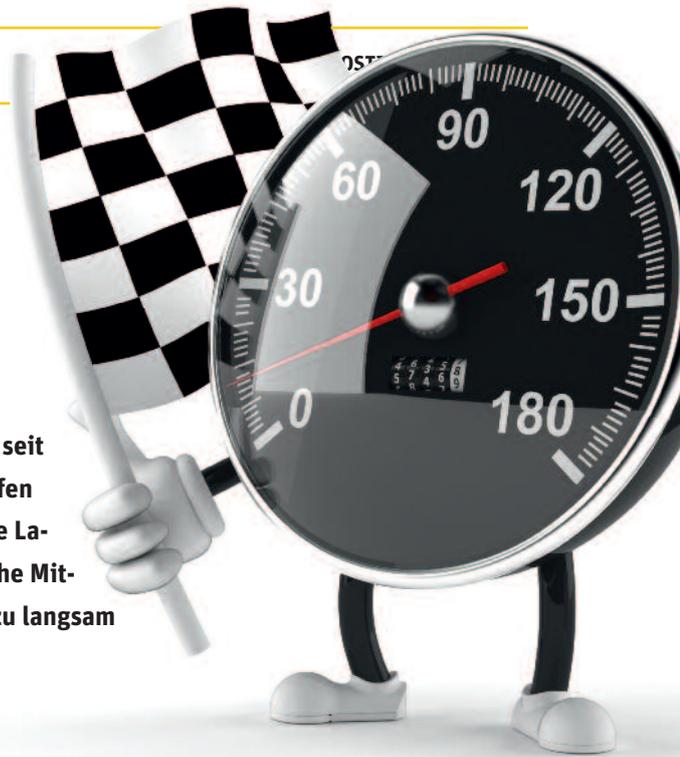


Daniel Unger & Daniel Friedrich

»SPEEDBOOSTING

Schnelle Ladezeiten einer Website sind mittlerweile weit mehr als Nice-to-have. Für Google ist die Ladegeschwindigkeit einer Website seit gewisser Zeit sogar ein Rankingfaktor. Bereits mit einfachen Eingriffen an der Website kann man einiges an Bandbreite einsparen und so die Ladegeschwindigkeit verbessern. Im folgenden Artikel werden mögliche Mittel und Wege vorgestellt, wie man erkennen kann, ob eine Website zu langsam ist, und wie man dieses Problem löst.



Eine schnelle Website für Google und die User

Im April 2010 wurde von Google offiziell bekannt gegeben, dass die Ladegeschwindigkeit einer Website bei der Bewertung der Suchmaschinenrankings eine Rolle spielt. Google gab zwar damals an, dass weniger als 1 % der Suchanfragen betroffen seien, jedoch ist dies schon über ein Jahr her. In der seitdem vergangenen Zeit wird Google diesen Rankingfaktor sicherlich weiter verfeinert haben und man kann davon ausgehen, dass die Ladegeschwindigkeit mittlerweile eine etwas größere Rolle für gute Rankings spielt. Im Verhältnis zu anderen Rankingfaktoren wie z. B. Backlinks wird diese Rolle jedoch weiterhin sehr gering sein. Vor allem bei stark umkämpften Keywords wird die Ladegeschwindigkeit einer Website bei der Rankingbewertung berücksichtigt. Bei suchvolumenstarken Keywords wie z. B. „Private Krankenversicherung“ entscheiden oft nur Nuancen, auf welcher Position eine Website bei Google gefunden wird. Genau für solche Fälle hat Google mit dem Website-Speed eine weitere Bewertungsmöglichkeit.

Kurz nach der Veröffentlichung des neuen Rankingfaktors nahm damals sogar der oberste Google-Geheimniswächter Matt Cutts zu dem Thema Website-Speed Stellung. In einem Blogartikel ging er verstärkt darauf ein, dass dieses neue Rankingkriterium erst einmal nichts Weltbewegendes sein werde. Vielmehr versuchte er zu erklären, dass Website-Speed im Interesse aller sei und Google deshalb hier als Vorreiter agiere. Wie wichtig das Thema Ladegeschwin-

digkeit bei Google wirklich ist, sieht man nicht zuletzt daran, dass immer neue Tools und Features von Google zur Verfügung gestellt werden, die beim Speedboosten einer Website helfen. Hierzu zählt unter anderem auch der im August vorgestellte Page-Speed-Service von Google, der Websites bis zu 60 % schneller machen soll.

Es geht um den User!

Eine schnelle Website allein ist jedoch weiterhin bei Weitem nicht der heilige Gral für Top-Rankings. Der Hauptgrund, warum man seine Website überhaupt beschleunigen sollte, ist nämlich nicht unbedingt der zu erwartende Vorteil im Ranking. Vielmehr sollten die Ladezeiten für die eigenen Website-Besucher geboostet werden. Eine langsame Website macht niemandem Spaß und führt zwangsläufig dazu, dass die Besucher unzufrieden werden und eventuell die Website oder den Onlineshop in Zukunft meiden. Bei Onlineshops haben verschiedene Tests auch schon gezeigt, dass die Konversionsraten bei schneller ladenden Seiten tendenziell sogar höher sind. Auch dieser Grund spricht für eine geschwindigkeitsoptimierte Website und sollte deshalb nicht außer Acht gelassen werden.

