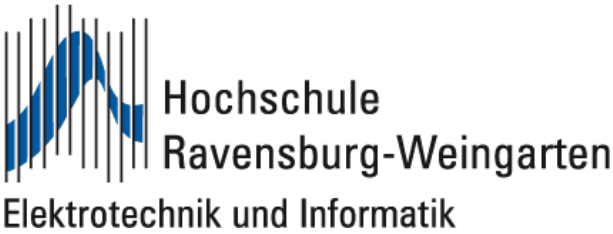

Bachelor-Thesis
Web Performance für den mobilen Endanwender

Zusammenfassung

3. März 2015

Columbus Interactive
Eywiesenstraße 6
88212 Ravensburg

Fakultät Elektrotechnik und Informatik
Studiengang Angewandte Informatik
Hochschule Ravensburg-Weingarten
Doggenriedstraße, 88250 Weingarten

Autor:
Andreas Lorer
andreas.lorer@hs-weingarten.de
88250 Weingarten
Wilhelmstraße 4

Bachelor-Thesis

Web Performance für den mobilen Endanwender

Inhaltsverzeichnis

1 Einleitung

2 Motivation

Larry Page, CEO und Mitgründer von Google, sagt:

> *„As a product manager you should know that speed is the number one feature.“*(Holzle 2010)

Niemand mag es zu warten, auch nicht auf eine Website. Die Studie „The Psychology of Web Performance“ kam bereits im Jahr 2008 schon auf folgende Ergebnisse:

> *„Slow web pages lower perceived credibility and quality. Keep your page load times below tolerable attention thresholds, and users will experience less frustration, lower blood pressure, deeper flow states, higher conversion rates, and lower bailout rates. Faster websites are actually perceived to be more interesting and attractive.“ (WebSiteOptimization.com 2008)*

Das haupt Vermarktungsargument für den Chrome Browser war damals: er sei schneller als die Konkurenz. Tatsächlich ist für Google Geschwindigkeit alles. Deshalb hat Google im Jahr 2010 angekündigt, dass Geschwindigkeit in die Berechnung des `Page Rankings` mit einfließt.

> *„Faster sites create happy users and we've seen in our internal studies that when a site responds slowly, visitors spend less time there. [...] Recent data shows that improving site speed also reduces operating costs. Like us, our users place a lot of value in speed — that's why we've decided to take site speed into account in our search rankings“(Google 2010)*

Aktuell (2015) geht Google sogar noch einen Schritt weiter und informiert tausende Webmaster per E-Mail über die schlechte Usability ihrer Websites für mobile Besucher und warnt ausdrücklich vor dementsprechend „angepassten Rankings“.(t3n 2015) Im Hinblick auf die Zukunft wird der Marktanteil an mobilen Internetnutzern noch weiter wachsen und die Optimierung der Ladezeiten gewinnt dadurch noch mehr an bedeutung. Zwischen 2011 und 2014 stieg die Anzahl der Smartphone nutzer von 18% auf 50% an. Dies ist ein Wachstum von 32% innerhalb von nur 3 Jahren.(TNS Infratest 2014)

Die Antwort auf diesen Trend läutete eine Ära ein, die wir heute unter dem Namen `Responsive Webdesign` kennen. „Responsive“ muss aber sehr viel mehr bedeuten, als nur eine angepasste Darstellung für eine bestimmte Art von Gerät. „Two out of three mobile shoppers expect pages to load in 4 seconds or less.“ (Radware 2013). Der Anwender erwartet also auf dem Smartphone änliche oder gleiche Ladezeiten wie er auch von der Nutzung eines Desktop-Pc's gewohnt ist. Diese Erwartungen werden von dem Großteil der im Internet besuchbaren Seiten nicht erfüllt.
Der Inhalt einer Seite muss darum so aufbereitet werden, dass dieser auch auf Geräten mit langsamer Internetverbindung, hoher Latenz und einem begrenzten Datentarif, in einer für den Anwender annehmbaren Geschwindigkeit, angezeigt werden kann.

