

STRESSTESTS WITH GATLING

NO MORE **STRESS** WITH YOUR TESTS

Niko Köbler

Software-Architect, Developer & Trainer

niko@n-k.de | www.n-k.de | [@dasniko](https://twitter.com/dasniko)

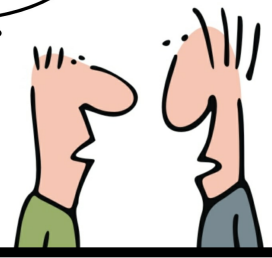


@dasniko

geek & poke

NO, BUT I'VE
WRITTEN IT UNDER
STRESS

HAVE YOU TESTED
YOUR CODE UNDER
STRESS?



```
99 + //TODO This test is flaky, please fix
100 + @Ignore
101 @Test
```

JMETER



<http://jmeter.apache.org>

TON'S OF

XML

CONFIGURATION

View Results Tree

Name: View Results Tree

Comments:

Write results to file / Read from file

Filename

Browse...

Log/Display Only: Errors Successes

Configure

- HTTP Request
 - http://jmeter.apache.org
 - http://jmeter.apache.org
 - http://www.apache.org/
 - http://jmeter.apache.org
 - http://jmeter.apache.org
- HTTP Request
 - http://jmeter.apache.org
 - http://jmeter.apache.org
 - http://www.apache.org/
 - http://jmeter.apache.org
 - http://jmeter.apache.org
- Duration Assertion

Sampler result Request Response data



The **Apache**
Software Foundation
<http://www.apache.org/>



About

- [Overview](#)
- [Changes](#)
- [Subscribe to What's New](#)
- [Issues](#)
- [License](#)
- [Contributors](#)

Download

- [Download Releases](#)
- [Developer \(Nightly\) Builds](#)

Documentation

Apache JMeter™

The **Apache JMeter™** desktop application is open source software, a 100% pure [Java](#) application designed to load test functional behavior and measure performance. It was originally designed for testing Web Applications but has since expanded to other test functions.

What can I do with it?

HTML (download resources)

Scroll automatically?

Search:

Find next

Case sensitive

Regular exp.

Load Test Results - Internet Explorer

C:\jmeter\extras\Test.html

Load Test Results

Date report: 2014/04/10 16:13 Designed for use with [JMeter](#) and [Ant](#).

Summary

# Samples	Failures	Success Rate	Average Time	Min Time	Max Time
60	4	93.33%	220 ms	102 ms	351 ms

Pages

URL	# Samples	Failures	Success Rate	Average Time	Min Time	Max Time	
Sample 1	30	4	86.67%	228 ms	102 ms	338 ms	+
Sample 2	30	0	100.00%	213 ms	109 ms	351 ms	+

Failure Detail

Sample 1

Response	Failure Message
200 - OK	Test failed: text expected not to contain /3/
200 - OK	Test failed: text expected not to contain /3/
200 - OK	Test failed: text expected not to contain /3/
200 - OK	Test failed: text expected not to contain /3/

THE NEW KID IN TOWN



<http://gatling.io>

based on

SCALA, AKKA & NETTY

NON-BLOCKING

ASYNCHRONOUS

Wait... what?

SCALA?

DSL!

GETTING STARTED

Download zip bundle

Use Maven

```
<dependency>  
  <groupId>io.gatling.highcharts</groupId>  
  <artifactId>gatling-charts-highcharts</artifactId>  
  <version>2.2.0</version>  
</dependency>
```


Use SBT

```
libraryDependencies += "io.gatling.highcharts" %% "gatling-charts-h
```

...or use [3rd party extensions](#)

GATLING RECORDER

Gatling Recorder - Configuration



Recorder mode: HTTP Proxy

Network

Listening port*: localhost HTTP/HTTPS 8000 HTTPS mode: Self-signed Certificate

Outgoing proxy: host: HTTP Username Password

Simulation Information

Package: io.dasniko.gatling Class Name*: RecordedSimulation

Follow Redirects? Infer html resources? Automatic Referers?
 Remove cache headers? Save & check response bodies?