Google spricht bezüglich der allgemeinen Zufriedenheit der User auch immer wieder von der sogenannten User Experience. Zu diesem Nutzererlebnis gehören das Look & Feel einer Website und vor allem die Usability, also die Benutzerfreundlichkeit.

Diesen Leitsatz verfolgt Google selbst, indem die am besten passenden Suchergeb-

DER AUTOR



Daniel Unger ist Geschäftsführer und Gründer

der eology GmbH. Mit seinem aktuell 14-köpfigen Team betreut er Kunden aus ganz Deutschland in den Bereichen SEO, PPC, Usability und Social Media.

DER AUTOR



Daniel Friedrich ist Online-Marketing-Consultant

bei der eology GmbH und vor allem für die Entwicklung firmeninterner Tools zuständig. Weiterhin betreut er einige Projekte im Linkaufbau.

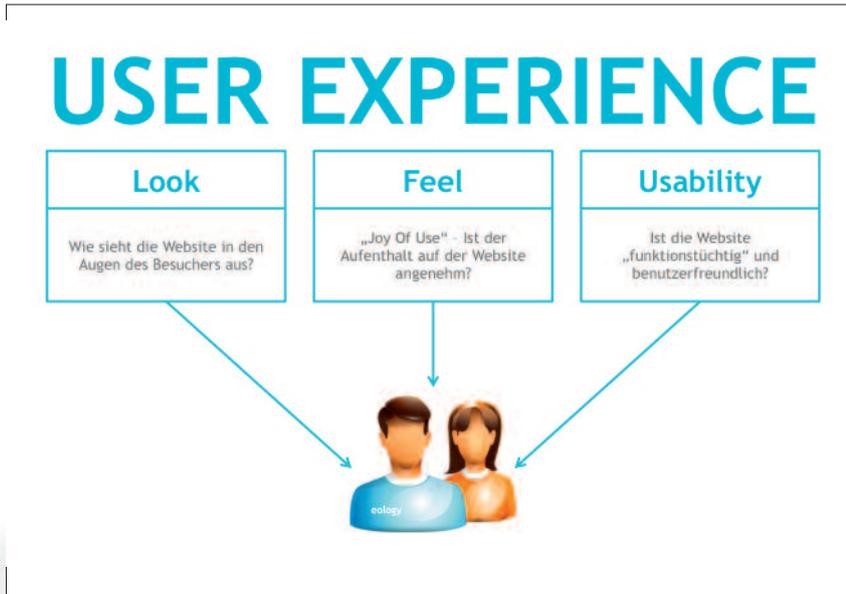


Abbildung 1: Die User Experience (Quelle: eigene Erstellung eology)

„Über acht Millionen Deutsche surfen aktuell noch mit einer Schmalband-Internetverbindung!“

Jahr	Breitband	Schmalband	Sonstiges	keine Onliner
2010	49,6 %	16,3 %	6,2 %	28,0 %
2011	52,5 %	15,9 %	6,2 %	25,3 %

Abbildung 2: Internetnutzung in Deutschland (Quelle: Zahlen aus (N)Onliner Atlas; www.nonliner-atlas.de)

nisse zu einem Begriff möglichst schnell und einfach angezeigt werden sollen. Um also den Usern das bestmögliche Nutzungserlebnis zu bieten, muss eine Website unter anderem auch eine angemessene, schnelle Ladezeit besitzen.

Einen weiteren Grund, warum man seine Website beschleunigen sollte, zeigt Deutschlands größte Studie zur Internetnutzung, der sog. (N)Onliner Atlas 2011. Diese jährliche Studie wurde von der Initiative D21 unter anderem mit dem Bundesministerium für Wirtschaft und Technologie durchgeführt. Laut dieser Untersuchung sind in Deutschland aktuell 52,7 Millionen

Menschen über 14 Jahren online. Ganze 15,9 % sind jedoch auch heute noch mit Schmalband, also ISDN oder Modem, unterwegs. Dies bedeutet, dass momentan noch über acht Millionen potenzielle Kunden mit einer sehr langsamen Leitung im Netz surfen. Nicht zu vergessen ist auch die stetig steigende Anzahl an Nutzern des mobilen Internets. Bekannterweise lässt auch hier häufig die Geschwindigkeit zu wünschen übrig. Mit der Geschwindigkeits-Optimierung der Website kann man auch für diese Nutzer ein angenehmes Nutzungserlebnis schaffen und eventuell dadurch sogar Neukunden gewinnen.

Möglichkeiten zum Herausfinden des Website-Speeds

Google Analytics

Im Mai dieses Jahres hat Google eine sehr nützliche Funktion für Google Analytics veröffentlicht, mit deren Hilfe man die Website-Geschwindigkeit der eigenen Site messen kann. Um dieses neue Feature nutzen zu können, muss man lediglich den Google Analytics Code um eine Zeile erweitern und die Funktion „trackPageLoadTime“ einfügen.

