
php developer
day 2015
#medd15

Ansätze zum **Profiling** bei PHP-Anwendungen

It´s all about Speed

Übersicht

What´s up in here?

http://m5.paperblog.com/i/55/552041/dogs-travel-through-space-at-warp-speed-L-Twepne.jpeg

http://www.motor-freizeit-trends.at/wp-content/uploads/110202-Kalorie-Calmund-0577-1-Pick.jpg?07ef5a

http://devmeme.winben.hu/wp-content/uploads/2015/06/8HUukjF.gif

https://upload.wikimedia.org/wikipedia/commons/1/19/USS_Pennsylvania_(SSBN_735).jpg

http://www.bildarchiv-hamburg.de/folders/Bilder%20von%20Schiffen%20im%20Hafen%20Hamburg%20und%20auf%20der%20Elbe/100340_7113_Frachtschiff-APL-FLORIDA-Elbe-bei-Stadte.jpg

http://www.floss-mv.de/images/Slideshow760/Floss760_3_gross.jpg

http://pix.echtlustig.com/d/1403/mit-dem-fahrrad-ins-wasser-springen.jpg

http://img.nauticexpo.de/images_ne/photo-g/innenbordmotor-runabout-holz-klassisch-26924-388669.jpg

http://images5.fanpop.com/image/answers/381000/381532_1335957017331_240_274.jpg

Wer bin ich?

Who am I?

Wer bin ich?

Steve Schütze – Senior CMS-Developer

PHP-Developer seit 10 Jahren

Bei move:elevator seit diesem Jahr

Haupteinsatzgebiet TYPO3 und Shopsysteme

Kontakt:
sts@move-elevator.de

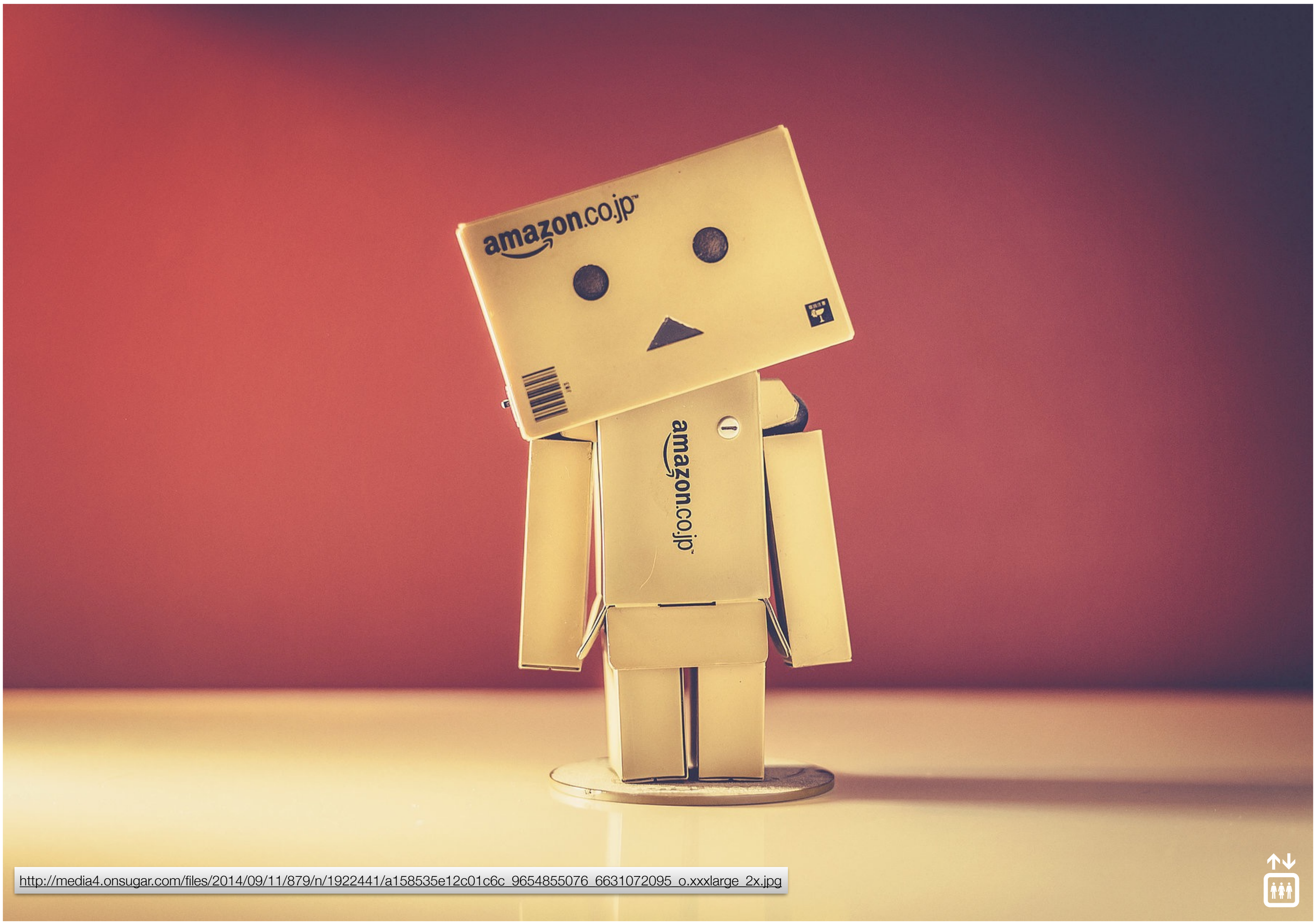
Warum?

Now I should also deal with speed?