Output

Output folder*: /Users/Niko/gatling/user-files/simulations Browse

Encoding: Unicode (UTF-8)

Filters

Java regular expressions that matches the entire URI Strategy Disabled

Whitelist	Blacklist

+ - Clear + - Clear No static resources

Save preferences Start !

GATLING RECORDER

works as

HTTP PROXY

or

HAR CONVERTER

...and has an API

EXECUTING



BROWSER SIMULATION

```
val httpConf = http
    .baseUrl("http://localhost:8080")
    .acceptHeader("text/html,application/xhtml+xml,application/xml;q=0.9,application/atom+xml,application/javascript;q=0.8,*/*;q=0.7")
    .doNotTrackHeader("1")
    .acceptLanguageHeader("en-US,en;q=0.5")
    .acceptEncodingHeader("gzip, deflate")
    .userAgentHeader("Mozilla/5.0 (Windows NT 5.1; rv:31.0) Gecko/20100101 Firefox/31.0")
```

FEEDER

```
val csvFeeder = csv("values.csv").circular
```

- csv, tsv, ssv
- jsonFile, jsonUrl
- jdbcFeeder
- redisFeeder
- queue, circular, random

VALUES.CSV

```
key,value  
john,23  
james,42  
jacob,1337
```

TEST SCENARIO

```
val scn = scenario("BasicSimulation")  
  .feed(csvFeeder)  
  .exec(http("request_1")  
    .get("/").queryParams("${key}", "${value}"))
```

HOW MANY USERS?

- `atOnceUsers(1)`
- `rampUsers(500)`
- `constantUsersPerSec(1000) during (10)`

<http://gatling.io/#/cheat-sheet/2.2.0>

SETUP AND ASSERTIONS

```
setUp(  
    scn.inject(atOnceUsers(1))  
)  
    .protocols(httpConf)  
    .assertions(  
        global.responseTime.max.lessThan(50),  
        global.successfulRequests.percent.greaterThan(95)  
    )  
)
```

PUTTING IT ALL TOGETHER

```
class LoadScenario extends Simulation {  
  
  val csvFeeder = csv("values.csv").circular  
  
  val httpConf = http  
    .baseUrl("http://localhost:8080")  
    .acceptHeader("text/html,application/xhtml+xml,application/xml")  
    .doNotTrackHeader("1")  
    .acceptLanguageHeader("en-US,en;q=0.5")  
    .acceptEncodingHeader("gzip, deflate")  
    .userAgentHeader("Mozilla/5.0 (Windows NT 5.1; rv:31.0) Gecko/  
    .disableWarmUp  
  
  val scn = scenario("BasicRedisSimulation")  
    .feed(csvFeeder)  
    .exec(http("request_1"))  
}
```