Bei richtiger Implementierung des Codes erscheint nach spätestens 24 Stunden im Analytics-Account die Funktion Website-Geschwindigkeit. Der Google Analytics-Account muss jedoch vorher auf die neue Benutzeroberfläche umgestellt werden. Wenn dies noch nicht geschehen ist, findet man im rechten oberen Bereich den Hinweis „Neue

```
<script type="text/javascript">
var _gaq = _gaq || [];
_gaq.push(['_setAccount', 'UA-XXXX-X']);
_gaq.push(['_trackPageview']);
_gaq.push(['_trackPageLoadTime']);

(function() {
var ga = document.createElement('script'); ga.type = 'text/javascript'; ga.async = true;
ga.src = ('https:' == document.location.protocol ? 'https://ssl' : 'http://www') + '.google-analytics.com/ga.js';
var s = document.getElementsByTagName('script')[0]; s.parentNode.insertBefore(ga, s);
})();
</script>
```

Abbildung 3: Modifizierter Google Analytics Code

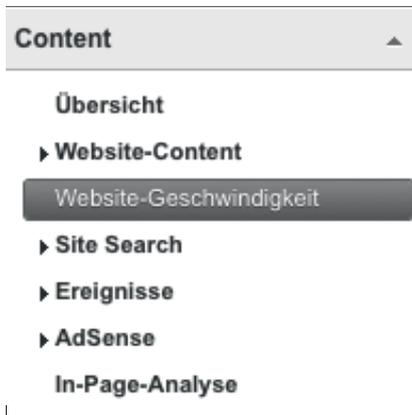


Abbildung 4: Google Analytics Website-Geschwindigkeit

Version“. Sobald die Benutzeroberfläche umgestellt ist, erscheint unter dem Punkt „Content“ der Menüpunkt „Website-Geschwindigkeit“.

Diese Trackinglösung hat den Vorteil, dass die Geschwindigkeit direkt beim Nutzer gemessen wird. Somit werden alle Verbindungsgeschwindigkeiten bei der Messung berücksichtigt, mit denen die User auf der Website unterwegs sind. Als Ladezeit zählt dabei der komplette Aufbau der Seite. Es müssen also alle Elemente der Seite vollständig geladen werden, erst dann wird die Ladezeit gestoppt. Google misst jedoch nicht bei jedem Aufruf die Geschwindigkeit. Es werden nach dem Stichproben-Prinzip ungefähr 2 % der gesamten Seitenaufrufe von Google gemessen.

Wie auf Abbildung 5 zu sehen, werden die Ladezeiten für jede URL einzeln angezeigt. Dies ist sehr praktisch, da so die Diagnose vereinfacht wird. Man hat somit die Möglichkeit, Unterseiten zu identifizieren, welche Performance-Probleme haben, und diese gezielt zu optimieren. Weiterhin werden auch noch die Absprungraten und die prozentualen Ausstiege jeder URL angezeigt. Dadurch hat man einen Überblick und kann einschätzen, ob eine hohe Absprungrate eventuell direkt mit einer zu langsam ladenden Seite zusammenhängt. Für die grafische Darstellung wird täglich ein Mittelwert gebildet und im Verlauf ange-



Abbildung 5: Ausschnitt aus Google Analytics

zeigt. Was leider fehlt, ist die Betrachtung eines einzelnen Tages mit Uhrzeiten. Mit solch einer Darstellung könnte man analysieren, ob der Server bei hohen Zugriffszahlen Probleme mit der Performance bekommt. Wer solch detaillierte Auswertungen benötigt, muss auf externe Performance-Monitoring-Tools wie z. B. Gomez (www.gomez.com) zurückgreifen.

Google Webmaster-Tools

Mit den Webmaster-Tools stellt Google ein weiteres Tool zur Verfügung, mit dem man die Ladezeit seiner Website herausfinden kann. Die Webmaster-Tools können jedoch noch viel mehr und aus diesem Grund sollte sie eigentlich jeder Website-Betreiber „installiert“ haben. Mit den Tools kann man sehr viele nützliche Informationen wie z. B. Crawling-Statistiken über seine Website herausfinden und Anweisungen an Google übergeben. Hierzu gehört z. B. das Einstellen der Google Sitelinks oder das Einreichen von XML-Sitemaps.

Um die Google Webmaster-Tools nutzen zu können, muss man sich auf www.google.de/webmasters/tools einen Account einrichten. Bevor die Webmaster-Tools jedoch genutzt werden können, muss die Website authentifiziert werden. Dies geschieht z. B. über den Upload einer von Google bereitgestellten HTML-Datei in dem Webpace. Die

zweite Möglichkeit ist das Einbinden eines speziellen Codeschnipsels namens „google-site-verification“ in den Meta-Angaben der Website. Somit weiß Google, dass man Zugriff auf die Website hat, und nach erfolgreicher Überprüfung, wird der Zugriff auf die Webmaster-Tools gewährt.