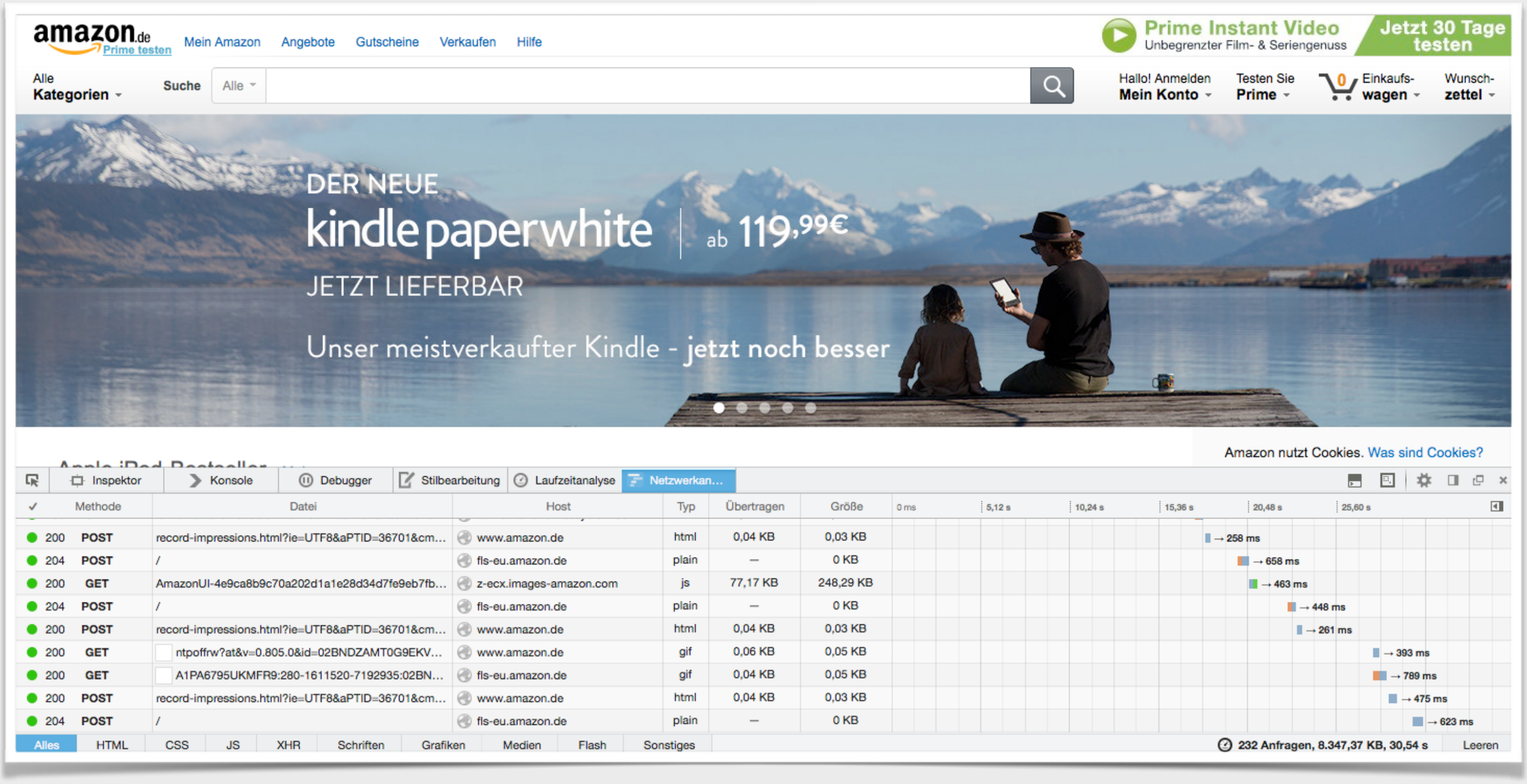
http://assets.nydailynews.com/polopoly_fs/1.948618!/img/httpImage/image.jpg_gen/derivatives/article_970/alg-money-pile-jpg.jpg

http://www.adweek.com/files/imagecache/node-blog/blogs/yahoo-original.jpg

https://lh5.ggpht.com/8PODwBXKk4L201m4lO1wifRDfbn4Q1JxNxOzj-5TXPJ85_S-vOqntLi7TsVyeFQM0w4%3Dw300



http://media4.onsugar.com/files/2014/09/11/879/n/1922441/a158535e12c01c6c_9654855076_6631072095_o.xxxlarge_2x.jpg