REPORTS

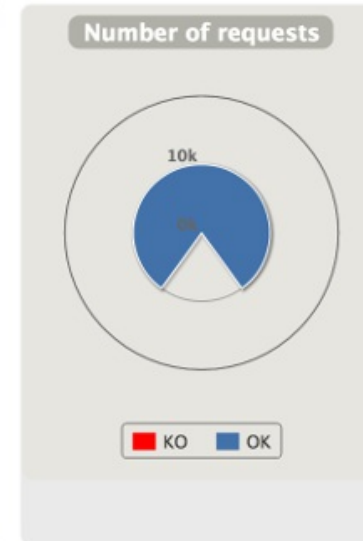
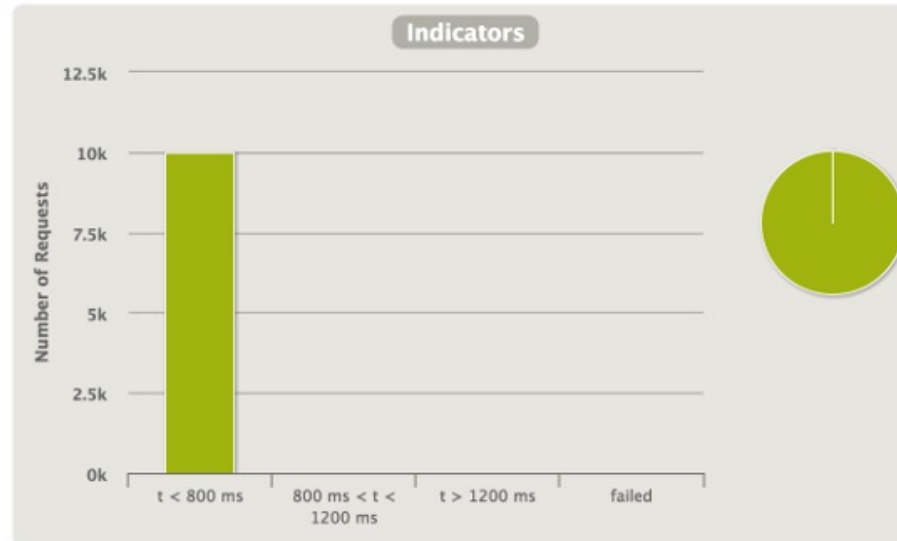
▶ GLOBAL