Unter dem Punkt Google Labs befindet sich die Funktion Website-Leistung. Nach dem Aufruf der Funktion findet man eine grafische Darstellung, die im zeitlichen Verlauf die Ladegeschwindigkeit der Website aufzeigt.

Die im Beispiel (Abbildung 6) gezeigte Website lädt im Schnitt 2,0 Sekunden. Nach Angaben von Google lädt diese Seite also schneller als 69 % aller Websites im Netz. Obwohl dies eigentlich nicht einmal so schlecht ist, wird diese Seite trotzdem noch als „langsam“ eingestuft. Für Google ist eine Seite nämlich erst schnell genug, wenn diese sich unter 1,5 Sekunden aufbaut. Dies erreichen laut Aussagen von Google gerade einmal 20 %. Auch daran erkennt man, wie ernst es Google mit dem Thema Site-Speed nimmt.

Die Daten für diese Auswertung erhebt Google über die hauseigene Toolbar und genau hier liegt auch schon das Problem. Google hat bei dieser Messung nur Daten von Usern, welche die Google Toolbar installiert haben. Und wer hat das schon? Hauptsächlich wohl interne-



Abbildung 6: Website-Leistung Google Webmaster-Tools

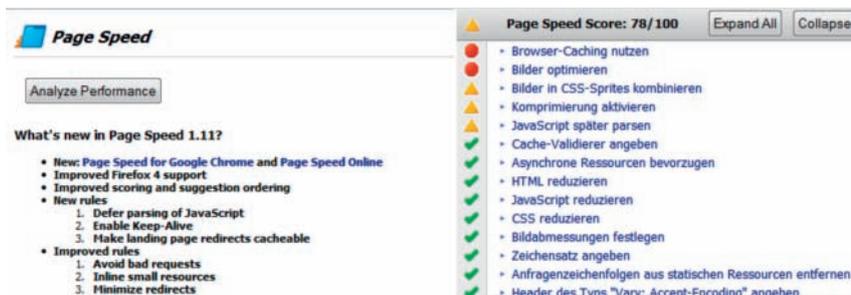


Abbildung 7: Google Page Speed zeigt an, was auf der Website nicht passt

taffine Nutzer. Somit ist die Stichprobe wahrscheinlich nicht besonders repräsentativ.

Aus diesem Grund weichen auch die gemessenen Zeiten von Analytics und die aus Webmaster-Tools zum Teil massiv voneinander ab. Google gibt jedoch auch in den Webmaster-Tools an, wie genau die vorliegenden Daten sind. In dem Beispiel liegen Google mehr als 1.000 Datenpunkte vor. Deshalb haben die Schätzungen eine hohe Genauigkeit. Bei kleineren Websites, welche nicht so viel Traffic bekommen, können die Daten unter Umständen nicht so genau bestimmt werden.

Weiterhin werden unter der Grafik Beispiel-Ladezeiten einzelner URLs angezeigt, wie bei Google Analytics. Es werden hier jedoch nur ein paar Beispiele aufgelistet, einen kompletten Überblick aller URLs bekommt man über die Google Webmaster-Tools nicht. Für

einen schnellen und groben Überblick reicht die Funktion Website-Leistung der Google Webmaster-Tools auf alle Fälle aus. Wenn man jedoch genauere und umfassendere Daten benötigt, sollte man die Google Analytics-Methode nutzen.

Google Page Speed

Eine weitere Möglichkeit, die Geschwindigkeit einer Webseite zu überprüfen bzw. einzustufen, ist das von Google zur Verfügung gestellte Page Speed. Was vor zwei Jahren als Browser-Extension für Firefox bzw. Firebug begann, hat seitdem eine gehörige Weiterentwicklung erfahren. So wurde wenig später sogar das Apache-Modul mod_pagespeed entwickelt, welches als Apache-Erweiterung eingesetzt werden kann. Mithilfe dieser Erweiterung sollen Seiten schon optimiert ausgegeben werden. Mittlerweile kann man Google Page

Speed auch direkt online unter page-speed.googlelabs.com nutzen und als Entwickler sogar die API dazu verwenden, um die Ausgaben des Tools weiterzuverarbeiten. Was aber macht Page Speed genau?

Kurz gesagt überprüft das Tool die Webseite hinsichtlich einiger vorher definierter Regeln, die Google als wichtig erachtet. Darunter fallen Klassiker wie:

- » Sind Bilder optimiert?
- » Ist JavaScript komprimiert?
- » Wird Browser-Caching genutzt?

Das ist jedoch lediglich eine kleine Auswahl an Faktoren, welche von Page Speed überprüft werden. Bewertet wird auf einer Skala von 0–100 Punkten, wobei 100 Punkte erwartungsgemäß den Idealwert darstellen.

Um einen Test zu starten, benötigt man im Firefox lediglich die Firebug-Extension und eben die Erweiterung Page Speed (code.google.com/intl/de-DE/speed/page-speed/download.html). Man besucht die Website, welche man analysieren möchte, und startet das Tool.