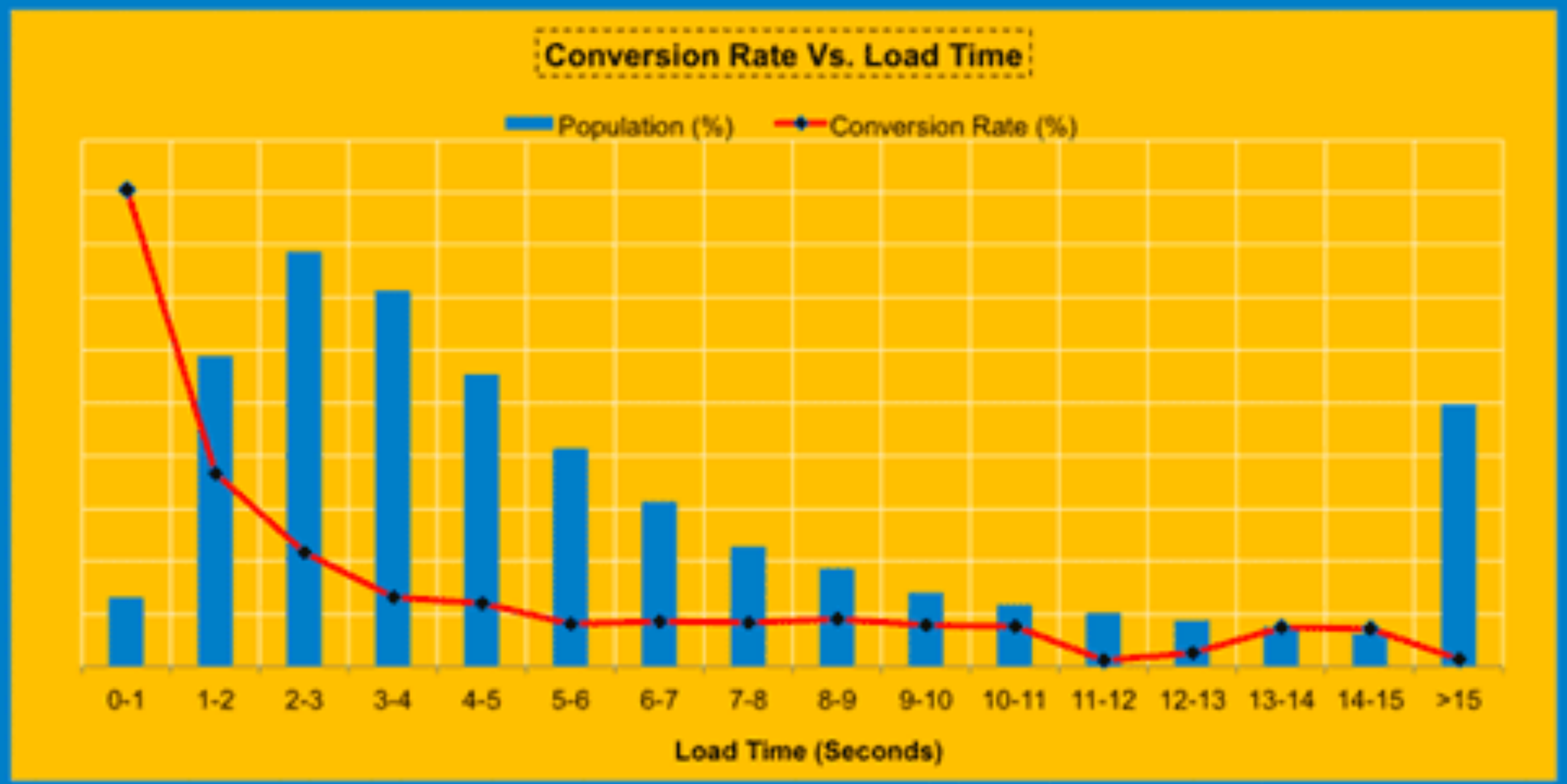

amazon.de
Prime testen
Mein Amazon
Angebote
Gutscheine
Verkaufen
Hilfe
Prime Instant Video
Unbegrenzter Film- & Seriengenuss
Jetzt 30 Tage testen
Alle Kategorien
Suche
Alle
Hallo! Anmelden Mein Konto
Testen Sie Prime
0 Einkaufs-wagen
Wunsch-zettel
DER NEUE
kindle paperwhite
ab 119,99€
JETZT LIEFERBAR
Unser meistverkaufter Kindle - jetzt noch besser
Amazon nutzt Cookies. Was sind Cookies?
Inspektor
Konsole
Debugger
Stilbearbeitung
Laufzeitanalyse
Netzwerkan…
Methode
Datei
Host
Typ
Übertragen
Größe
0 ms
5,12 s
10,24 s
15,36 s
20,48 s
25,60 s
200 POST record-impressions.html?ie=UTF8&aPTID=36701&cm… www.amazon.de html 0,04 KB 0,03 KB → 258 ms
204 POST / fls-eu.amazon.de plain — 0 KB → 658 ms
200 GET AmazonUI-4e9ca8b9c70a202d1a1e28d34d7fe9eb7fb… z-ecx.images-amazon.com js 77,17 KB 248,29 KB → 463 ms
204 POST / fls-eu.amazon.de plain — 0 KB → 448 ms
200 POST record-impressions.html?ie=UTF8&aPTID=36701&cm… www.amazon.de html 0,04 KB 0,03 KB → 261 ms
200 GET ntpoffrw?at&v=0.805.0&id=02BNDZAMT0G9EKV… www.amazon.de gif 0,06 KB 0,05 KB → 393 ms
200 GET A1PA6795UKMFR9:280-1611520-7192935:02BN… fls-eu.amazon.de gif 0,04 KB 0,05 KB → 789 ms
200 POST record-impressions.html?ie=UTF8&aPTID=36701&cm… www.amazon.de html 0,04 KB 0,03 KB → 475 ms
204 POST / fls-eu.amazon.de plain — 0 KB → 623 ms
Alles
HTML
CSS
JS
XHR
Schriften
Grafiken
Medien
Flash
Sonstiges
232 Anfragen, 8.347,37 KB, 30,54 s
Leeren

Impact of site performance on overall site conversion rate….

Baseline – 1 in 2 site visits had response time > 4 seconds

* Sharp decline in conversion rate as average site load time increases from 1 to 4 seconds

* Overall average site load time is lower for the converted population (3.22 Seconds) than the non-converted population (6.03 Seconds)



Note: Load Time here is the time taken from head of the page to page ready (T_Page)

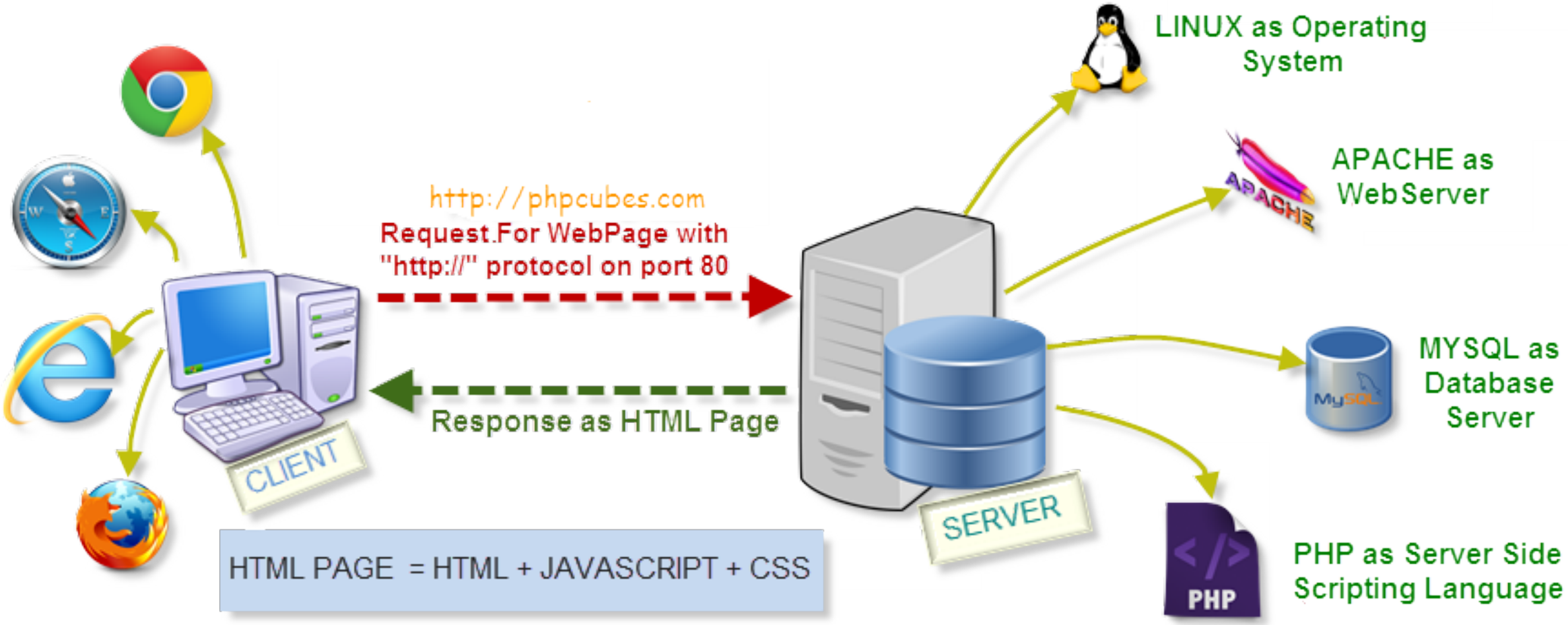
http://www.webperformancetoday.com/wp-content/uploads/2012/02/walmart-2.png

Welche Ansatzpunkte?

And where shall I start?

