

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

1	Einführung	4
1.1	Motivation	4
1.2	Manuelle vs automatisierte Tests	4
1.2.1	Manuelles Testen	4
1.2.2	Automatisiertes Testen	5
2	Apache JMeter	5
2.1	Was kann JMeter	6
2.1.1	Lastentests	6
2.1.2	Captcha Analyse	6
2.1.3	Auto vervollständigung von Formularen	6
2.1.4	JUnit Tests	7
2.2	Installation	7
2.2.1	Setting up the environment	7
2.2.2	Running JMeter	7
2.3	JMeter GUI	7
2.4	JMeter Command Line	7
2.4.1	Ausführen des Console Clients	7
3	Der Testplan	8
3.1	Elemente eines Testplans	8
3.1.1	Thread-Groups	8
3.1.2	Sampler	9
3.1.3	Listener	9
3.1.4	http request	9
3.1.5	Assertions	9
4	Lasttests von Webseiten	9
5	Funktionale Tests	9
6	Wissenswertes und Besonderheiten	9
6.1	JMeter Plugin Manager	9
6.2	Aufzeichnen von Aktionen - UI web Test	9

6.3	Datenbankabfragen - Datenbank testplan?	11
6.4	Virtuelle Compute Unit	12
6.5	JMeter change Settings	12
7	Nachteile von JMeter	12
8	Fazit	12
	Literatur	13

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 [2], 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 [6].

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 ressourcenaufwändig ist. Zusätzlich muss das Personal, sprich die Tester organisiert, betreut und gepflegt 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. [3, p. 11]

Trotz dieser Nachteile sollte man auf zusätzliche manuelle Tests¹ 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 funktioniert immer dann recht gut, wenn häufige Wiederholungen auftreten. Beispielsweise bei Regressionstests. 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 [5]. Dadurch werden Systeme bis zu ihren Grenzen und darüber hinaus getestet.

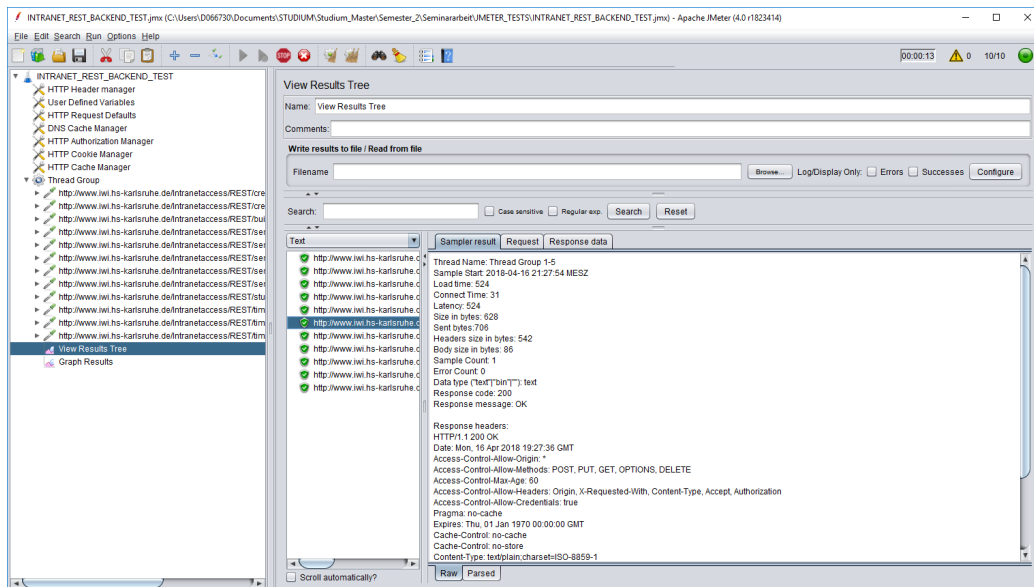
2 Apache JMeter

Auf den ersten Blick erscheint Apache JMeter 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 sich alle Arten von Tests erstellen und ausführen lassen. [4]

Tatsächlich stammt JMeter aus einer Zeit, als das Web und Application Server noch in den Kinderschuhen steckten. Ursprünglich wurde JMeter

