



FRONT-END

PERFORMANCE PROFILING

Kai L. Hoppenheidt

Web Basics II

17.06.2013

Bauhaus-Universität
Weimar

Basics.

Why Performance Optimization?

Kurze *response times* sind relevant für eine *gute User Experience*.

Bisherige Case Studies suggerieren eine klare *positive Korrelation zwischen Page Speed und relevanten Business-Metrics* im E-Commerce Bereich (Conversion Rate / Sales etc.).

Bounce Rates und Page Speed korrelieren negativ.

SEO profitiert sowohl direkt als auch indirekt (via verringerter Bounce Rates) von verbessertem Page Speed.

Basics.

Front-End Vs. Back-End Optimization?

The Performance Golden Rule:

*"80-90% of the end-user response time is spent on the frontend.
Start there." (Souders 2007)*

Basics.

5 STEPS // Web Resource Lifecycle

Das Abrufen einer Web-Ressource via Browser kann in fünf Schritte geteilt werden. Diese Schritte helfen bei der Identifikation womöglicher Performance-Schwächen.

Der Browser sendet DNS queries und ermittelt die zur URL gehörige IP.

Resolving

DNS Lookup

Der Webserver generiert den geforderten Content oder liest ihn vom Speicher.

Processing

Server processes request

Client-seitiges Rendern der übertragenen Elemente.

Rendering

Browser renders page



Requesting

resource from web server

Der Browser fordert die benötigte Ressource vom Webserver an.

