

Exposé zur Bachelorarbeit

Webperformance für den mobilen Endanwender

Autor:

Andreas Lorer | 22652
andreas.lorer@hs-weingarten.de
88250 Weingarten
Wilhelmstraße 4

Inhaltsverzeichnis

1 Einleitung

Webperformance ist oft ein stark vernachlässigter Teil der Web- und Anwendungsentwicklung. Viele Anwender empfinden das Internet als sehr langsam. Sucht man bei der Suchmaschine

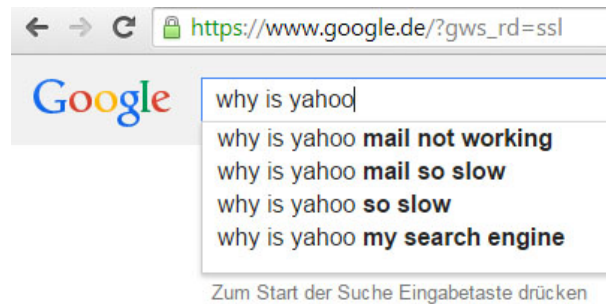


Abbildung 1: Why is Yahoo so slow?

Google nach dem Begriff: „Why is Yahoo so...“ bekommt man unter den Top vier der Suchanfragen den Vorschlag „slow“. Dies trifft auch auf Seiten wie Ebay, Youtube oder Reddit zu. Nicht nur kleine Webseiten, sondern auch große Plattformen haben ein Problem mit Webperformance oder werden zumindest von deren Nutzern so Wahrgenommen. Warum wird also das Internet generell als langsam empfunden?

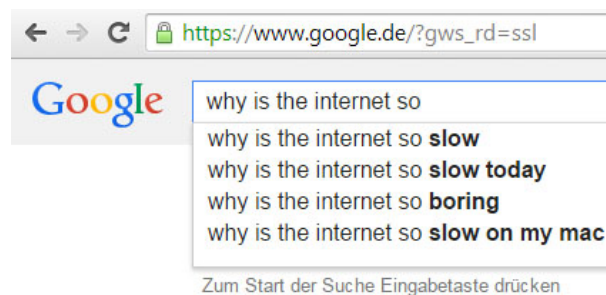


Abbildung 2: Why is the Internet so slow?

Das hat sehr viele Gründe. So ist die Geschwindigkeit einer Webseite auf den ersten Blick nicht sichtbar und Kunden zahlen vor allem gerne für Dinge, die für sie greifbar sind. Eine auf Geschwindigkeit optimierte Seite stellt zudem auch einen Mehraufwand dar, den nicht jeder Kunde bezahlen möchte oder es wird von ihm als nicht wichtig angesehen.

Auch die Infrastruktur spielt eine entscheidende Rolle. So ist ein langsamer Seitenaufbau vorrangig von der Latenz abhängig, die zwischen dem Anwender und dem Servers besteht. Zahlen, Fakten und Entwicklungstrends bestätigen, dass Webperformance einen sehr wichtigen Aspekt darstellt und vor allem in der Zukunft eine immer wichtigere Rolle spielen wird.

2 Motivation

Webperformance ist und wird immer wichtiger. Die Nutzung von Mobilien Endgeräten ist in den letzten Jahren stark gewachsen. Zwischen 2011 und 2014 stieg die Anzahl der Smartphone Nutzer von 18% auf 50% an. Dies ist ein Wachstum von 32% innerhalb von 3 Jahren.(**tns14**) Das

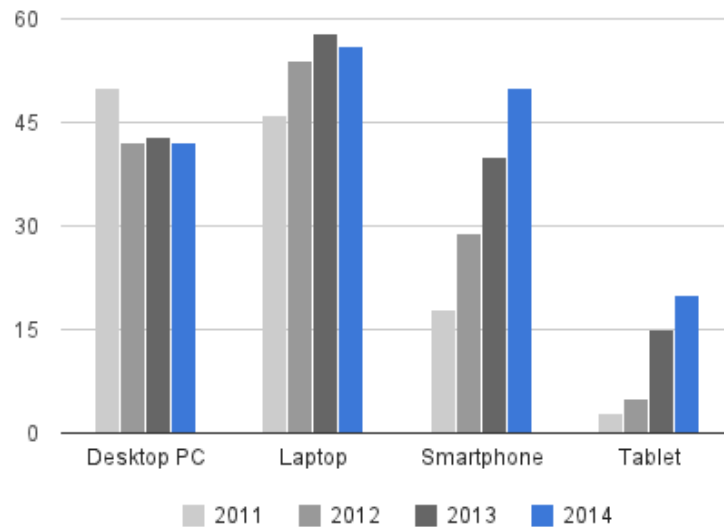


Abbildung 3: Gerätenutzung in der Gesamtbevölkerung (2011 – 2014)(**tns14**)