¹ Das sogenannte User Acceptance Testing (UAT) ist gerade im finalen Entwicklungsstadium einer Anwendung unabdingbar. [7]

entwickelt um den Application Server Tomcat zu testen. Seitdem wurde das Programm fortlaufend weiter entwickelt, so dass man heute in der Lage ist von verteilte Tests in der Cloud, über das Testen von dynamischen Webseiten bis hin zu JUnit Tests, alle möglichen Arten von Tests zu realisieren. [1]



JMeter Plattformunabhängig wegen java anwendung. kann sowohl auf mac als auch auf windows laufen. lediglich die java runtime

Wann wurde das Tool das erste mal entwickelt? wie oft wird es eingesetzt. auch von großen unternehmen? kommt bei sap auch zum einsatz

2.1 Was kann JMeter

Es lassen sich ebenfalls mehrere Benutzer erzeugen und eine Anwendung mit diesen simultan testen.

2.1.1 Lastentests

2.1.2 Captcha Analyse

Captcha analysieren und erkenne?

2.1.3 Auto vervollständigung von Formularen

befüllen von formularen

2.1.4 JUnit Tests

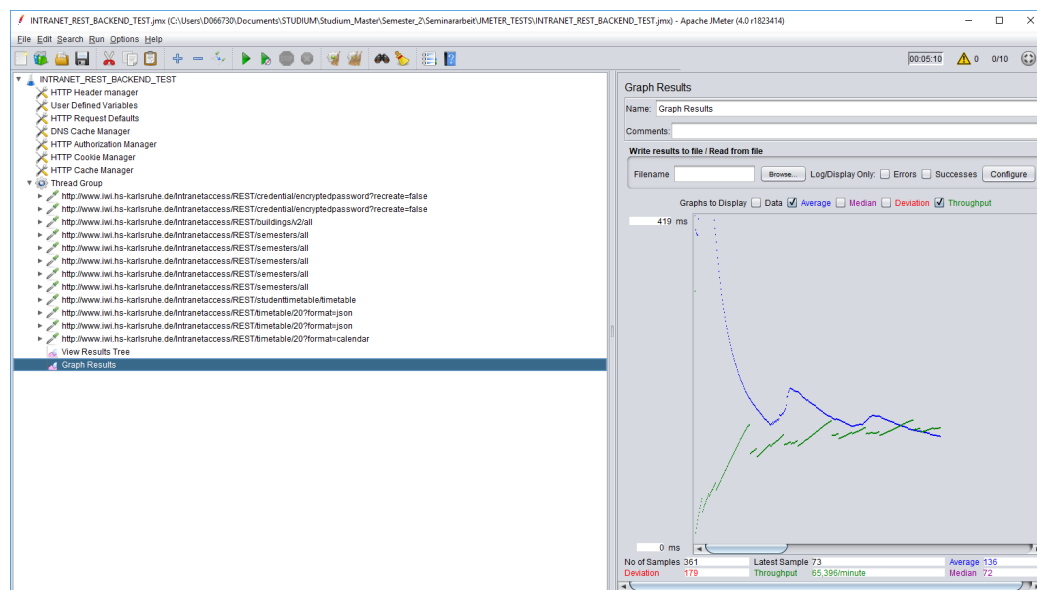
junit tests in jmeter importieren

2.2 Installation

2.2.1 Setting up the environment

2.2.2 Running JMeter

2.3 JMeter GUI



2.4 JMeter Command Line

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

2.4.1 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. Danach 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