2.1 Zielsetzung

Um gängige Methoden und Techniken der Ladezeit Optimierung anzuwenden wird das Projekt anhand der Website `http://andreaslorer.de` durchgeführt. Das Ziel ist es, die Ladezeit der

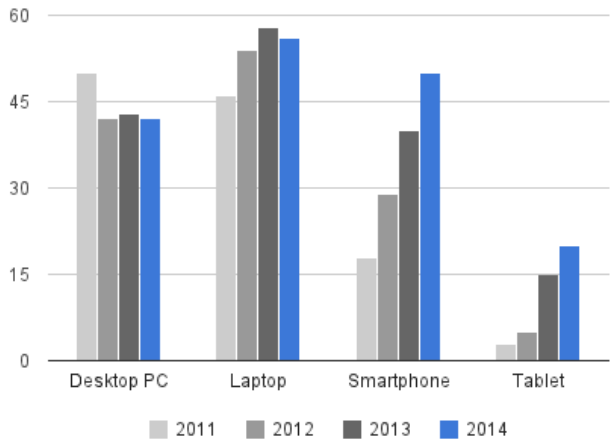

Abbildung 1: Gerätenutzung in der Gesamtbevölkerung (2011 – 2014)(TNS Infratest 2014)

Website auf dem Smartphone, als auch auf dem Desktop von 10 Sekunden auf unter 1 Sekunden zu verringern. Mit Ladezeit ist dabei nicht die Zeit gemeint, die benötigt wird um die Website komplett zu laden, sondern die Zeit bis ein erste visuelle Rückmeldung für den Anwender zu sehen ist. Diese vom Anwender wahrgenommene Rückmeldung nennt man auch „Perceived Performance“ und bedeutet, dass die Ladezeit als schneller empfunden wird, als es eigentlich laut Messwerten der Fall ist. Näheres dazu wird in Punkt 5.6 beschrieben.

3 Eigene Leistung

Meine Leistung besteht darin, einen Leitfaden zu erstellen, der einen Gesamtüberblick ermöglicht. Die Arbeit soll es dem Leser ermöglichen Fehler in der Struktur von Webanwendungen zu finden, die für die Geschwindigkeit hinderlich sind. Es soll herausgefunden werden, was die „Best Practices“ sind um die Ladezeit zu minimieren, wie ein moderner „workflow“ aussehen kann, damit eine Webanwendung schon bei seiner Entstehung schnell ladet und im Projektverlauf schnell bleibt. Des weiteren soll erklärt werden, was für herausforderungen es zu meistern gilt um eine schnelle Webanwendung zu erreichen, welche Tools es gibt und welche Vor- oder Nachteile diese mit sich bringen.
Diese Arbeit befasst sich nicht, mit der Geschwindigkeit von Datenbanken, SQL-Abfragen oder sonstigen problemen die durch einen Engpass ein schnelles Laden der Seite verhindern könnten.

4 Ist-Zustand

Die Webseite ist auf einem `shared Hosting` [1] aufgesetzt und antwortet auf ein Ping Kommando in rund 13ms. Dadurch, dass es keine Möglichkeit gibt `root Rechte` [2] auf einem shared hosting zu bekommen können so manche serverseitige Einstellungen nicht durchgeführt werden. Diese werden dann zwar Aufgezeigt, aber kommen für dieses Projekt nicht zum Einsatz.
Die Website hat als Ausgangsbasis einen gängigen Aufbau. Sie besteht aus einer Bilder Gallerie basierend auf PHP und dem Bootstrap Framework.

5 Begriffe

5.1 Pattern und Antipattern

„Entwurfsmuster (englisch design patterns) sind bewährte Lösungsschablonen für wiederkehrende Entwurfsprobleme sowohl in der Architektur als auch in der Softwarearchitektur und -entwicklung. Sie stellen damit eine wiederverwendbare Vorlage zur Problemlösung dar, die in einem bestimmten Zusammenhang einsetzbar ist.“(Wikpedia 2015)

„Während „Design Patterns“ in der Software-Entwicklung allgemein übliche und bekannte Ansätze sind, um Probleme zu lösen, sind Anti-Patterns Negativ-Beispiele – die zeigen, wie man es nicht macht - von bereits durchgeführten, gescheiterten Projekten, die dem erkennenden Mitarbeiter zielgerichtete Hinweise darauf geben, wie die Aufgabenstellung besser gelöst werden könnte. Als Synonym ist auch der Begriff Negativmuster im Gebrauch. Es ist tatsächlich möglich, daß das, was gestern noch als allgemein gangbarer Lösungsweg bezeichnet wurde, heute schon ein „Antipattern“ ist [...]“ (Stepken 2006)

5.2 TCP Three Way Handshake

TCP ist das meistgenutzte Verbindungsprotokoll im Internet. Auf diesem Protokoll wird der HTTP Request aufgebaut, der die eigentlichen Daten enthält. Bevor Daten zwischen Server und Browser ausgetauscht werden können, muss eine Verbindung aufgebaut werden. Abbildung 2 beschreibt den Prozess des Verbindungsaufbaus.[3]

5.3 TCP Slow Start

Ein Round Trip[4] kann nicht beliebig viele Bytes transportieren sondern ist durch die sogenannte „Congestion Window Size“[5] limitiert. Der Überbegriff für dieses Verhalten nennt sich „Slow Start“