▼ DETAILS

2015-09-19 18:15:41 +02:00, duration : 10 seconds

Active Users
Requests / sec
Responses / sec

> Global Information



ASSERTIONS

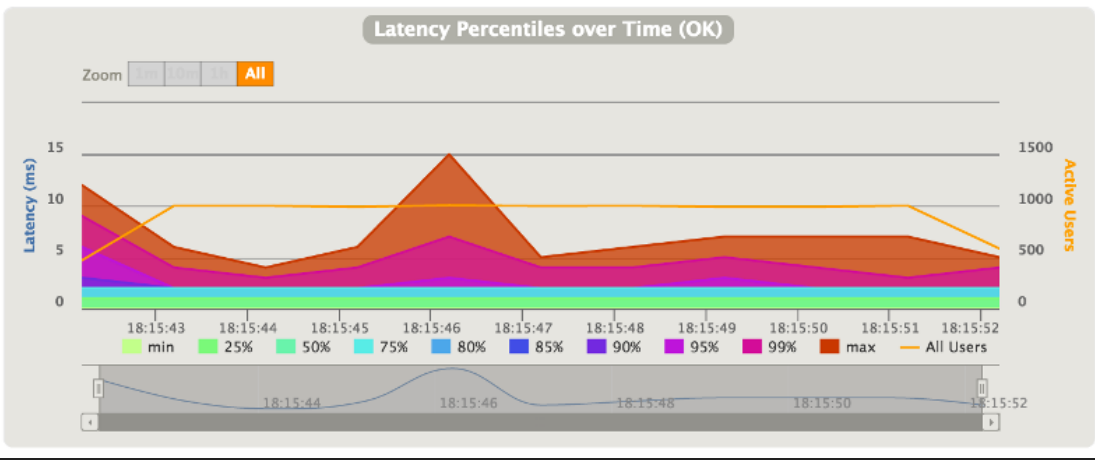
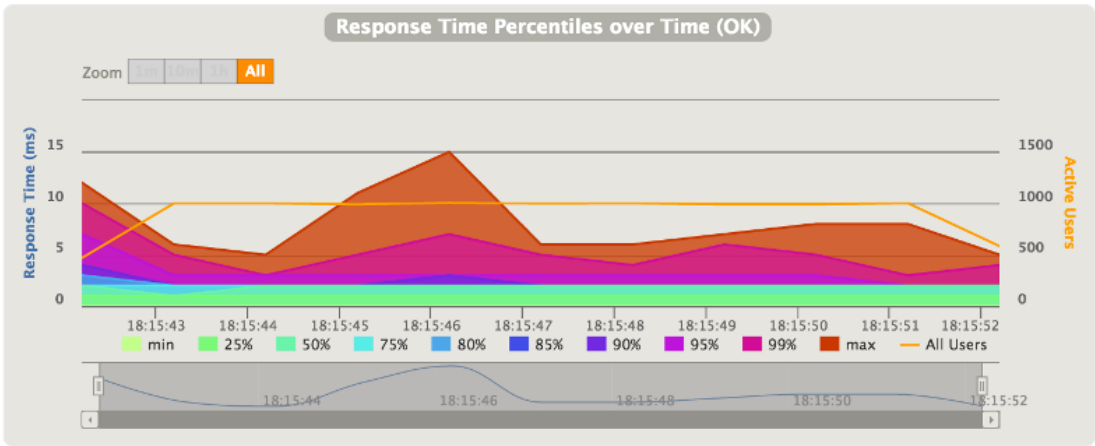
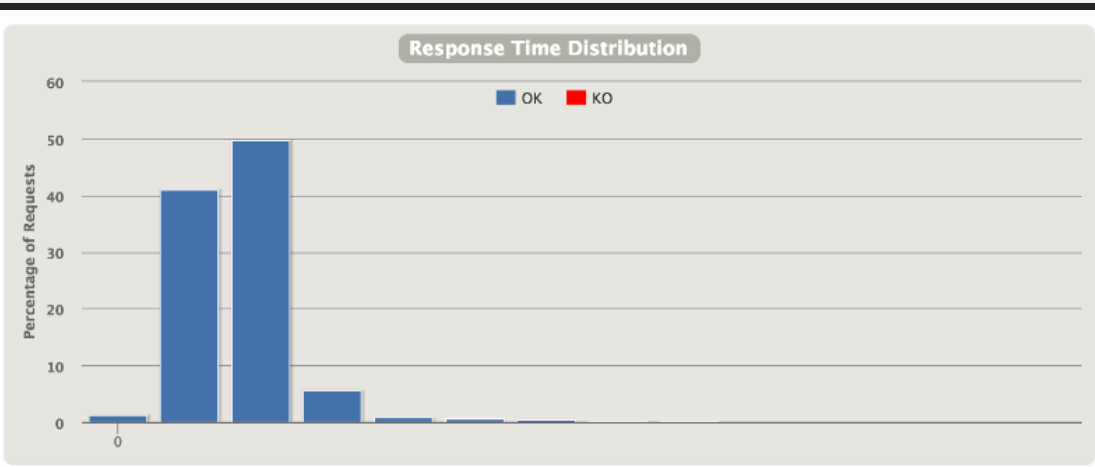
Assertion	Status
Global: max of response time is less than 50	OK
Global: percentage of successful requests is greater than 95	OK

STATISTICS

Expand all groups | Collapse all groups

Requests	Executions				Response Time (ms)								
	Total	OK	KO	% KO	Req/s	Min	50th pct	75th pct	95th pct	99th pct	Max	Mean	Std Dev
Global Information	10000	10000	0	0%	992.063	0	2	2	3	6	15	1	0
redis_request	10000	10000	0	0%	992.063	0	2	2	3	6	15	1	0





REALTIME MONITORING

GRAPHITE PROTOCOL

INFLUX DB

Time-series data store.