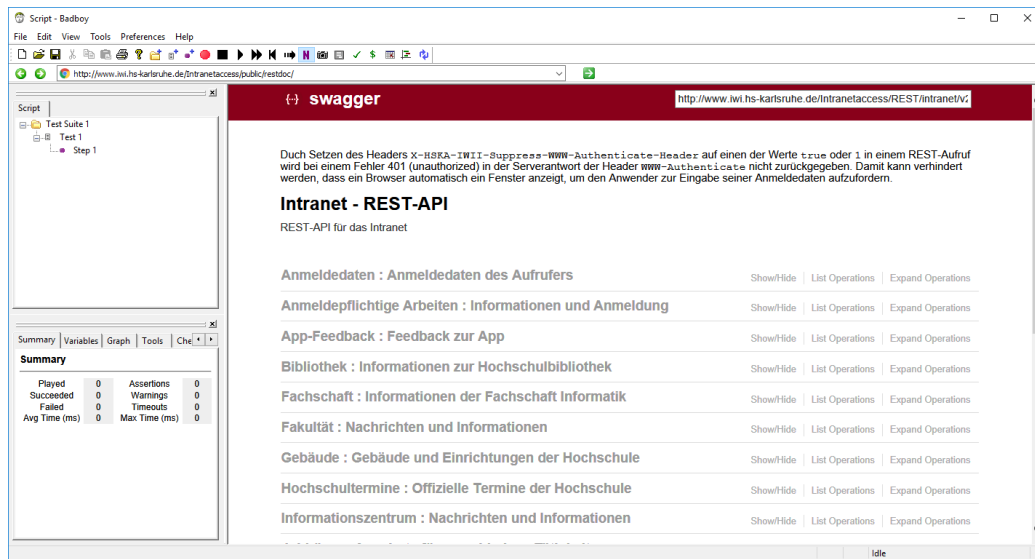
6 Wissenswertes und Besonderheiten

Man kann die Ergebnisse zur Laufzeit in Logfiles schreiben.

6.1 JMeter Plugin Manager

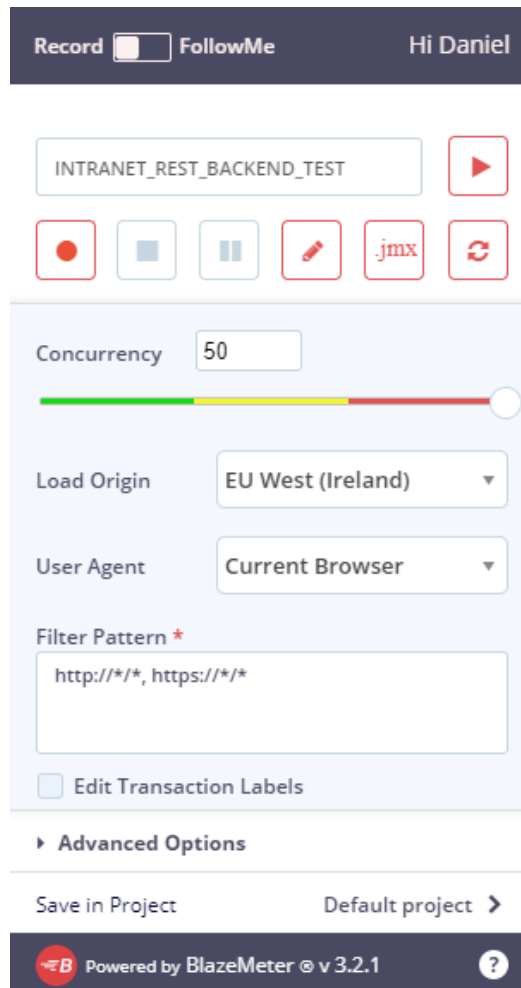
6.2 Aufzeichnen von Aktionen - UI web Test

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



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

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



6.3 Datenbankabfragen - Datenbank testplan?

Auch datenbankabfragen lassen sich damit visualisieren

Falls interresse 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

6.4 Virtuelle Compute Unit

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

6.5 JMeter change Settings

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

7 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 ressourcenhungrig. Trotz 16GB Arbeitsspeicher hängt sich das Ding gerne mal auf.

8 Fazit

Alles in allem ist JMeter ein Super tool

Literatur

- [1] The Apache Software Foundation. Apache JMeter - What can I do with it? <https://jmeter.apache.org>, 2018. (Zugriff am 23.04.2018).
- [2] 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).
- [3] Emily H. Halili. *Apache JMeter*. Packt Publishing, Birmingham, 1 edition, 2008.
- [4] Frank Pientka. Last- und Performance-Tests mit JMeter oder Gatling. <https://www.heise.de/-3648505>, 03 2017. (Zugriff am 23.04.2018).
- [5] 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).
- [6] 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).
- [7] Wikipedia. Akzeptanztest (Softwaretechnik). [https://de.wikipedia.org/w/index.php?title=Akzeptanztest_\(Softwaretechnik\)&oldid=176319126](https://de.wikipedia.org/w/index.php?title=Akzeptanztest_(Softwaretechnik)&oldid=176319126), 2018. (Zugriff am 23.04.2018).