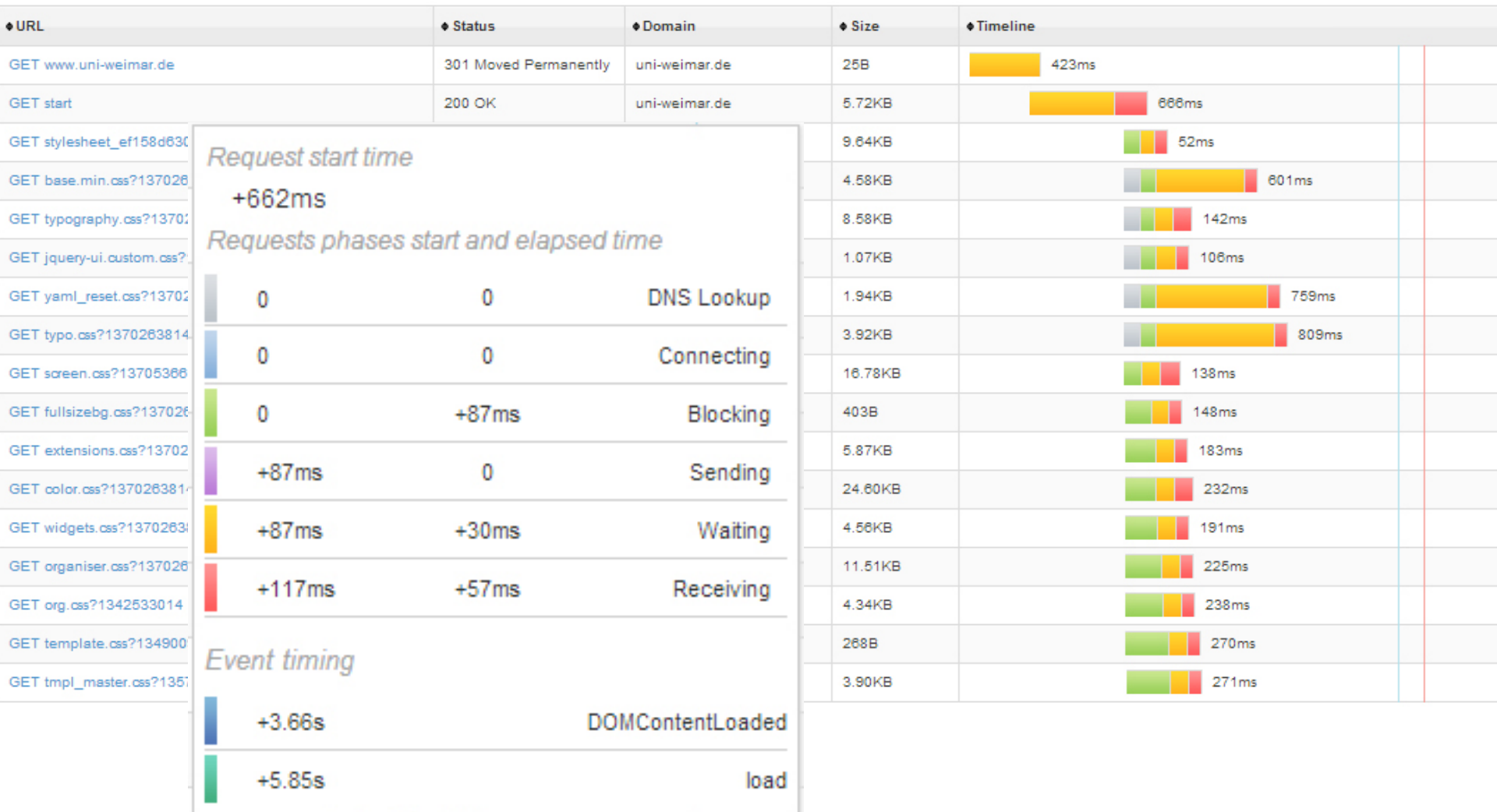
Transferring

Resource is sent to browser

Die angeforderte Ressource wird vom Webserver an den Client übertragen.

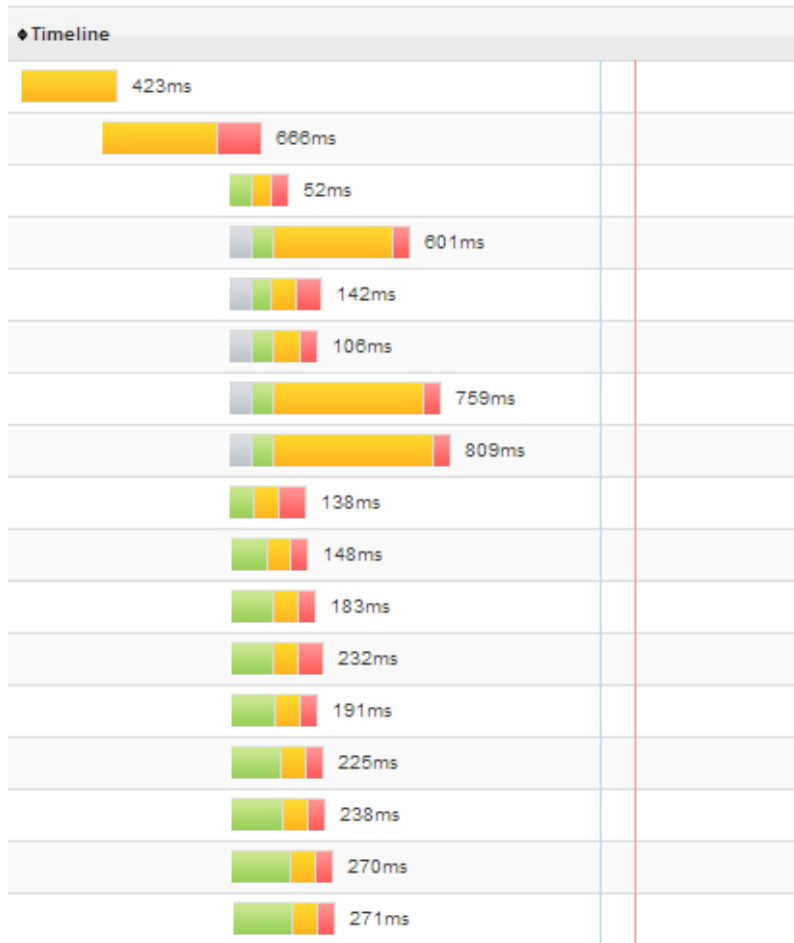
Basics.

Waterfall-Charts



Basics.