bedeutet, dass immer mehr Nutzer auf Webseiten zugreifen, die entweder erst gar nicht für den Mobilanwender optimiert sind (kein Responsive-Webdesign) oder die zwar Responsive sind, aber deren Ladezeit und Ladelast so hoch sind, dass die Nutzbarkeit der Seite stark eingeschränkt ist. Vor allem im E-Commerce ist Geschwindigkeit ein entscheidender Faktor, ob ein Besucher die Seite bei zu langem Laden verlässt und im schlimmsten Fall auf ein Angebot der Konkurrenz verwendet. Eine Untersuchung von Radware.com hat ergeben:

„Two out of three mobile shoppers expect pages to load in 4 seconds or less.“(**radware13**)

Dabei haben im Schnitt Webseiten die nicht für Mobile optimiert sind eine durchschnittliche Ladezeit von 7.84 Sekunden und Webseiten die für Mobile optimiert sind fallen mit 4.33 Sekunden auch über die 4 Sekunden Marke.(**radware13**)

Auch Google findet, dass Webperformance wichtig ist und den Anwender glücklicher und zufriedener macht und veröffentlichte bereits 2010, dass Webperformance in das Suchranking 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“(**google10**)

Es gibt noch weitere Gründe, warum es wichtig ist, auf Geschwindigkeit als Feature zu setzen. Dies soll in der Bachelorarbeit tiefer aufgeführt werden.

3 Stand der Forschung

Es gibt momentan eine große Anzahl an Maßnahmen die ergriffen werden können, um eine Webseite auf ihre Performance zu optimieren. Sowohl Clientseitig als auch für den Server gibt es viele „Best-Practices“, Tools oder Plugins, die die Performance erhöhen. Viele der „Best-Practices“ stellen sich aber momentan als „Workaround“ für das 20 Jahre alte HTTP/1.1 Protokoll heraus. In der Zukunft wird dieses Protokoll von HTTP/2.0 abgelöst werden, was viele Verbesserungen mit sich bringen wird die momentan noch sehr umständlich manuell gelöst werden müssen.

4 Zielsetzung

Die Bachelorarbeit soll einen Überblick über die heute gängigen Methoden zur Verbesserung der Webperformance schaffen. Es sollen „Best-Practices“ gefunden und vorgestellt werden, Pattern und Anti-Pattern diskutiert und im Praktischen Teil der Arbeit soll eine Webseite mit einer Bildergalerie entstehen, die in möglichst kurzer Zeit (< 2 Sekunden) auf dem Smartphone lädt.

5 Bedeutung der Arbeit

Die Arbeit soll untersuchen, welche Auswirkungen die verschiedenen Methoden auf die Ladezeit einer Seite haben. Es soll herausgearbeitet werden, wie ein optimaler „Workflow“ aussehen könnte und wie dieser bereits bei Projektstart eingesetzt werden kann. Das soll ermöglichen Projekte zu verwirklichen die bereits bei der Entwicklung performant sind und nicht erst am Schluss an der Performance nachgearbeitet werden muss.

6 Gliederung

6.1 Abstract

6.2 Projektbeschreibung

- Problem beschreiben: Warum ist das Internet vor allem auf Mobile Devices langsam?
- Motivation: Google rankt Performante Seiten höher. First visitors bleiben auf der Seite ect...
- Features
- Zielgruppe
- Fähigkeiten der Nutzer?
- Eingrenzung des Themas: Was wird behandelt und was nicht?
- Zielsetzung: Schnell: "First-Render" nach unter 2 Sekunde von einem Deutschen Server (Messbarkeit). Best Practices, Methoden, Schranken / Grenzen
- „Ist-Zustand:“ Webperformance wird nicht als sehr wichtiges Feature angesehen. Punkte dafür finden was dafür Spricht. Negative Aspekte aufgreifen und vorweg nehmen. Argumente für Webperformance aufführen.
- „Soll-Zustand:“ Workflow und Best-Practices die bereits bei Projekt beginn eingesetzt werden können.

6.3 Grundlagen

- Begriffe (Above the Fold, Critical Render Path, Payload, TTFB, RTT)
- Tools (pro / contra)
- Latenz & Netze
- Http/1.1
- Datenvolumen / Datenrate (Steigende anz. an Bildern)

6.3.1 Prozess der Suche

- Auf welche Frage suche ich eine Antwort? -> was gibt es momentan für Methoden und welche davon sind für mich einsetzbar?
- Was habe ich gefunden? Was ist daran gut / schlecht und was gibt es für alternativen?
- Warum verwende ich XY und nicht AB?