- Congestion Control: Nach dem eine neue Verbindung per TCP aufgebaut wurde, können weder Server noch Client wissen, wie schnell die Verfügbare Bandbreite ist, mit der Daten ausgetauscht werden können. Um das Netzwerk, vor einem Datenstau zu schützen, wird mit

[1] Bei shared Hosting werden mehrere Websites von verschiedenen Website-Betreibern von dem gleichen Webserver gehostet. Bei Shared Hosting teilen sich in der Regel Hunderte andere Websites einen Server (ItWissen.info 2015)

[2] Standardmäßig existiert unter Linux immer ein Konto für den Benutzer „root“ mit der User-ID 0. Dies ist ein Systemaccount mit vollem Zugriff auf das gesamte System, und damit auch auf alle Dateien und Einstellungen aller Benutzer (wiki.ubuntuusers 2014)

[3] Für ein tieferes Verständnis empfiehlt sich dieser Artikel: High Performance Browser Networking - Chapter 2: Building Blocks of TCP

[4] „Round Trip Time“ wird im Deutschen Paketumlaufzeit genannt. Es bezeichnet die Zeit die ein Datenpaket braucht um in einem Netzwerk von Sender A zu Empfänger B und wieder zurück zu gelangen.

[5] engl. congestion: Stauung, Überlastung, Anhäufung

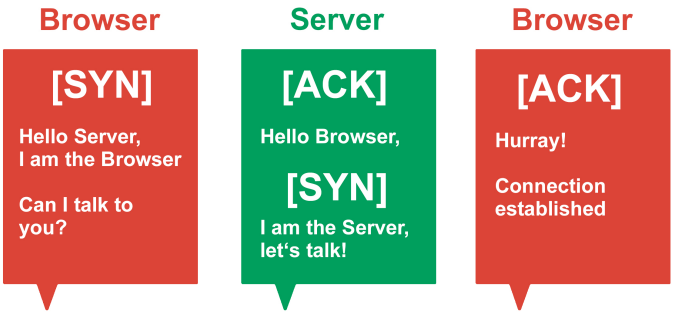

Abbildung 2: Three-Way-Handshake zum Aufbau einer TCP Verbindung zwischen Browser und Server (Eigene Abbildung nach: (Stefanov 2011))

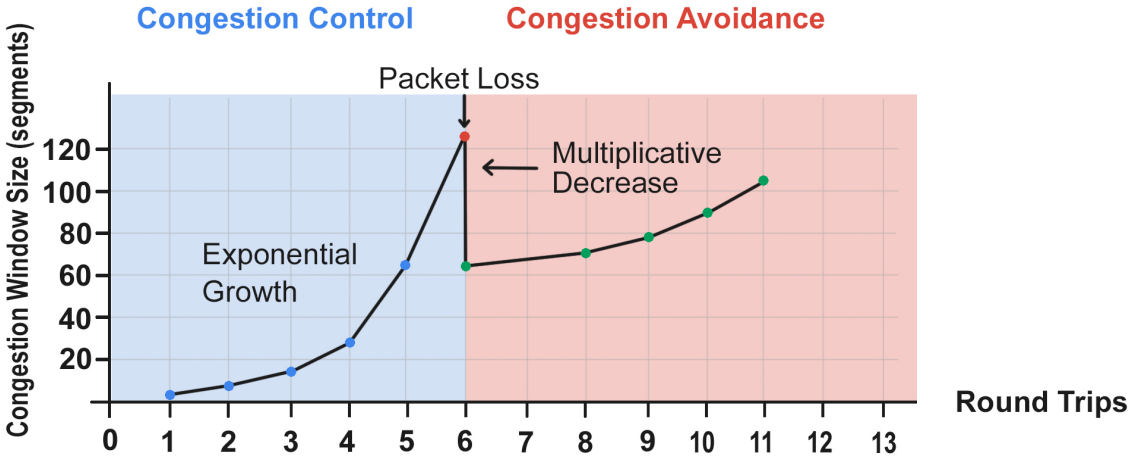

Abbildung 3: Congestion Control und Congestion Avoidance (Eigene Abbildung nach (Grigorik 2013a))

einem sehr niedrigen Wert begonnen, der dann ansteigt bis das Limit erreicht ist. Dieses Verhalten nennt sich auch „Congestion Control“ und verhindert das Aufstauen von Daten.