Waterfall-Charts



Visualisierung kumulativer, sequentieller Abläufe zwischen Server und Client bei Abruf einer Website.

Anzahl der Zeilen entspricht Anzahl der geladenen Objekte / Requests (z.B. Text, Bilder, Scripts).

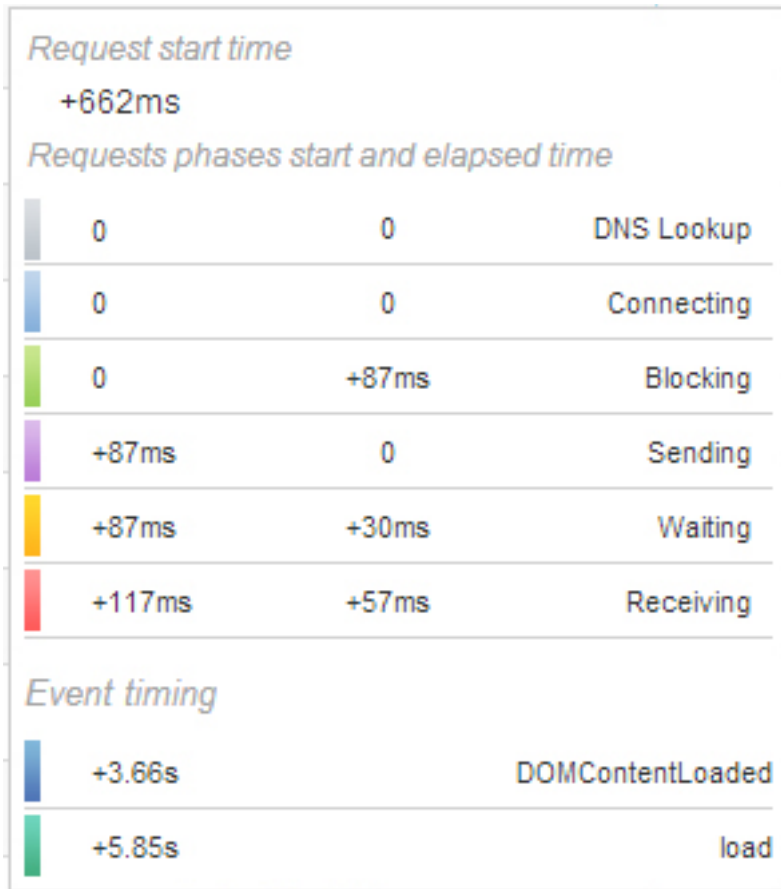
Die **hellblaue Linie** gibt den Startpunkt des Renderns an.

Die **rote Linie** indiziert den Zeitpunkt an dem das DOM fertiggestellt ist. HTML und JavaScript sind vollständig geparsed.

Das **rechte Ende der Tabelle** gibt den Zeitpunkt an, an dem alle Objekte vollständig geladen und gerendert sind.

Basics.

Waterfall-Charts



DNS-Lookup: Domains der angeforderten Objekte werden übersetzt, kaum optimierbar.

Connecting: TCP Connection wird aufgebaut.

Blocking: Der Browser wartet auf einen Download oder auf die Ausführung eines Scripts.

Sending: Der Browser sendet den Request an den Webserver.

Waiting: Der Browser wartet auf die Übertragung des Content. Dieser Block wird auch als „*Time to first Byte*“ bezeichnet.

Receiving: Der Server sendet den Content an den Browser.

Tools.