Bereits wenige Sekunden nach dem Start des Tools erhält man eine Liste mit möglichen Verbesserungsvorschlägen. Jedes dieser Elemente ist auch der Priorität nach geordnet. Man weiß sofort, welche Möglichkeiten priorisiert ange-



Abbildung 8: Mit Yahoo's Smush.it Bandbreite sparen

gangen werden sollten. Dies sind allerdings auch nur Vorschläge von Google. Wie viel schneller die Webseite am Ende wirklich ist, kann vorher natürlich nicht festgestellt werden. Möchte man diese Werte noch einmal vergleichen, kann man alternativ auch das von Yahoo! zur Verfügung gestellte Firebug-AddOn YSlow nutzen. Wie der Name schon verrät (Why slow? – Warum langsam?) zeigt dieses Tool ebenfalls Gründe für langsame Ladezeiten, ähnlich wie Googles Page Speed.

Die eben genannten Browser-AddOns und Analyse-Werkzeuge helfen dabei, langsame Ladezeiten überhaupt zu erkennen. Nachfolgend werden nun Optimierungsmöglichkeiten für den Site-Speed vorgestellt. Um hier einen guten Einstieg in die Ladezeiten-Optimierung zu haben, darf es an etwas grundlegender Theorie nicht fehlen.

Etwas Theorie vorab

Mit jedem Element, das eine Webseite lädt bzw. laden muss, wird ein sogenannter HTTP-Request gestartet. Nach den Spezifikationen des HTTP-Protokolls können zwar mehrere Dateien parallel von einem Hostnamen geladen werden, allerdings liegt die Anzahl der parallelen HTTP-Requests zwischen vier und sechs (siehe Quelle: www.browserscope.org/?category=network&v=top). Ein Flaschenhals ist quasi

vorprogrammiert. Sobald vier bzw. sechs Elemente geladen werden, müssen nachfolgende Elemente warten. Ein weiterer Faktor, der sich auf die Geschwindigkeit der Webseite auswirkt, ist die zu übertragende Datenmenge. Gerne werden hochaufgelöste Bilder genutzt oder mehrere JavaScripts oder Cascading Stylesheets (CSS) in den Browser geladen, welche zusätzliche Datenmengen darstellen. Benutzer von Content-Management- bzw. Blog-Systemen wie z. B. Wordpress oder TYPO3 haben aufgrund deren Komplexität zudem einen erhöhten Zugriff auf die Datenbank. Jeder Aufruf an die Datenbank kostet zusätzlich Zeit. Die hauptsächlichen Ziele der Optimierung kann man daher ganz klar definieren:

- » Reduzierung und Optimierung der zu ladenden Bilder
- » Reduzierung der Datenmenge
- » Reduzierung der HTTP-Requests
- » Optimieren von CSS und Javascript Dateien

Bilder bieten das größte Optimierungspotenzial

Die Dimensionen der Bilder einer Website sollten nicht nur durch die Width- und Height-Angaben im HTML-Quelltext skaliert werden. Sie sollten auch direkt in der Größe auf dem Server abgespeichert sein, die in diesen Attributen angegeben ist, denn wenn man

die Größenangabe der Bilder per Hand – also mit den Width- und Height-Attributen – ändert, wird das Bild erst im Browser gerendert. Speichert man das Bild direkt im passenden Format, kann es von den Browsern direkt ausgegeben werden. Dies spart mitunter nicht nur Zeit für das Rendern, sondern in den meisten Fällen auch Speicherplatz. Damit das Rendern des Browsers schneller vonstatten geht, sollte man generell die Größenattribute mit in die Bildangaben (img-Tag) setzen. So wird der Platz für das Bild quasi schon vorab „reserviert“ und muss nicht erst gesondert berechnet bzw. gerendert werden. Dies sieht man z. B., wenn man auf News-Seiten auf Bilder wartet, das Layout aber schon vorher fertig steht. Sind diese Attribute nicht gesetzt, sieht man hin und wieder, dass sich der Content entsprechend verschiebt bzw. positioniert.

Bilder komprimieren

Vor allem die Größe der Bilder spielt bei der Optimierung eine große Rolle. Die meisten Bildbearbeitungsprogramme wie z. B. Adobe Photoshop oder die Open-Source-Software Gimp speichern neben dem eigentlichen Bild noch jede Menge Meta-Daten mit ab. Die Bilder werden dadurch unnötig aufgeblasen. Hier kann man nur durch Entfernen dieser Meta-Daten noch einmal einiges an Bandbreite sparen. Für diese Aufgabe hat sich vor allem der Dienst smushit.com von Yahoo! etabliert. Dieser Dienst entfernt nicht benötigte Daten aus den Bildern. Dabei muss man beachten, dass das Bild an sich nicht verändert wird. Bei der Bildqualität müssen somit keine Abstriche gemacht werden, da lediglich sogenannte Header-Informationen aus den Bildern entfernt werden. Für einige Content-Management-Systeme wie etwa Wordpress gibt es bereits Plugins, die beim Hochladen von Bildern diese automatisch mithilfe von Smush.it optimieren. Aktuell