- Congestion Window Size: Diese Größe bestimmt, wieviel Bytes der pro Segmente geschickt werden darf, bis diese vom Empfänger per `ACK` (acknowledgement) bestätigt werden müssen. Die Größe der Segmente ist Standardmäßig 1460 bytes und die Rate bis zum ACK ist im April 2013 von 4 auf 10 Segmente erhöht worden.(Grigorik 2013a). In der Grafik wird davon ausgegangen, das der erste Round Trip 4 Segmente senden darf. Die Datenrate Wächst exponentiell an, damit möglichst schnell die volle Bandbreite nutzbar ist.

- Congestion Avoidance bedeutet, dass sich die Datenrate wieder um ein Vielfaches verringert, falls es zu einem Paketverlust kommt. Da es besonders bei WLAN oder Mobilfunknetzen des öfteren zu Packetverlusten kommen kann ist dieser Aspekt besonders hervorzuheben, denn er verzögert das erreichen der maximal möglichen Datenrate.

Slow Start bedeutet also aus sicht der Performance, dass bei einer neuen TCP Verbindung nicht die maximale Bandbreite zu Verfügung steht. Bei größeren Dateien wird zwar durch das exponentielle Wachstum das Maximum schnell erreicht, gerade aber bei kleineren Dateien mit wenigen Kilobyte ist dies oft nicht der Fall.

5.4 Der HTTP-Request

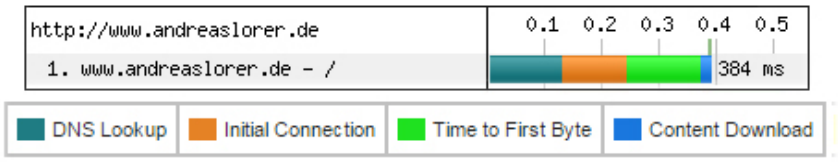

Abbildung 4: Anfrage der HTML-Datei vom server http://andreaslorer.de (Abbildung nach `http://webpagetest.org`)

- DNS Lookup: Um eine Verbindung mit dem Server herzustellen benötigt das HTTP Protokoll die IP Adresse des Ziels. Das heißt beim DNS Server wird für den Namen „http://andreaslorer.de“ die zu diesem Namen zugehörige IP Adresse zurückgegeben.

- Initial Connection bezeichnet die Zeit die vergeht, bis eine Verbindung zum Server hergestellt wurde und eine Kommunikation zwischen Browser und Server stattfinden kann. Hierbei findet der sogenannte TCP „Three-Way-Handshake“ statt.

- TTFB: Ist die Abkürzung für „Time to first byte“. Dieser Begriff beschreibt die Zeit die vergeht, bis das erste Byte vom Server beim Browser ankommt. Die Geschwindigkeit und Güte des Servers, Zugriffzeiten auf eine Datenbank oder die Entfernung des Servers können diesen Wert hauptsächlich beeinflussen.

- Content Download: Die Zeit die benötigt wird bis die Datei vom Server Heruntergeladen wurde.

Wie in Abbildung 4 zu sehen ist, ist die Zeit für das Herunterladen des HTML Dokuments verschwindend gering im Vergleich zu der Zeit die damit verbracht wurde, um das Herunterladen zu ermöglichen.

5.5 Above The Fold

Damit ist der auf einem Bildschirm sichtbare Bereich vor dem Scrollen gemeint. Diesem Bereich wird eine besondere Wichtigkeit zugesprochen.

Abbildung 5: Darstellung des sichtbaren Bereichs vor dem Scrollen

> *„In an analysis of 57,453 eyetracking fixations, we found that there was a dramatic drop-off in user attention at the position of the page fold. Elements above the fold were seen more than elements below the fold: the 100 pixels just above the fold were viewed 102% more than the 100 pixels just below the fold.“* (Schade 2015)

Wichtige Informationen oder Navigationselemente sind meistens dort zu finden. Eine Webseite die nach dem Paradigma des Responsive-Webdesign aufgebaut ist kann dabei 3 oder mehrere Ansichten haben die alle einen unterschiedlichen „above the fold“ bereich haben. Eine Anwendung kann aber auch unterschiedliche Seiten haben, auf dem der Anwender beim Aufrufen der Seite landen kann. Zum Beispiel wenn dieser An- oder Abgemeldet ist. Paradebeispiel dafür sind Facebook oder Twitter.