Devices
Browser
Load-Balancer
Caching
Images
Javascript
CSS
Server
Downloadrate
PHP
Datenbank
CDN

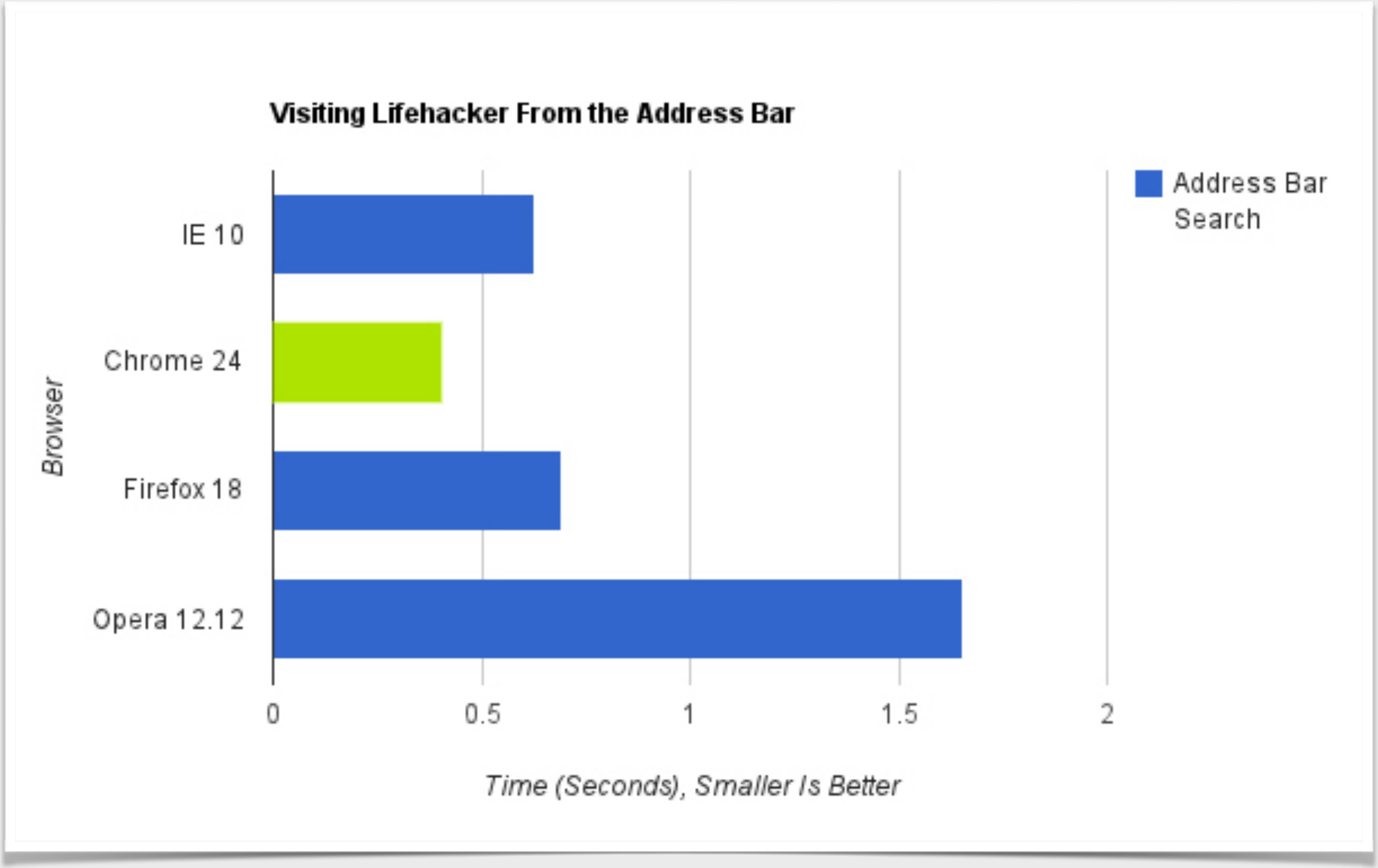
http://www.wordle.net/



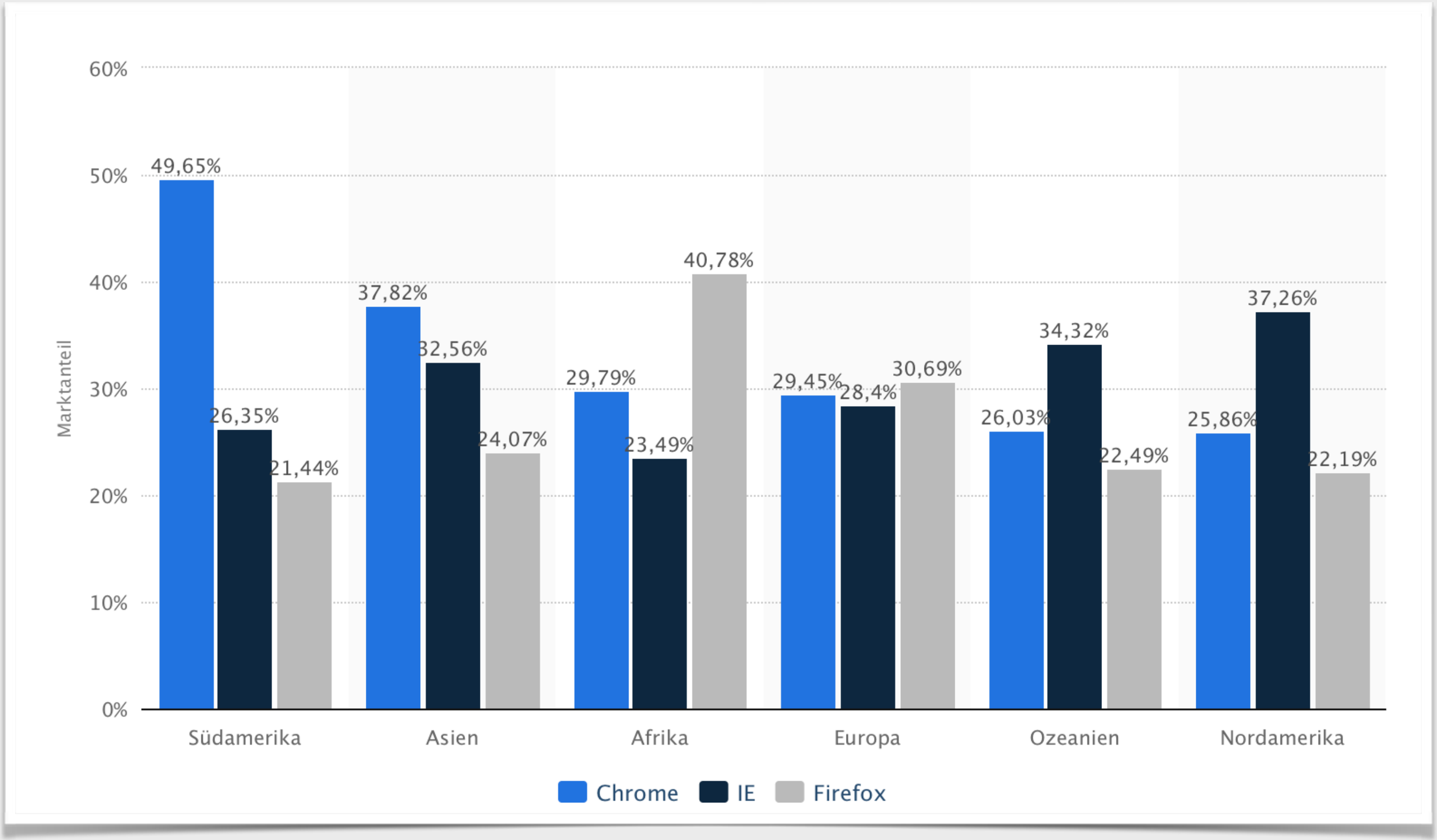
http://www.phpcubes.com/upload/images/client-server-request.png

Der Client

Browser



https://i.kinja-img.com/gawker-media/image/upload/18bl5q9ayousejpg.jpg



http://de.statista.com/graphic/1/214514/marktanteile-der-internet-browser-nach-kontinenten.jpg

Devices

Downloadrate

Caching

Die Ressourcen

CDN

Images

CSS, Javascript & HTML

Der Server

Caching

Redirects

Datenbank

Bigpipe

Profiling der PHP-Anwendung

We are ready to start.