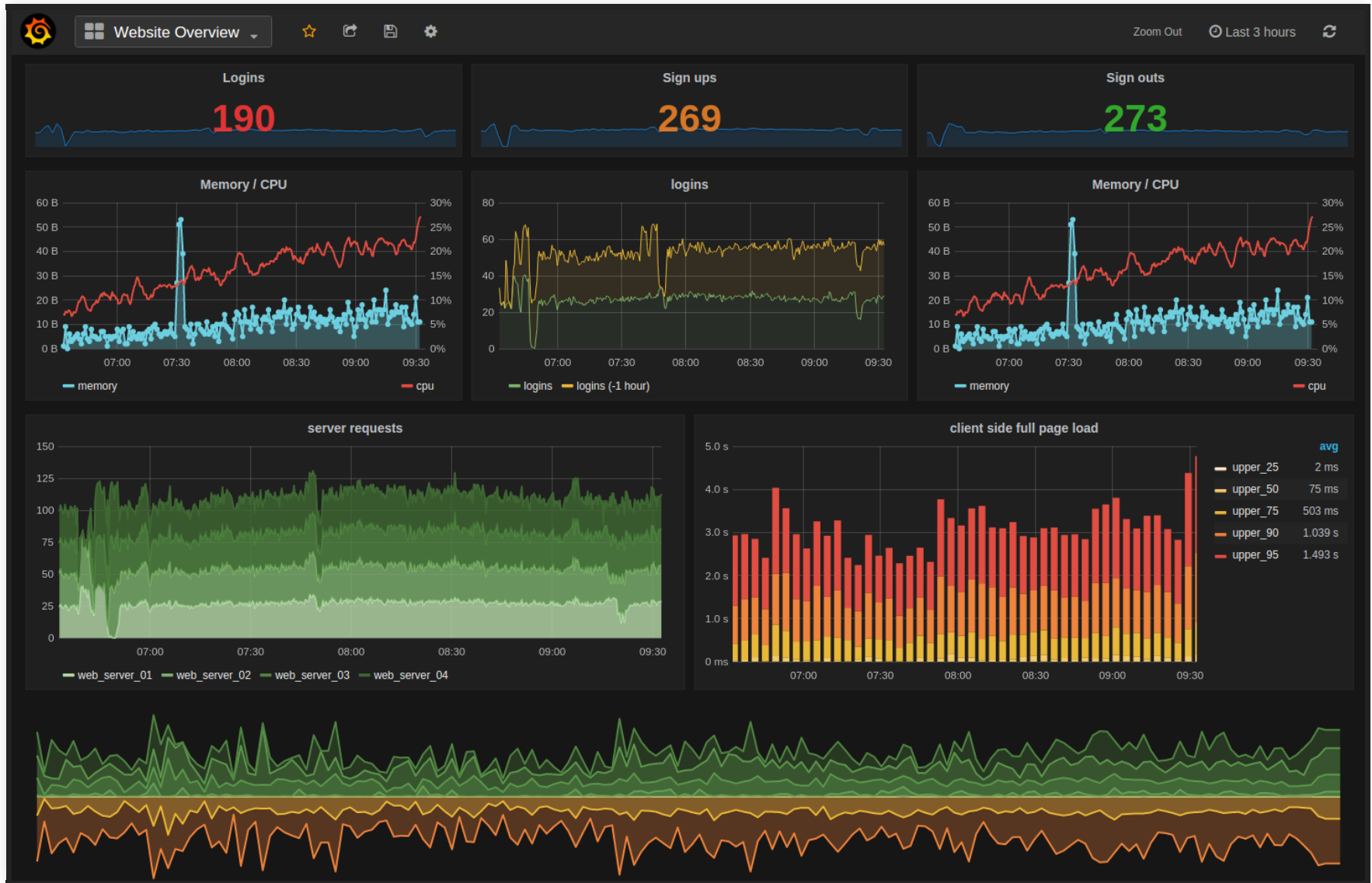
<https://influxdata.com/>

GRAFANA

Graph and Dashboard builder for visualizing time series metrics.

<http://grafana.org/>

GRAFANA DASHBOARD



DISTRIBUTED TESTING

BUILD-SLAVES

FAT-JARS

INFLUX (& GRAFANA)

TRENDING



Gatling
LOAD TESTING

200,000
downloads

THANKS

2013

2014

2015

GATLING TOOL

- Open Source:
<https://github.com/gatling/gatling>
- Professional Services:
Support, Consulting & Training
gatling.io

THANK YOU!

ANY QUESTIONS?

Niko Köbler

Software-Architect, Developer & Trainer

niko@n-k.de | www.n-k.de | [@dasniko](https://twitter.com/dasniko)

<https://github.com/dasniko/gatling-playground>