5.6 Perceived Performance

Abbildung 6 zeigt die Seiten A und B, mit nahezu identischer Ladezeit. Der Unterschied besteht darin, dass Seite B bereits nach 1.5 Sekunden eine erste visuelles Rückmeldung für den Anwender zu sehen ist wohingegen Seite A erst nach 5.5 Sekunden dem Anwender zeigt, dass sie überhaupt ladet. „Perceived Performance“ steht also für die Zeit bis ein erste visuelle Rückmeldung für den Anwender zu sehen ist und bedeutet, dass die Ladezeit als schneller empfunden wird, als es eigentlich laut Messwerten der Fall ist. Warum diese „Perceived Performance“ für eine Webanwendung so wichtig ist zeigen mehrere Studien, deren Daten in folgender Infographik aufbereitet sind.
Bereits kleine Verbesser- oder Verschlechterungen der Ladezeit können einen großen Einfluss in auf den Anwender haben. Yahoo hat herausgefunden, dass wenn eine Seite um nur 400 Millisikunden schneller ist, sich der Traffic um 9% erhöhte.(Stefanov 2008) 57% der Online Konsumenten haben eine Seite, die länger als 3 Sekunden ladet bereits wieder verlassen. 78% der Anwender empfinden sogar Zorn oder Stress wenn eine Seite nicht Ladet oder dies nicht ersichtlich ist.

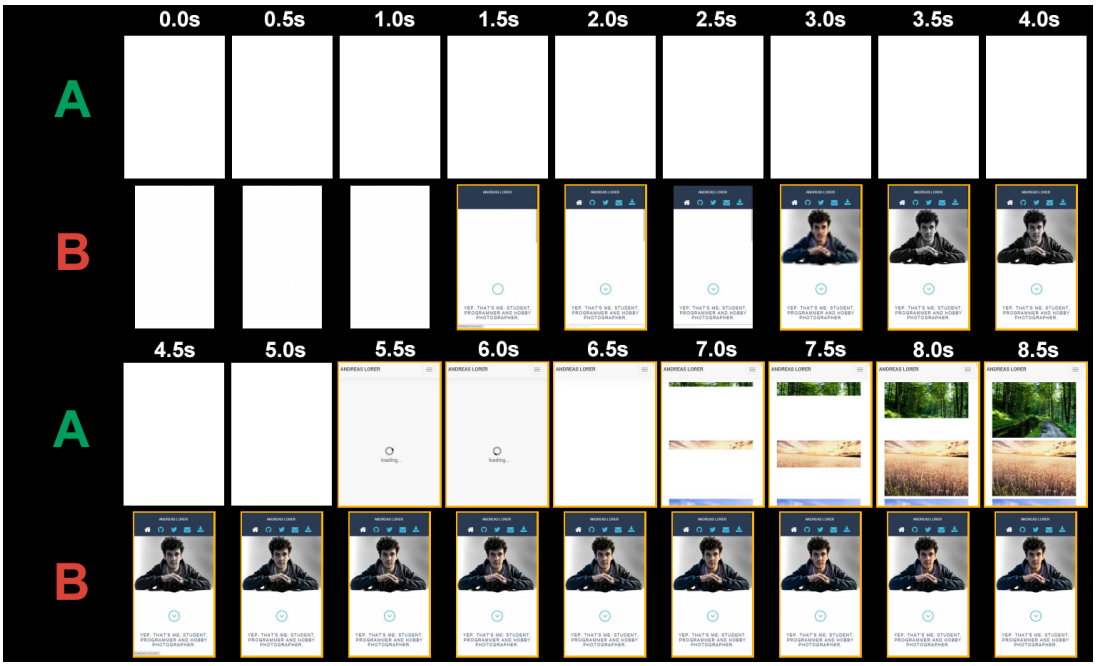

Abbildung 6: Zwei Seiten im Vergleich (Eigene Abbildung via webpagtest.org)

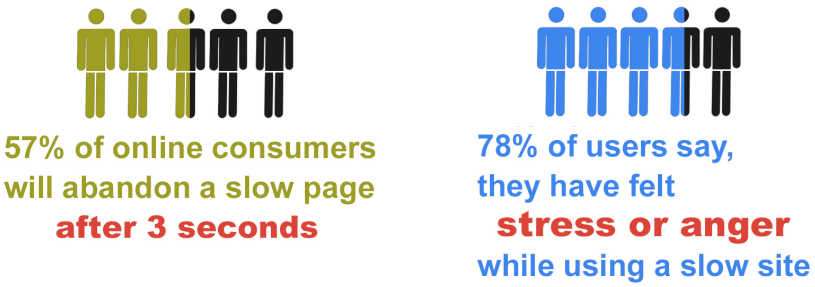

Abbildung 7: Einfluss und Effekt einer langsamen Seite auf den Anwender (Eigene Abbildung nach Daten von: (Radware 2014, p. 8))

6 Die 1000 ms Barriere

Das Ziel dieser Arbeit, die 1000 Millisekunden Barriere zu durchbrechen, wurde nicht durch einen Zufall gewählt. Der Anwender nimmt die Geschwindigkeit einer Seite subjektiv wahr. Sie wird in der folgenden Grafik interpretiert:

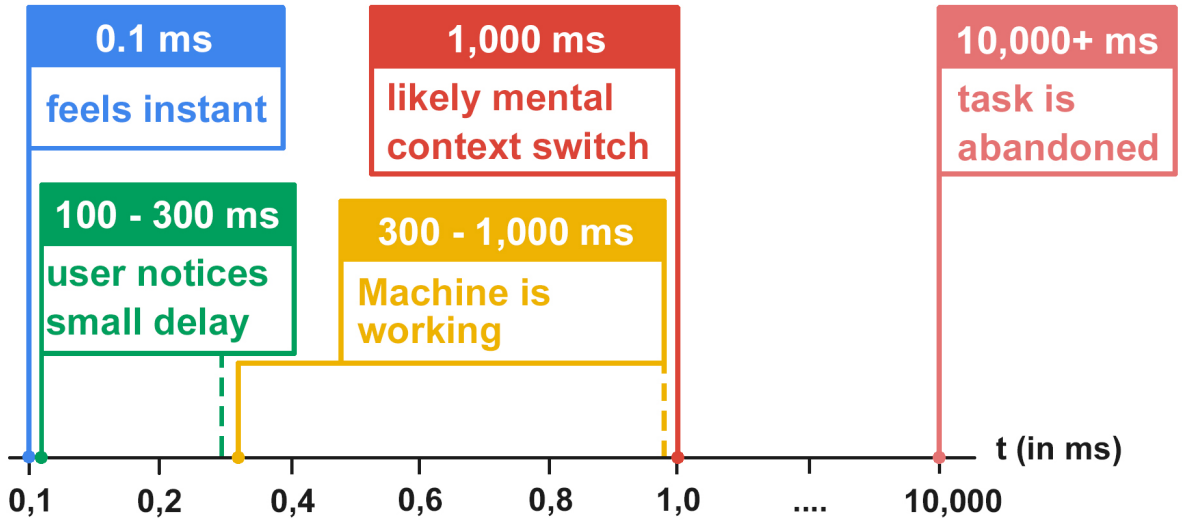

Abbildung 8: Zeit und Wahrnehmung durch den Anwender (Abbildung nach Daten von: (Girgorik 2013))

Wie zu sehen ist bleiben gerade einmal eine Sekunde, bevor das Gehirn uns sagt, man solle doch einer anderen Aufgabe nachgehen bis der Ladevorgang abgeschlossen ist. Der Anwender verlangt visuelle Rückmeldung um „am Ball zu bleiben“, dies wurde bereits in Punkt 5.6 „Perceived Performance“ angesprochen. Auf vielen Webseiten sieht man deshalb, Ladebalken oder sogenannte `Spinner`, die dem Anwender sagen, dass der Ladevorgang in gange, aber noch nicht abgeschlossen ist.
Um das Ziel von einer Sekunde Ladezeit bis zum ersten Render zu erreichen, ist es nötig zu verstehen womit die meiste Zeit beim Aufrufen einer Webanwendung verbracht wird. Bevor eine Seite mittels Smartphone vom Browser dargestellt wird läuft eine ganze Reihe von Prozessen ab.

6.1 Touch Event

Der Aufruf einer Seite über das Smartphone erfolgt über ein Touch Event auf einen Link, Button oder die Seite wird per URL aufgerufen. Hierbei können je nach Gerät zwischen 50 (IPhone 5) und 123 Millisekunden (Moto X - Android) zwischen der Berührung des Touch Screen und dem Registrieren des Events vergehen.(Takahashi 2013) Der Browser wartet allerdings nochmals bis zu 300 Millisekunden, denn er muss abwarten ob vielleicht noch ein zweiter Finger aufgelegt wird, oder ob der Anwender Scrollen oder Zoomen möchte.(Google 2011)
Dieses Verhalten lässt sich bei vielen Browsern per `Meta Tag` abstellen:

```
1                    <meta name="viewport" content="user-scalable=no">
```

Dies setzt natürlich voraus, dass die Webanwendung kein Zoomen benötigt um sie zu bedienen! Gerade bei älteren Webseiten trifft das oftmals nicht zu. Eine vollständige Liste mit Meta Tags

für die verschiedenen Browser ist der Fußnote zu entnehmen.[6]

6.2 Netzwerke

Gernation	Data rate	Latency
2G	100 - 400 Kbit/s	300 - 1000 ms
3G	0,5 - 5 Mbit/s	100 - 500 ms
4G	1-50 Mbit/s	< 100 ms