Profiling VM

https://github.com/move-elevator/profiler-vm

Der Klassiker

Xdebug

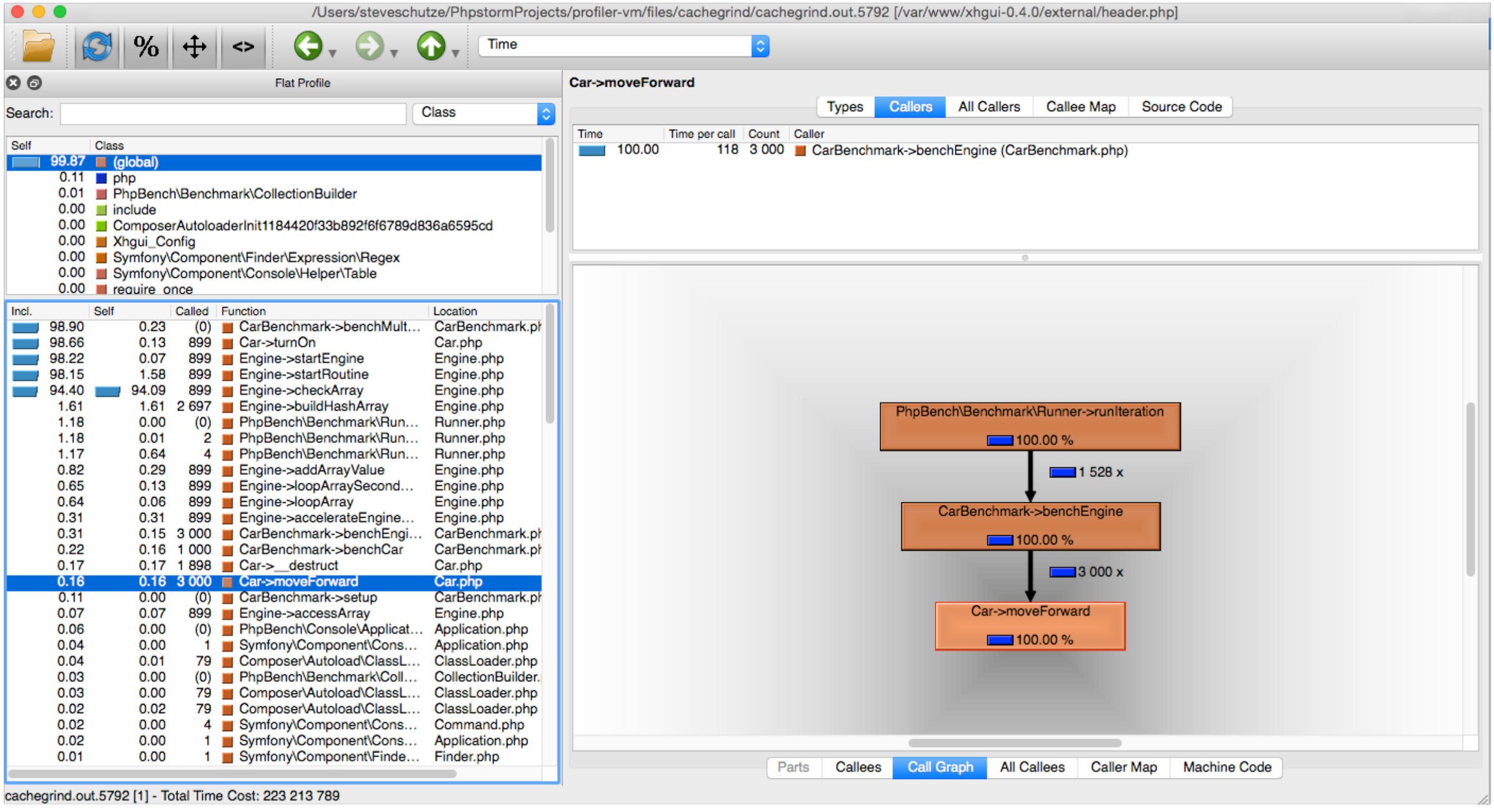
→ Xdebug

Das Schweizer Taschenmesser unter den Profilern

erster Release 2002

in C geschrieben

für den Einsatz auf Entwicklungssystemen geeignet


/Users/steveschutze/PhpstormProjects/profiler-vm/files/cachegrind/cachegrind.out.5792 [/var/www/xhgui-0.4.0/external/header.php]
Time
Flat Profile
Search:
Class
Self
Class
99.87 (global)
0.11 php
0.01 PhpBench\Benchmark\CollectionBuilder
0.00 include
0.00 ComposerAutoloaderInit1184420f33b892f6f6789d836a6595cd
0.00 Xhgui_Config
0.00 Symfony\Component\Finder\Expression\Regex
0.00 Symfony\Component\Console\Helper\Table
0.00 require_once
Incl. Self Called Function Location
98.90 0.23 (0) CarBenchmark->benchMult... CarBenchmark.ph
98.66 0.13 899 Car->turnOn Car.php
98.22 0.07 899 Engine->startEngine Engine.php
98.15 1.58 899 Engine->startRoutine Engine.php
94.40 94.09 899 Engine->checkArray Engine.php
1.61 1.61 2 697 Engine->buildHashArray Engine.php
1.18 0.00 (0) PhpBench\Benchmark\Run... Runner.php
1.18 0.01 2 PhpBench\Benchmark\Run... Runner.php
1.17 0.64 4 PhpBench\Benchmark\Run... Runner.php
0.82 0.29 899 Engine->addArrayValue Engine.php
0.65 0.13 899 Engine->loopArraySecond... Engine.php
0.64 0.06 899 Engine->loopArray Engine.php
0.31 0.31 899 Engine->accelerateEngine... Engine.php
0.31 0.15 3 000 CarBenchmark->benchEngi... CarBenchmark.ph
0.22 0.16 1 000 CarBenchmark->benchCar CarBenchmark.ph
0.17 0.17 1 898 Car->__destruct Car.php
0.16 0.16 3 000 Car->moveForward Car.php
0.11 0.00 (0) CarBenchmark->setup CarBenchmark.ph
0.07 0.07 899 Engine->accessArray Engine.php
0.06 0.00 (0) PhpBench\Console\Applicat... Application.php
0.04 0.00 1 Symfony\Component\Cons... Application.php
0.04 0.01 79 Composer\Autoload\ClassL... ClassLoader.php
0.03 0.00 (0) PhpBench\Benchmark\Coll... CollectionBuilder.
0.03 0.00 79 Composer\Autoload\ClassL... ClassLoader.php
0.02 0.02 79 Composer\Autoload\ClassL... ClassLoader.php
0.02 0.00 4 Symfony\Component\Cons... Command.php
0.02 0.00 1 Symfony\Component\Cons... Application.php
0.01 0.00 1 Symfony\Component\Finde... Finder.php
cachegrind.out.5792 [1] - Total Time Cost: 223 213 789
Car->moveForward
Types
Callers
All Callers
Callee Map
Source Code
Time
Time per call
Count
Caller
100.00 118 3 000 CarBenchmark->benchEngine (CarBenchmark.php)
PhpBench\Benchmark\Runner->runIteration
100.00 %
1 528 x
CarBenchmark->benchEngine
100.00 %
3 000 x
Car->moveForward
100.00 %
Parts
Callees
Call Graph
All Callees
Caller Map
Machine Code