BROWSER PLUG-INS & WEB SERVICES

Built-In Developer Tools

Firebug

YSlow

PageSpeed Insights

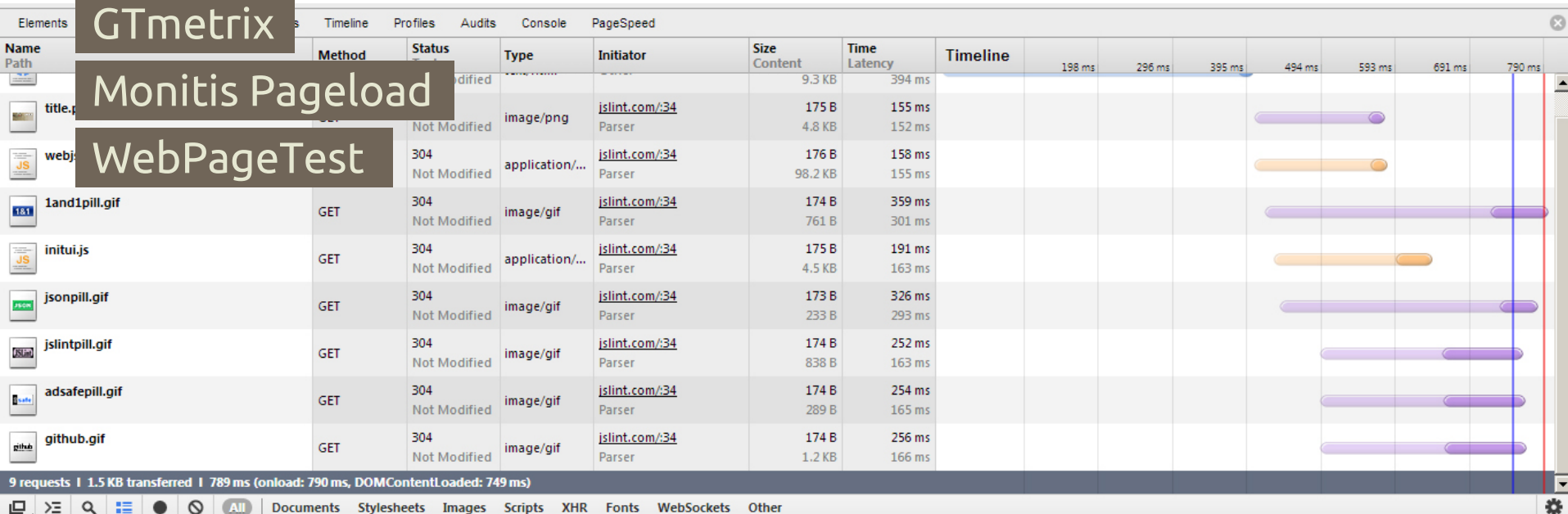
Speed Tracer

GTmetrix

Monitis Pageload

WebPageTest

- Chrome, Safari, Opera und Internet Explorer bieten bereits integrierte Entwickler-Tools.
- Übersicht über alle Objekte der geladenen Seite.
- Angabe von Waterfall-Charts.
- JavaScript & CSS Performance + Debugging.
- Daneben noch zahlreiche weitere Features.



Tools.

