

Alles in allem ist JMeter ein Super tool

Literatur

[1] The Apache Software Foundation. Apache JMeter - User's Manual: History/Future. http://jmeter.apache.org/usermanual/history_future.html, 2018. (Zugriff am 26.04.2018).

[2] The Apache Software Foundation. Apache JMeter - What can I do with it? https://jmeter.apache.org, 2018. (Zugriff am 23.04.2018).

[3] Hauke Gierow. Amazon, Spotify, Twitter, Netflix: Mirai-Botnetz legte zahlreiche Webdienste lahm. https://www.golem.de/news/ddos-massiver-angriff-auf-dyndns-beeintraechtigt-github-und-amazon-1610-123966.html, 10 2016. (Zugriff am 23.04.2018).

[4] Emily H. Halili. *Apache JMeter*. Packt Publishing, Birmingham, 1 edition, 2008.

[5] Frank Pientka. Last- und Performance-Tests mit JMeter oder Gatling. https://www.heise.de/-3648505, 03 2017. (Zugriff am 23.04.2018).

[6] Waldemar Siebert. Vorteile der Testautomatisierung in der Praxis. https://www.testautomatisierung.org/welche-vorteile-bietet-die-testautomatisierung-in-der-praxis/, 07 2015. (Zugriff am 25.04.2018).

[7] SwissQ Consulting AG Waldemar Erdmann. Last- und Performancetest - Teil 3: Die Messkurven führen uns zu den Flaschenhälsen. https://swissq.it/de/testing/last-und-performancetest-teil-3-die-messkurven-fuehren-uns-zu-den-flaschenhaelsen/, 02 2014. (Zugriff am 23.04.2018).

[8] Wikipedia. Swing (Java). https://de.wikipedia.org/w/index.php?title=Swing_(Java)&oldid=171402620, 12 2017. (Zugriff am 25.04.2018).

[9] Wikipedia. Akzeptanztest (Softwaretechnik). https://de.wikipedia.org/w/index.php?title=Akzeptanztest_(Softwaretechnik)&oldid=176319126, 2018. (Zugriff am 23.04.2018).