Xdebug

Trace, Code-Coverage,
Debugging and Profiling

Der Krösus

XHProf

→ XHProf

Dient als Grundlage für viele andere Tools

OpenSource seit 2009

von Facebook entwickelt

in C geschrieben und über PHP Library ergänzt

für den Einsatz auf Produktivsystemen geeignet

Es wurde schon mehrfach versucht diese PHP-Library zu überarbeiten

XHG | Recent | Longest wall time | Most CPU | Most memory | Custom View | Watch Functions | Waterfall

Profile data for GET /

THIS RUN

URL /
Time Jun 30th 13:52:18
ID 559282f2c921921c15776028
Wall Time 13,290 µs
CPU Time 11,998 µs
Memory Usage 65,888 bytes
Peak Memory Usage 1,709,520 bytes

GET

No GET data

SERVER

PATH
/sbin:/usr/sbin:/bin:/usr/bin
SCRIPT_NAME
/index.php
REQUEST_URI
/
QUERY_STRING
REQUEST_METHOD
GET
SERVER_PROTOCOL
HTTP/1.1
GATEWAY_INTERFACE
CGI/1.1
REMOTE_PORT
63248
SCRIPT_FILENAME
/var/www/profiler/index.php
SERVER_ADMIN
[no address given]
DOCUMENT_ROOT
/var/www/profiler
REMOTE_ADDR
192.168.33.1

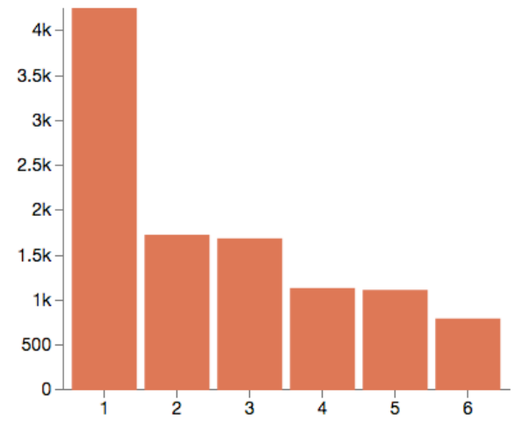
Watch Functions

View Callgraph | Compare this run

Function	Call Count	ewt	emu	epmu

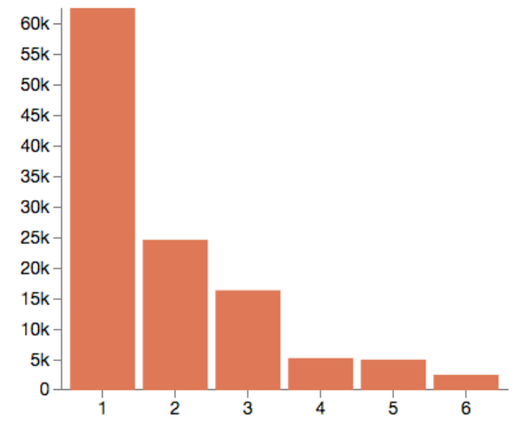
You have no watch functions setup. Add a watch function now.

Self Wall Time



Engine::buildHashArray
4,254 µs
Engine::startRoutine
1,730 µs
Engine::checkArray
1,690 µs

Memory Hogs



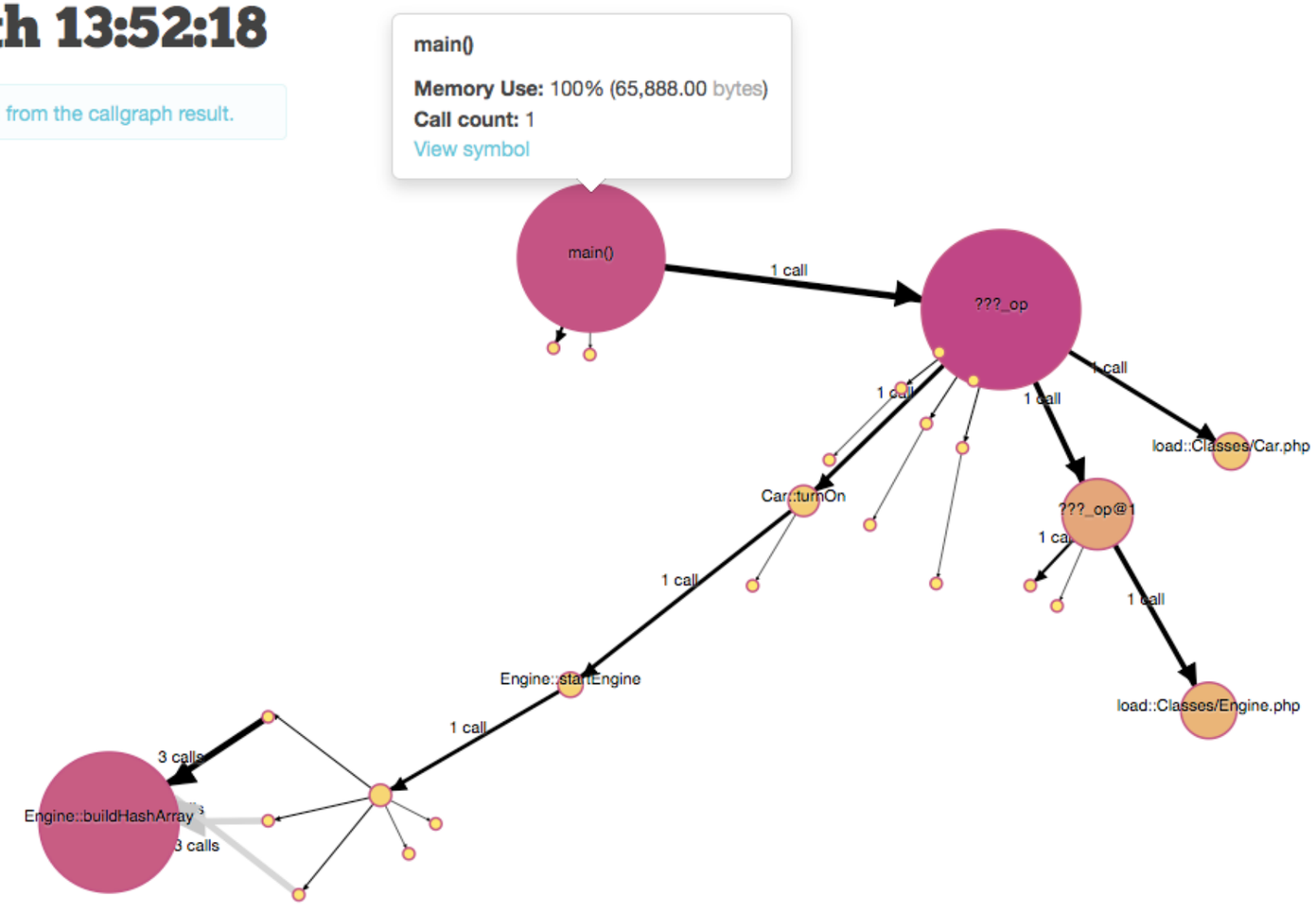
Engine::buildHashArray
62,568 bytes
load::Classes/Engine.php
24,632 bytes
load::Classes/Car.php
16,360 bytes