Abbildung 9: Datenrate und Latenz für eine aktive mobile Verbindung (Eigene Abbildung nach Tabelle (Grigorik 2013b))

6.3 Kritischer Rendering-Pfad

Auf Englisch „critical render path“ genannt, ist der wohl wichtigste Begriff, wenn es um schnelle Ladezeiten geht. Durch die Optimierung des Rendering Pfads kann die benötigte Zeit für das erste Rendern der Seite erheblich verkürzt werden. Das Verständnis des Rendering-Pfads ist zudem eine wesentliche Voraussetzung für die Erstellung von schnellen Webanwendungen und soll in diesem Abschnitt ausführlich erklärt werden. Dabei wird der Begriff in seine zwei Teile Zerlegt: Kritischer und Rendering-Pfad.

Der Rendering-Pfad setzt sich aus den für die Anwendung nötige Ressourcen zusammen. Webanwendungen bestehen meist aus mehreren ausgelagerten Javascript und CSS Dateien die benötigt werden um überhaupt mit dem Rendern zu beginnen. Ein Seitenaufruf erzeugt folgenden Prozess:

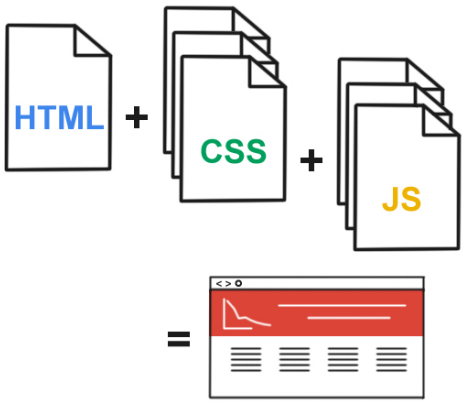

Abbildung 10: Ressourcen die für das Rendern von nöten sind

1. Nachdem die Adresse per DNS-Lookup (siehe Punkt: **??**) in eine IP Adresse übersetzt wurde, wird eine TCP Verbindung hergestellt und der Browser beginnt damit die HTML Datei vom Server zu laden.

[6] Suppressing 300ms delay for touchscreen interactions: `http://tinyurl.com/psj5nxz`

2. Der Browser liest das HTML von oben nach unten und sieht nach, ob zusätzliche Ressourcen benötigt werden (Bilder, Javascript oder CSS).
3. Diese Ressourcen werden beim Server angefragt.
4. Der Browser kann mit dem Rendern der Seite noch nicht beginnen, da er noch auf den Abschluss des Downloads wartet.
5. Der Browser liest das Javascript und sieht nach, welche CSS Selektoren auf das HTML Dokument passen.
6. Der Browser stellt fest, dass die Seite nun Angezeigt werden kann und beginnt mit dem Rendern.

Abbildung 11: Der Render Prozess der im Browser abläuft (Eigene Abbildung)

Wie im Prozessablauf abgebildet, ist ein frühes Rendering oft nicht möglich, weil Javascript und CSS das Rendern dadurch Blockieren, da sie zuerst Herunterladen werden und dann Übersetzt werden müssen, bevor der Browser entscheiden kann, dass die Seite gerendert werden kann. Kritischer Rendering-Pfad bedeutet, dass dem Browser die Daten zuerst zur Verfügung gestellt werden, die nötig sind um den „above the fold“ Inhalt zu rendern. Alle anderen Ressourcen sollten erst nach dem Laden der Seite Heruntergeladen werden um das Rendern nicht zu verzögern.

7 Entwicklung

7.1 Tools

7.1.1 Google Chrome Developer Tool

Stalled / Waiting DNS Lookup TTFB Start render Document Complete Document Loaded Audits Timeline

7.1.2 Webpagetest

Was ist Webpagetest Speedindex Features

7.1.3 Google Spreadsheet

7.2 Ausgangspunkt

7.3 Prozess der Suche

7.4 Prozess der Validierung

8 Best-Practices

8.1 Serverseitig

8.1.1 Verringern von DNS Lookups

8.1.2 HTTP Requests

pro / contra von sprites und großen concatenated files

8.1.3 Caching

pro / contra / problem bei updates

8.1.4 Pagespeed Mod

8.1.5 Images

5. Most sites fail to leverage best practices for optimizing images. Despite the fact that images represent one of the single greatest performance challenges (and opportunities), 34% of pages failed to properly implement image compression, and 76% failed to take advantage of progressive image rendering. (http://www.radware.com/assets/0/314/6442478110/c810eee1-e86f-438a-b82f-3ad002bf1c75.pdf)

Literatur

[Gir13] Ilya Girgorik. *High Performance Browser Networking.* http://tinyurl.com/lz8t3mh [Aufgerufen am 02.03.2015]. Chapter 10 Speed, Performance, and Human Perception. 2013 (siehe S. 9).