6.4 Entwicklung

- Was hat die Abteilung bereits Entwickelt?
- Baut es auf etwas Bestehendem auf?
- Was soll am Ende des Projekts rauskommen?
- Vorgehensweise
- Entwicklungsverlauf (Messungen, Daten, Analyse, Folgerungen)

6.5 Prozess der Validierung

- Wie überprüfe ich den Erfolg des Projektes? Diagramme? Flow-Charts? Zahlen, Fakten ect...

6.6 Zusammenfassung

Erkenntnisse (best-practices): Auflistung aller Performance möglichkeiten unterteilt in Server / Client / Workflow / Programming / Tools.

6.6.1 Workflow & Programming

- Performance durch performanten Code
- Dom Manipulation bei onload vermeiden
- Loops vermeiden
- Concatenating von files (Achtung: load and execute order!!)
- Uglyfy / Minify code
- Libraries / Frameworks wirklich benötigt

- Bootstrap customize
- (optional) Was für Programmcode braucht mehr Leistung als anderer bei dem selben Ergebnis? (Stichwort: Eventlistener, Event-Delegation, Event-Bubbling)

6.6.2 Server

- DNS lookups
- Caching von statischen elementen (beachte: how to override cached files for clients)
- Server side rendering vs client side (siehe zitate: twitter)
- ModPageSpeed Apache / Nginx Plugin

6.6.3 Client

- Critical Render path,
- Render blocking javascript,
- above the fold
- Lazy loading
- Schlaue request? / Ajax loading
- inline css and scripts vs External scripts
- Image format webp / jpg pros (30% kleiner) and cons (nicht supported)
- (`<img srcset="..."`)
- Image optimierung (save as./ save for web), Progressive Images, Scaling usw.
- Browser caching, allgemeines caching

6.6.4 Best Practices

- Einschränkungen (shared hosts / kein root-Zugang ect.)
- Patterns und Anti-Pattern
- Methoden
- Ausblick (HTTP/2.0, responsive img-Tag)

6.7 Fazit

- Was ist in der Praxis durchsetzbar und was nicht?
- Was ist in diesem Projekt vielleicht möglich aber sonst generell nicht? Wo sind die Schranken?
- Was sind meine Erkenntnisse? Neue?
- Was hat sich zum IstZustand geändert und was nicht?
- Ist das Projekt abgeschlossen? Was ist der Stand?

7 Vorgehen

Eine bereits in ihren Grundgerüst bestehende Bildergallery soll als „Dummy-Projekt“ benutzt werden um zu testen, anzuwenden und messbar zu machen. Die daraus resultierenden Erkenntnisse sollen in der Arbeit ausgewertet und bewertet werden.

8 Herausforderungen

Eine Herausforderung wird sein, das Projekt auf einem Server aufzusetzen auf dem ich mit **Root-Rechten** zugreifen kann und diesen mit den entsprechenden Plugins und Einstellungen zu konfigurieren. Hier kommt entweder der Cloudcomputing-Service von Amazon (AWS - Amazon Web Services) oder Microsoft (Azure) in Frage. Vor allem meine geringen Kenntnisse in diesem Bereich macht es sehr schwer Abschätzbar wieviel Aufwand und Einarbeitungszeit hier benötigt werden.

Es wird Schwierigkeiten geben die angewendeten Methoden zur Performance Steigerung Messbar zu machen und darzustellen.

Zudem wird es eine Herausforderung sein, die Webseite so zu verbessern, dass sie in weniger als 2 Sekunden Ladet.

9 Zeitplan

1. Meilenstein: Setup

- Outline der Arbeit erstellen
- Anmeldung der Arbeit
- Aufsetzen des Projekts Lokal und auf einem der unter ?? aufgelisteten Webservices.
- Automatisierte Messung der Ladezeiten einer Webseite einrichten (webpagetest API)

2. Meilenstein: Performance Tuning

- Recherche nach Performance Methoden
- Implementierung der gängigen Practices
- Performance Tuning der bestehenden Bildergallery

3. Meilenstein: Auswertung der Messdaten

- Archivierung
- Verbildlichung
- Strukturierung des Materials für den Schriftlichen Teil
- Treffen mit Betreuer

4. Meilenstein: Schreiben der Arbeit

- Quellen zu den einzelnen Punkten strukturieren
- Reinschrift

- Korrekturen
- Treffen mit Betreuer
- Korrekturen
- Abgabe der Arbeit