Filter by function name

Function	Call Count	Self Wall Time	Self CPU	Self Memory Usage	Self Peak Memory Usage	Inclusive Wall Time	Inclusive CPU	Inclusive Memory Usage	Inclusive Peak Memory Usage
Engine::buildHashArray	3	4,254 µs	6,998 µs	62,568 bytes	20,536 bytes	4,254 µs	6,998 µs	62,568 bytes	20,536 bytes
Engine::startRoutine	1	1,730 µs	0 µs	2,368 bytes	0 bytes	9,820 µs	9,998 µs	9,616 bytes	1,666,680 bytes
Engine::checkArray	1	1,690 µs	0 µs	784 bytes	0 bytes	1,690 µs	0 µs	784 bytes	0 bytes
???_op	1	1,137 µs	1,000 µs	2,520 bytes	0 bytes	12,380 µs	11,998 µs	70,600 bytes	1,709,520 bytes
Engine::accessArray	1	1,117 µs	3,000 µs	784 bytes	1,634,920 bytes	1,117 µs	3,000 µs	784 bytes	1,634,920 bytes
{closure}	1	798 µs	0 µs	808 bytes	0 bytes	798 µs	0 µs	808 bytes	0 bytes
Engine::accelerateEngineRotation	1	452 µs	1,000 µs	832 bytes	0 bytes	452 µs	1,000 µs	832 bytes	0 bytes
Engine::addArrayValue	1	445 µs	0 µs	-18,976 bytes	0 bytes	1,465 µs	0 µs	1,856 bytes	0 bytes
Engine::loopArraySecondWay	1	439 µs	0 µs	-18,968 bytes	0 bytes	1,754 µs	2,999 µs	1,872 bytes	0 bytes
load::Classes/Car.php	1	346 µs	0 µs	16,360 bytes	11,464 bytes	346 µs	0 µs	16,360 bytes	11,464 bytes
load::Classes/Engine.php	1	277 µs	0 µs	24,632 bytes	25,328 bytes	277 µs	0 µs	24,632 bytes	25,328 bytes
???_op@1	1	229 µs	0 µs	504 bytes	0 bytes	585 µs	0 µs	31,152 bytes	31,376 bytes
Engine::loopArray	1	145 µs	0 µs	-18,944 bytes	11,224 bytes	2,064 µs	3,999 µs	1,952 bytes	31,760 bytes
main()	1	110 µs	0 µs	-10,528 bytes	0 bytes	13,290 µs	11,998 µs	65,888 bytes	1,709,520 bytes
load::Classes/Vehicle.php	1	60 µs	0 µs	5,256 bytes	6,048 bytes	60 µs	0 µs	5,256 bytes	6,048 bytes
???_op@2	2	19 µs	0 µs	760 bytes	0 bytes	19 µs	0 µs	760 bytes	0 bytes

Callgraph for / on Jun 30th 13:52:18

Methods that represent 1% or less of the entire execution will be omitted from the callgraph result.



XHProf

Der Profiler für PHP-Anwendungen

Der Spezialist

PHPBench

→ PHPBench

Wir entwickeln PHP
und setzen dort beim Optimieren an

GitHub Repo gibt es seit 2015
-> https://github.com/phpbench/phpbench

von Daniel Leech entwickelt

möchte gern das PHPUnit des Benchmarkings werden

Beispiel

negativ

```
PhpBench 0.4. Running benchmarks.

.....

Done (5 subjects, 15 iterations) in 13.12s

  [CarBenchmark->benchCar] [car]
  check complete Car
  +-----+------+----------+--------+-------------+------+---------+----------------+
  | run | iter | time     | memory | memory_diff | revs | rps     | deviation_mean |
  +-----+------+----------+--------+-------------+------+---------+----------------+
  | 0   | 0    | 0.838354s | 448b   | 448b        | 1000 | 1,192.81 | 0.00%          |
  +-----+------+----------+--------+-------------+------+---------+----------------+

  [CarBenchmark->benchEngine] [car]
  move 500 miles
  +-----+------+----------+--------+-------------+------+---------+----------------+
  | run | iter | time     | memory | memory_diff | revs | rps     | deviation_mean |
  +-----+------+----------+--------+-------------+------+---------+----------------+
  | 0   | 0    | 0.580666s | 320b   | 320b        | 1000 | 1,722.16 | -1.30%         |
  | 0   | 1    | 0.591850s | 512b   | 192b        | 1000 | 1,689.62 | +0.60%         |
  | 0   | 2    | 0.592425s | 704b   | 192b        | 1000 | 1,687.98 | +0.70%         |
  +-----+------+----------+--------+-------------+------+---------+----------------+

  [CarBenchmark->benchMultipleCars] [car]
  check 3 cars
  +-----+------+----------+--------+-------------+------+---------+----------------+
  | run | iter | time     | memory | memory_diff | revs | rps     | deviation_mean |
  +-----+------+----------+--------+-------------+------+---------+----------------+
  | 0   | 0    | 0.824745s | 224b   | 224b        | 1000 | 1,212.50 | +1.73%         |
  | 0   | 1    | 0.799997s | 416b   | 192b        | 1000 | 1,250.00 | -1.33%         |
  | 0   | 2    | 0.807524s | 608b   | 192b        | 1000 | 1,238.35 | -0.40%         |
  +-----+------+----------+--------+-------------+------+---------+----------------+

  [CarBenchmark->benchParameterized] [car] ["car-type":"BMW"]
  check different cars
  +-----+------+----------+--------+-------------+------+---------+----------------+
  | run | iter | time     | memory | memory_diff | revs | rps     | deviation_mean |
  +-----+------+----------+--------+-------------+------+---------+----------------+
  | 0   | 0    | 0.944359s | 328b   | 328b        | 1000 | 1,058.92 | -0.10%         |
  | 0   | 1    | 0.950362s | 520b   | 192b        | 1000 | 1,052.23 | +0.54%         |
  | 0   | 2    | 0.940471s | 712b   | 192b        | 1000 | 1,063.30 | -0.51%         |
  | 0   | 3    | 0.945840s | 904b   | 192b        | 1000 | 1,057.26 | +0.06%         |
  +-----+------+----------+--------+-------------+------+---------+----------------+

  [CarBenchmark->benchParameterized] [car] ["car-type":"Audi"]
  check different cars
  +-----+------+----------+--------+-------------+------+---------+----------------+
  | run | iter | time     | memory | memory_diff | revs | rps     | deviation_mean |
  +-----+------+----------+--------+-------------+------+---------+----------------+
  | 0   | 0    | 0.956402s | 192b   | 192b        | 1000 | 1,045.59 | +1.14%         |
  | 0   | 1    | 0.949864s | 384b   | 192b        | 1000 | 1,052.78 | +0.45%         |
  | 0   | 2    | 0.925533s | 576b   | 192b        | 1000 | 1,080.46 | -2.13%         |
  | 0   | 3    | 0.950723s | 768b   | 192b        | 1000 | 1,051.83 | +0.54%         |
  +-----+------+----------+--------+-------------+------+---------+----------------+
```