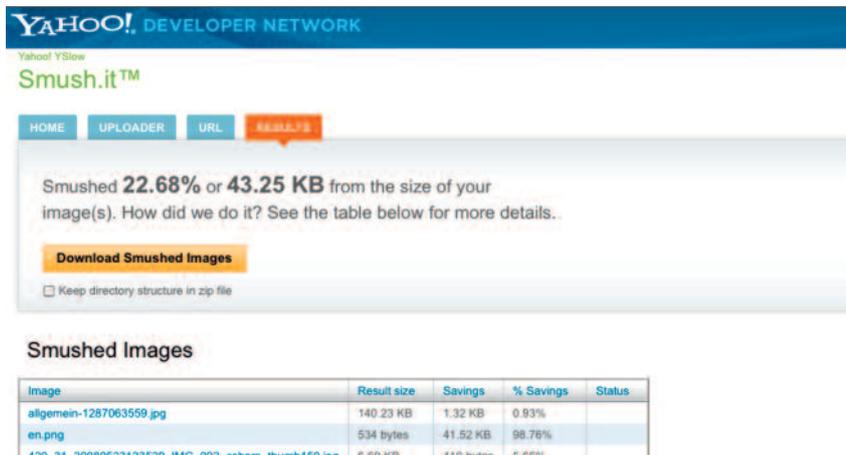


Abbildung 9: Bilder bergen mit das größte Einsparpotenzial

/css/reset.css	200	css
/css/jquery-ui-1.8.11.custom.css	200	css
/css/seite.css	200	css
/css/allgemein.css	200	css
/css/anlagen.css	200	css
/css/sprites.css	200	css
/css/prettyPhoto.css	200	css
/js/jquery-1.5.1.min.js	200	js
/js/jquery-ui-1.8.11.custom.min.js	200	js
/js/jquery.cycle.js	200	js
/js/jquery.validate.js	200	js
/js/jquery.prettyPhoto.js	200	js

Abbildung 10: Auswerten der Datenmenge mit Google Page Speed

gibt es aber noch keine offizielle API (Stand August 2011), daher ist man gezwungen, den Uploader der Webseite zu nutzen. Alternativ kann man auch ein URL-Set übergeben, um Zeit beim Hochladen zu sparen. Die von Shmush.it optimierten Bilder können dann anstelle der Originale verwendet werden und schon hat man wieder ein paar Sekundenbruchteile an Ladezeit eingespart. In dem in Abbildung 9 gezeigten Beispiel konnten bei einem Bild sogar fast 99 % an Speicherplatz eingespart werden. Dieses Bild ist eine Länderflagge für die Sprachumschaltung und ist so aktuell auf einer Website online. Von solch einem großen Einsparungspotenzial darf man jedoch nicht zwangsläufig ausgehen. Wenn allerdings über die Masse an Bildern pro Bild ca. 1–5 % eingespart werden können, ist das schon ein guter Wert. Vor allem für Onlineshops, die viele Produkte und somit Bilder haben, ergibt sich hier ein enormes Einsparungspotenzial. Generell kann man be-

haupten, dass Bilder häufig das größte Potenzial bieten, um Bandbreite und damit wichtige Zeit beim Laden einer Seite zu sparen.

Möglichkeiten mit CSS

Was auf den ersten Blick schwierig und nach viel Arbeit klingt, lässt sich je nach Projekt und genutztem Content-Management-System recht schnell und zum Teil ohne großen Aufwand umsetzen. Um sich noch einmal mehr zu verdeutlichen, wie viele Daten eigentlich bei einem einzigen Aufruf vom Server geladen werden, kann man Tools wie das schon vorgestellte Google Page Speed oder Yahoo! YSlow nutzen.

In dem gezeigten Beispiel werden aktuell sieben Stylesheets geladen. Damit diese sieben CSS-Dateien vom Server geladen werden können, sind insgesamt zwei HTTP-Requests nötig (ausgehend von sechs gleichzeitigen Requests von einem Host). Das Optimierungspotenzial besteht also darin, die

Dateien zu einem in der Datengröße zu verkleinern und zum anderen in so wenig Dateien wie möglich zusammenzufassen.

Optimieren von CSS-Dateien

Je moderner und technisch anspruchsvoller eine Webseite ist, desto mehr CSS-Anweisungen werden in der Regel benötigt. So hat man dann z. B. eigene Anweisungen für den Bild-Slider und noch mal eine eigene CSS-Klasse für die Lightbox, um die Bilder möglichst schön darzustellen. In den meisten Fällen haben sich die Programmierer einen einheitlichen Schreibstil für CSS angewöhnt, den man leicht lesen, erfassen und pflegen kann. Eine Klasse ist demnach meist nach folgendem Schema aufgebaut:

```
#id-name {
    Anweisung 1: Werte für Anweisung 1;
    Anweisung 2: Werte für Anweisung 2;
}
```

In dem Code einer CSS-Datei würden hier z. B. Ausrichtungs-Anweisungen wie border-left stehen. Egal, wie einfach die Anweisungen sind, gibt es auch hier schon Potenzial, um die Datenmenge insgesamt zu verringern. So kann man die kompletten Anweisungen in einer einzigen Zeile schreiben. Im oberen Beispiel könnten dadurch bereits drei Zeichen eingespart werden. Bei einer CSS-Datei mit mehreren Tausend Zeilen an Code kann man hier nur durch Entfernen unnötiger Leerzeilen und das Zusammenfassen der Anweisungen in einer einzelnen Zeile eine nicht unerhebliche Reduzierung der Datenmenge herausholen.