BROWSER PLUG-INS & WEB SERVICES

Built-In Developer Tools

Firebug

YSlow

PageSpeed Insights

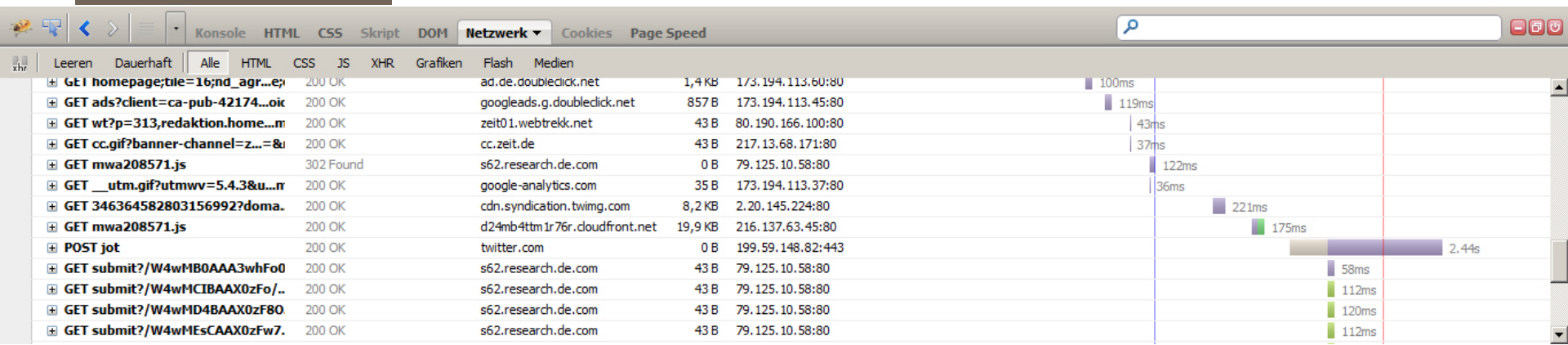
Speed Tracer

GTmetrix

Monitis Pageload

WebPageTest

- Browser Add-On für Firefox, das die Funktionalitäten anderer Developer-Tools integriert.
- Übersicht über alle Objekte der geladenen Seite.
- Angabe von Waterfall-Charts.
- JavaScript & CSS Performance + Debugging.
- <https://getfirebug.com/>



Tools.

BROWSER PLUG-INS & WEB SERVICES

Built-In Developer Tools

Firebug

YSlow

PageSpeed Insights

Speed Tracer

GTmetrix

Monitis Pageload

WebPageTest

- Bewertet Page Speed anhand eines vordefinierten aber modifizierbaren Regelsets.
- Gibt auf Basis der Anwendung dieses Regelsets eine Note.
- Führt Verbesserungsvorschläge an.
- Gibt Übersicht über alle Objekte der Seite.
- Verfügbar als Plug-In für Chrome, Firefox, Opera, Safari & als Source Code.
- <http://yslow.org/>

Grade **B** Overall performance score 84 Ruleset applied: YSlow(V2) URL: <http://yslow.org/>

ALL (23) FILTER BY: [CONTENT \(6\)](#) | [COOKIE \(2\)](#) | [CSS \(6\)](#) | [IMAGES \(2\)](#) | [JAVASCRIPT \(4\)](#) | [SERVER \(6\)](#)

[Tweet](#) [Share](#)

B Make fewer HTTP requests

F Use a Content Delivery Network (CDN)

A Avoid empty src or href

F Add Expires headers

A Compress components with gzip

A Put CSS at top

A Put JavaScript at bottom

A Avoid CSS expressions

Grade B on Make fewer HTTP requests

This page has 7 external Javascript scripts. Try combining them into one.

Decreasing the number of components on a page reduces the number of HTTP requests required to render the page, resulting in faster page loads. Some ways to reduce the number of components include: combine files, combine multiple scripts into one script, combine multiple CSS files into one style sheet, and use CSS Sprites and image maps.

[»Read More](#)

Tools.

BROWSER PLUG-INS & WEB SERVICES

Built-In Developer Tools

Firebug

YSlow

PageSpeed Insights

Speed Tracer

GTmetrix

Monitis Pageload

WebPageTest

PageSpeed Tools

- Verfügbar via Web und als Browser Extension für Chrome und Firebug.
- Verfügbarkeit eines API.
- Führt Test auf Basis eines vordefinierten Regelsets aus (basierend auf Best Practices).
- Ergebnis: *Page Speed Score* & Vorschläge zur Optimierung.
- <https://developers.google.com/speed/pagespeed/>

Analyze and optimize your website with PageSpeed tools to implement the web performance best practices.



Tools.