positiv

```
PhpBench 0.4. Running benchmarks.

.....

Done (5 subjects, 15 iterations) in 12.92s

  [CarBenchmark->benchCar] [car]
  check complete Car
  +-----+------+----------+--------+-------------+------+---------+----------------+
  | run | iter | time     | memory | memory_diff | revs | rps     | deviation_mean |
  +-----+------+----------+--------+-------------+------+---------+----------------+
  | 0   | 0    | 0.815366s | 448b   | 448b        | 1000 | 1,226.44 | 0.00%          |
  +-----+------+----------+--------+-------------+------+---------+----------------+

  [CarBenchmark->benchEngine] [car]
  move 500 miles
  +-----+------+----------+--------+-------------+------+---------+----------------+
  | run | iter | time     | memory | memory_diff | revs | rps     | deviation_mean |
  +-----+------+----------+--------+-------------+------+---------+----------------+
  | 0   | 0    | 0.576473s | 320b   | 320b        | 1000 | 1,734.69 | -0.12%         |
  | 0   | 1    | 0.574288s | 512b   | 192b        | 1000 | 1,741.29 | -0.50%         |
  | 0   | 2    | 0.580791s | 704b   | 192b        | 1000 | 1,721.79 | +0.62%         |
  +-----+------+----------+--------+-------------+------+---------+----------------+

  [CarBenchmark->benchMultipleCars] [car]
  check 3 cars
  +-----+------+----------+--------+-------------+------+---------+----------------+
  | run | iter | time     | memory | memory_diff | revs | rps     | deviation_mean |
  +-----+------+----------+--------+-------------+------+---------+----------------+
  | 0   | 0    | 0.834313s | 224b   | 224b        | 1000 | 1,198.59 | +1.88%         |
  | 0   | 1    | 0.815466s | 416b   | 192b        | 1000 | 1,226.29 | -0.43%         |
  | 0   | 2    | 0.807061s | 608b   | 192b        | 1000 | 1,239.06 | -1.45%         |
  +-----+------+----------+--------+-------------+------+---------+----------------+

  [CarBenchmark->benchParameterized] [car] ["car-type":"BMW"]
  check different cars
  +-----+------+----------+--------+-------------+------+---------+----------------+
  | run | iter | time     | memory | memory_diff | revs | rps     | deviation_mean |
  +-----+------+----------+--------+-------------+------+---------+----------------+
  | 0   | 0    | 0.932936s | 328b   | 328b        | 1000 | 1,071.88 | +0.57%         |
  | 0   | 1    | 0.917962s | 520b   | 192b        | 1000 | 1,089.37 | -1.05%         |
  | 0   | 2    | 0.925537s | 712b   | 192b        | 1000 | 1,080.45 | -0.23%         |
  | 0   | 3    | 0.934256s | 904b   | 192b        | 1000 | 1,070.37 | +0.71%         |
  +-----+------+----------+--------+-------------+------+---------+----------------+

  [CarBenchmark->benchParameterized] [car] ["car-type":"Audi"]
  check different cars
  +-----+------+----------+--------+-------------+------+---------+----------------+
  | run | iter | time     | memory | memory_diff | revs | rps     | deviation_mean |
  +-----+------+----------+--------+-------------+------+---------+----------------+
  | 0   | 0    | 0.940954s | 192b   | 192b        | 1000 | 1,062.75 | +1.43%         |
  | 0   | 1    | 0.929660s | 384b   | 192b        | 1000 | 1,075.66 | +0.21%         |
  | 0   | 2    | 0.913405s | 576b   | 192b        | 1000 | 1,094.80 | -1.54%         |
  | 0   | 3    | 0.926876s | 768b   | 192b        | 1000 | 1,078.89 | -0.09%         |
  +-----+------+----------+--------+-------------+------+---------+----------------+
```

PHPBench

Der wichtigste Ansatzpunkt
für einen Entwickler

Der Tellerrand

What else is out there?

forp

http://forp.io/

Z-Ray

http://www.zend.com/de/products/server/z-ray

Symfony Debug Toolbar

php-meminfo

https://github.com/BitOne/php-meminfo

Blackfire

https://blackfire.io/

Tideways

https://tideways.io/

New Relic

http://newrelic.com/

Es gibt sehr viele Tools. Die Entscheidung ist eine Frage des Einsatzgebietes und Geschmacksache.

Summa Summarum

And what exactly should I do now?

→ Was haben wir gehört?

Das weitläufige Feld des Profilings

Man kann an vielen Stellen ansetzen

Es ist ein Kampf um jede Millisekunde

Wir als PHP-Entwickler sollten uns für dieses Thema sensibilisieren

→ Jetzt beginnen

no pain no gain

Es gibt so viele Ansatzpunkte zur Optimierung.

Dieser Prozess ist nicht von heute auf morgen abgeschlossen.

Man sollte mit einer Messung beginnen.

Definiert euch Ziele.

https://s-media-cache-ak0.pinimg.com/originals/05/fe/e9/05fee97972bb7224e756946274d275c2.jpg

Profiling

„Less performance means less business.“

Fabien Potencier

Das war´s

That´s it! Questions?