Um noch zusätzlichen Speicherplatz und damit entsprechend Bandbreite zu sparen, sollten die unnötigen und doppelt deklarierten Anweisungen im Code entfernt werden. Oft werden CSS-An-



Abbildung 11: CSS-Sprite von Amazon – nur noch eine große Grafik anstatt vieler kleiner

weisungen beibehalten, die es so auf der Webseite nicht mehr gibt. Dies kommt oft in Content-Management- oder auch Shop-Systemen vor, in denen es meist schon eine CSS-Datei gibt. Diese findet man z. B. in Wordpress in Form der stylesheet.css. Hier lohnt es sich wirklich nachzuforschen, ob evtl. ganze Blöcke erst gar nicht geladen werden. Diese kann man dann guten Gewissens aus dem Stylesheet entfernen. Nicht selten spart man sich hier mehrere Kilobyte an Speicher, die am Ende nicht übertragen werden müssen. Dabei muss man nicht mal mehr CSS-Dateien doppelt speichern, um diese später einfacher zu pflegen. Viele Entwicklungsumgebungen, wie etwa Coda auf Mac-Systemen, bieten nämlich bereits solch eine Funktion out-of-the-box an.

Zusätzlich zu der Datenmenge kann man bei den CSS-Dateien auch noch HTTP-Requests einsparen. Der Trick ist, die Dateien so weit zusammenzufassen, wie es nur geht, beispielsweise aus vier CSS-Dateien eine zu machen. In den meisten Fällen kann man den gesamten

CSS-Code in eine einzelne Datei packen. So werden vier HTTP-Requests auf einen reduziert und die Seite lädt schneller.

Fortgeschrittene CSS-Techniken

Wer sich tiefer mit CSS auseinandersetzt, wird früher oder später auf den Begriff Shorthand-Anweisungen treffen. Dies ist nichts anderes als eine verkürzte Schreibweise für CSS-Anweisungen. Anwendungsmöglichkeiten dafür gibt es viele. So können beispielsweise komplexe Angaben zu Abständen und Positionierungen extrem verkürzt werden. Statt einzeln margin-left, margin-right etc. aufzurufen, schreibt man alles in eine Zeile. Dies kann dann wie folgt aussehen.

```
margin: 0px 0px 2px 3px;
```

Die Anweisung würde bewirken, dass oben wie rechts keinerlei Abstand zum nächsten Element besteht. Unterhalb wären jedoch zwei und links drei Pixel Abstand zum nächsten Element vorhanden. Ohne diese Kurzanweisung hätte man vier Codezeilen gebraucht. Je

nachdem, wie viele Pixel eine Anweisung hat, kann man die Anweisung gegebenenfalls noch weiter verkürzen. Ein brauchbares Cheatsheet für Shorthand-Anweisungen findet man auf www.leigebler.com/2008/04/css-shorthand-cheat-sheet/.

Um das Ganze noch auf die Spitze zu treiben, beherrschen mittlerweile viele Entwicklungsumgebungen das Minimieren von Stylesheets. Hier werden am Ende alle Möglichkeiten ausgeschöpft, um noch einzelne Kilobytes einzusparen. Darunter fallen Optimierungen wie z. B. das Entfernen aller Leerzeichen, sofern man dies bei den händischen Optimierungen nicht so wieso schon gemacht hat.

CSS-Sprites – grafische Elemente viel schneller laden

Ein weiteres sehr interessantes Thema im Bereich CSS und Minimierung der HTTP-Requests ist das Zurückgreifen auf sogenannte CSS-Sprites. Die Theorie ist denkbar einfach. Alle grafischen Elemente, die sich ohnehin wiederholen, werden in einer gesammelten Bilddatei gespeichert. Somit muss für das Laden der grafischen Elemente einer Website nur noch ein HTTP-Request gemacht werden und nicht mehr für jedes Bildchen ein eigener.

Der anzuzeigende Ausschnitt wird mittels CSS-Anweisung im Browser ausgegeben. Somit kann die Anzahl der Requests sehr schnell reduziert werden, indem man effektiv nur ein einziges Bild, beispielsweise für die Navigation, lädt. Große Onlineshops wie Amazon gehen hier mit gutem Beispiel voran (siehe Abbildung 9). Die Erstellung von CSS-Sprites ist mittlerweile recht einfach geworden, da es auch schon Automatisierungs-Tools im Web gibt. Hierzu zählt unter anderem der CSS-Sprites-Generator von csssprites.com. Dieser Generator macht im Prinzip nichts anderes, als die hochgeladenen Bilder in einer Datei zusammenzufassen. Die Bil-

der müssen anschließend über Background-Anweisung im CSS definiert werden. So kann der Browser interpretieren, an welcher Stelle der Website welche Position des Bildes angezeigt wird. Man gibt somit lediglich die Position des Ausschnitts an, welcher angezeigt werden soll. Eine gute Einleitung in das Thema CSS-Sprites in deutscher Sprache findet man auf www.webanhalter.de/1368-css-sprites.html.