BROWSER PLUG-INS & WEB SERVICES

Built-In Developer Tools

Firebug

YSlow

PageSpeed Insights

Speed Tracer

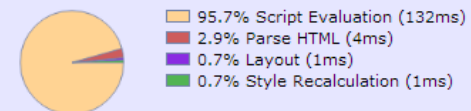
GTmetrix

Monitis Pageload

WebPageTest

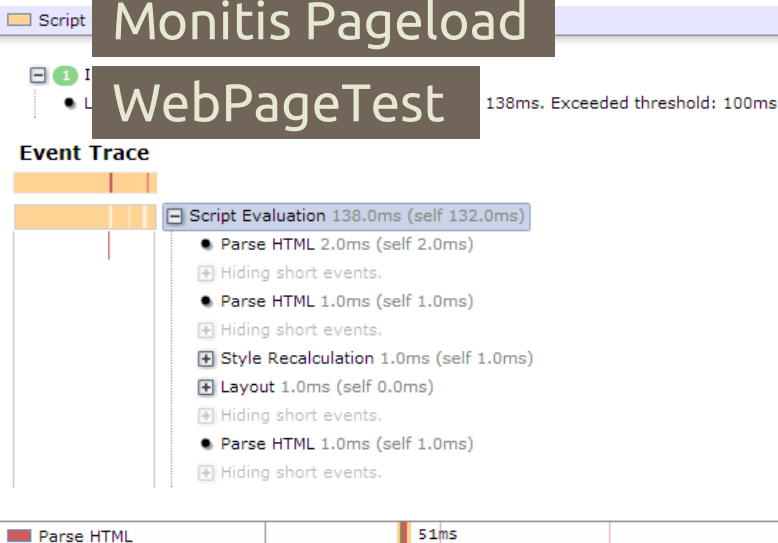
138ms. Exceeded threshold: 100ms

- Chrome Browser Extension für detaillierte Performance-Messungen an Web Applications.
- Analysiert, an welcher Stelle der App am meisten Zeit gebraucht wird (JavaScript Parsing und Execution, DOM event handling, network resource loading etc).
- Waterfall-Charts und Event-Analyse.
- <https://developers.google.com/web-toolkit/speedtracer/>



Details for Script Evaluation

Description	A block of JavaScript was parsed/compiled and executed. This only includes script encountered via an HTML <script> tag.
@	197866ms
Duration	138.000ms
Url	http://www.uni-weimar.de/uploads/tx_t3jquery/jquery-1.8.x-1.8.x-1.2.x.js?1357114070
Line Number	1



Tools.

BROWSER PLUG-INS & WEB SERVICES

Built-In Developer Tools

Firebug

YSlow

PageSpeed Insights

Speed Tracer

GTmetrix

Monitis Pageload

WebPageTest

- Angabe von *Page Speed Score* und *YSlow Grade*.
- Angabe von Waterfall-Charts.
- History Monitoring, Graphs & Alerts.
- Android Emulation.
- In PRO-Version: Remote Location Testing, Test Scheduling, API Funktionen.
- <http://gtmetrix.com/>



Latest Performance Report for: <http://www.uni-weimar.de/>

 [Download PDF](#)

Report generated: Mon, Jun 17, 2013, 1:45 AM -0700

Test Server Region: Vancouver, Canada

Using: Firefox (Desktop) 14.0.1, Page Speed 1.12.16, YSlow 3.1.6

Summary

Options

- [Re-Test Page](#)
- [Compare to another URL](#)

Page Speed Grade:
(57%)↓

E

YSlow Grade:
(68%)↓

D

Page load time: 9.03s
Total page size: 756KB
Total number of requests: 50

Tools.

BROWSER PLUG-INS & WEB SERVICES

Built-In Developer Tools

Firebug

YSlow

PageSpeed Insights

Speed Tracer

GTmetrix

Monitis Pageload

WebPageTest

- Page Speed Monitoring über USA, Europa und Asien.
- Angabe von Waterfall-Charts.
- Angabe von Request- und Byte-Statistiken.
- <http://pageload.monitis.com/>

UNITED STATES 3.13s ASIA/PACIFIC 20.29s

Domain

Bytes by domain



◆ URL	◆ Status	◆ Domain	◆ Size	◆ Timeline
GET www.uni-weimar.de	301 Moved Permanently	uni-weimar.de	25B	988ms
GET start	200 OK	uni-weimar.de	5.72KB	809ms
GET s64429589922885?AQB=1&ndh=1&t=1	200 OK	montblanc.122.2o7.net	43B	205ms
GET stylesheet_ef158d630f.css?13656	200 OK	uni-weimar.de	9.64KB	173ms

Tools.