[Goo10] Google. *Using site speed in web search ranking.* Website. 2010 (siehe S. 2).

[Goo11] Google. *Creating Fast Buttons for Mobile Web Applications.* https://developers.google.com/mobile/articles/fast_buttons [Aufgerufen am 03.03.2015]. 2011 (siehe S. 9).

[Gri13a] Ilya Grigorik. *High Performance Browser Networking.* http://tinyurl.com/p5dds9p [Aufgerufem am 02.03.2015. Chapter 2 Slow-Start. 2013 (siehe S. 5, 6).

[Gri13b] Ilya Grigorik. *High Performance Browser Networking.* http://tinyurl.com/nojaxxa [Aufgerufen am 02.03.2015]. Chapter 7 Table 7.1. 2013 (siehe S. 10).

[Hol10] Urs Holzle. *Velocity 2010: Urs Holzle.* http://tinyurl.com/px7m64m [Aufgerufen am 27.02.2015]. Video from Velocity Conference. 2010 (siehe S. 2).

[ItW15] ItWissen.info. *Shared Hosting.* http://www.itwissen.info/definition/lexikon/Shared-Hosting-shared-hosting.html [Aufgerufen am 26.02.2015]. 2015 (siehe S. 4).

[Rad13] Radware. *Radware Mobile Infographic.* Website. http://blog.radware.com/wp-content/uploads/2013/11/Radware_SOTU_Fall_2013_Mobile_Infographic_Final1.jpg [Aufgerufen am 15.01.2015]. 2013 (siehe S. 2).

[Rad14] Radware. *STATE OF THE UNION - Ecommerce Page Speed and Web Performance.* http://www.radware.com/assets/0/314/6442478110/c810eee1-e86f-438a-b82f-3ad002bf1c75.pdf [Aufgerufen am 27.02.2015]. Seite 8. 2014 (siehe S. 8).

[Sch15] Amy Schade. *The Fold Manifesto: Why the Page Fold Still Matters.* Techn. Ber. http://www.nngroup.com/articles/page-fold-manifesto/ [Aufgerufen am 26.02.2015]. Nielsen Normand Group, 2015 (siehe S. 7).

[Ste06] Guido Stepken. *Anti-Pattern in der Softwareentwicklung.* http://www.little-idiot.de/teambuilding/AntiPatternSoftwareentwicklung.pdf [Aufgerufen am 26.02.2015]. 2006 (siehe S. 4).

[Ste08] Stoyan Stefanov. *Exceptional Website Performance with YSlow 2.0.* http://de.slideshare.net/stoyan/yslow-20-presentation [Aufgerufen am 27.02.2015]. Slide Nummer 4. 2008 (siehe S. 7).

[Ste11] Stoyan Stefanov. *Book of Speed.* http://www.bookofspeed.com/chapter3.html [Aufgerufen am 02.03.2015]. siehe Abbildung 3.4 - The-three-way handshake. 2011 (siehe S. 5).

[t3n15] t3n. *Ist deine Website „mobile-friendly“? – Google steigert Druck auf Webmaster.* http://t3n.de/news/google-mobile-friendly-589402/ [Aufgerufen am 25.02.2015]. 2015 (siehe S. 2).

[Tak13] Dean Takahashi. *Apple's iPhone 5 touchscreen is 2.5 times faster than Android devices.* http://venturebeat.com/2013/09/19/apples-iphone-5-touchscreen-is-2-5-times-faster-than-android-devices/ [Aufgerufen am 03.03.2015]. Grafik. 2013 (siehe S. 9).

[TNS14] Google TNS Infratest BVDW. *Global Connected Consumer Study*. Website. `http://www.netzproduzenten.de/wp-content/uploads/2014/08/global-connected-consumer-studie-deutschland.pdf` [Aufgerufen am 14.12.2014]. 2014 (siehe S. 2, 3).

[Web08] WebSiteOptimization.com. *The Psychology of Web Performance*. `http://www.websiteoptimization.com/speed/tweak/psychology-web-performance/` [Aufgerufen am 25.02.2015]. 2008 (siehe S. 2).

[wik14] wiki.ubuntuusers. *Der Benutzer root*. `http://wiki.ubuntuusers.de/sudo` [Aufgerufen am 26.02.2015]. 2014 (siehe S. 4).

[Wik15] Wikpedia. *Entwurfsmuster*. `http://de.wikipedia.org/wiki/Entwurfsmuster` [Aufgerufen am 26.02.2015]. 2015 (siehe S. 4).