BROWSER PLUG-INS & WEB SERVICES

Built-In Developer Tools

Firebug

YSlow

PageSpeed Insights

Speed Tracer

GTmetrix

Monitis Pageload

WebPageTest

web Page Performance Test for

www.uni-weimar.de/

From: Buenos Aires, Argentina - Firefox - Cable
17.6.2013 11:33:06

[Summary](#)

[Details](#)

[Performance Review](#)

[Content Breakdown](#)

[Domains](#)

[Screen Shot](#)

[Re-run the test](#)

[Raw page data](#) - [Raw object data](#)
[Export HTTP Archive \(.har\)](#)
[See in ShowSlow](#)

		Document Complete				Fully Loaded		
		Load Time	First Byte	Start Render	DOM Elements	Time	Requests	Bytes In
First View		8.568s	1.835s	0.000s	1103	8.568s	52	795 KB
Repeat View		4.023s	1.066s	0.000s	1103	4.023s	33	58 KB

- Verschiedene Internetverbindungen und Browser simulierbar.
- Angabe von Waterfall-Charts, Connection-Charts, CPU-Utilization, Bandwith-Graph, Request- und Byte-Statistiken.
- Optimization-Checklist für jeden Request mit 6 Kriterien.
- Mobil-Testing, Video-Playback.
- RESTful API.
- <http://www.webpagetest.org/>



Problems.

Measurement Difficulties

Nicht analysierbare Faktoren beim Client:

1. Die Bandbreiten der Endnutzer variieren stark. 2. Die client-seitige Rechenkapazität nimmt Einfluss auf die Performance. 3. Zeitlich verteilte Messungen sind nicht reliabel. 4. Waterfall-Charts geben nicht an, wann ein User die Seite als benutzbar wahrnimmt. 5. Abhängigkeit des Page Speeds vom benutzten Browser.

Optimization.

BEST PRACTICES

Minimize HTTP Requests

Compression

Minify CSS and JavaScript

Script & StyleSheet Placement

Merging external Scripts and CSS

Avoid CSS Expressions

Use Content Delivery Network

Avoid Redirects

Remove Duplicate Scripts

Add Expires Header

Reduce DNS Look-Ups

Configure ETags

Make AJAX Cacheable

- Souders, Steve (2007): 14 Rules for Faster Loading Web Sites.
<http://stevesouders.com/hpws/rules.php>
- Google Developers:
https://developers.google.com/speed/docs/best-practices/rules_intro
- Yahoo Developer Network:
<http://developer.yahoo.com/performance/rules.html>

References.

LITERATURE & LINKS

Bursztein, Elie (2011): Analyzing Web application Performance.

<http://elie.im/blog/web/analyzing-web-application-performance/#.Ub3K5OdM8hF>

Everts, Tammy (2010): Waterfalls 101: How to understand your website's performance via waterfall chart.

<http://www.webperformancetoday.com/2010/07/09/waterfalls-101/>

Everts, Tammy (2012): 4 awesome slides showing how page speed correlates to business metrics at Walmart.com

<http://www.webperformancetoday.com/2012/02/28/4-awesome-slides-showing-how-page-speed-correlates-to-business-metrics-at-walmart-com/>

Gaebel, Warren (2012): How to Use a Waterfall Chart.

<http://blog.monitor.us/2012/06/how-to-use-a-waterfall-chart/>

Google Developers: Web Performances Best Practices.

https://developers.google.com/speed/docs/best-practices/rules_intro

Nielsen, Jakob (1993): Response Times: 3 Important Limits.

<http://www.nngroup.com/articles/response-times-3-important-limits/>

Souders, Steve (2007): 14 Rules for Faster-Loading Web Sites. <http://stevesouders.com/hpws/rules.php>

Souders, Steve (2011): Waterfalls UI Conventions.

<http://www.stevesouders.com/blog/2011/08/26/waterfall-ui-conventions/>

Souders, Steve (2012): The Performance Golden Rule.

<http://www.stevesouders.com/blog/2012/02/10/the-performance-golden-rule/>

Yahoo Developer Network: Best Practices for Speeding Up Your Website.

<http://developer.yahoo.com/performance/rules.html>