Webinhalte mittels CDN noch schneller ausliefern

Wer seine Seite noch weiter optimieren möchte, kann auch auf Content-Delivery- bzw. Content-Distribution-Netzwerke (CDN) zurückgreifen. Ein CDN stellt Daten je nach Ort des Users bereit. Sitzt man beispielsweise in Frankfurt, wird man von einem Server aus Frankfurt oder Umgebung „beliefert“, wogegen User aus den Staaten von einem Server aus den USA den Content erhalten. Der bekannteste Vertreter ist das Amazon S3 CDN. Moderne Content-Management-Systeme können standardmäßig mit CDNs umgehen. Bilduploads erfolgen dann auch automatisch auf das vorher eingestellte CDN.

Auch Google bietet seit Kurzem mit dem Google Page-Speed-Service ein ei-

genes CDN an. Hier wird versprochen, dass Inhalte bis zu 60 % schneller ausgeliefert werden als auf normalen Webservern. Auf den Servern von Google werden alle wichtigen Schritte durchgegangen, die zum Beschleunigen der Seite wichtig sind. Die Google-Server kümmern sich also darum, dass CSS-Dateien sowie Javascript-Dateien bereits zusammengefasst werden. Auch Bilder werden direkt in das passende Format optimiert. Um die Geschwindigkeit noch weiter zu erhöhen, verteilt Google die Daten auf verschiedene Server im eigenen CDN, sodass mehrere Dateien parallel geladen werden können.

Die Abbildung 12 stellt grafisch die Arbeitsweise eines Content Delivery Networks dar. Ziel eines CDN ist es, Daten auf kürzestem Weg im Browser des Besuchers anzuzeigen. Dazu werden die CDN-Server weltweit regelmäßig synchronisiert, sodass am Ende von jedem Server dieselben Daten ausgeliefert werden können.

Optimierung innerhalb der Templates

Die meisten Optimierungsmöglichkeiten, um HTTP-Requests zu minimieren, bieten vor allem Templates von Content-Management-Systemen. Vor-

allem das CMS Wordpress kann hier großes Potenzial vorweisen. In vielen Template-Systemen verwendet man dynamische Anweisungen wie z. B. `bloginfo('blogname')`, um lediglich den Titel bzw. den Namen des erstellten Blogs auszugeben. Wenn sich der Blog-Name ändert, würde die dynamische Anweisung diesen automatisch ändern. Aber mal ehrlich, wie oft ändert sich ein Blog-Name? Um zu verstehen, dass es hier Einsparpotenzial gibt, muss man über ein wenig Hintergrundwissen verfügen. Wird ein dynamischer Befehl aufgerufen, wird erst eine Verbindung zur Datenbank aufgebaut (Schritt 1). Anschließend wird eine Abfrage an die Datenbank gestellt (Schritt 2) und auf das Ergebnis gewartet. Dieses Ergebnis wird wiederum an das Template-System zurückgegeben (Schritt 3) und erst dann ausgegeben (Schritt 4). Effektiv ergeben sich daraus vier Teilschritte. Um sich diese zu sparen, kann man auch lediglich direkt den Namen als HTML-Code eingeben, anstatt auf dynamische Funktionen zurückzugreifen. Das Ergebnis wird also unmittelbar ausgegeben, ohne dass erst Verbindungen unnötig aufgebaut werden müssen. Diese Vorgehensweise lässt sich auf fast alle verfügbaren Systeme übertragen, da jedes CMS

DAS LEISTUNGSSTARKE UND KOSTENGÜNSTIGE SEO TOOL



Suchmaschinen Ranking

Sichtbarkeitsindex für Deutschland, die Schweiz und Österreich, eigene Keywords tracken



Backlink Analyse

mit eigenen Crawlern, nicht von externen Backlink Daten abhängig



Keyword Datenbank

Über 10 Mio. Keywords, Recherchefunktion, inkl. SEO/SEM Daten



Linküberwachung

tägliche Prüfung von vereinbarten Links, Warnung per E-Mail

www.wiseseosuite.de

SEO SUITE SILBER
für jeden Webmaster
ab 9,90 €
inkl. MwSt.

Neben unserem Einsteigerangebot **SEO SUITE SILBER** führen wir auch umfangreiche Versionen unseres SEO Tools. Testen Sie jetzt ohne Risiko mit unserer **14-Tage-Geld-Zurück-Garantie!**

GUTSCHEIN CODE: #WIB3EO

Jetzt online buchen und einen Gratismonat bei einer Buchung von mind. 3 Monaten sichern. Gültig bis 30.11.2011





Abbildung 12: Grafische Darstellung eines CDN-Netzwerks (Quelle: eigene Erstellung eology)

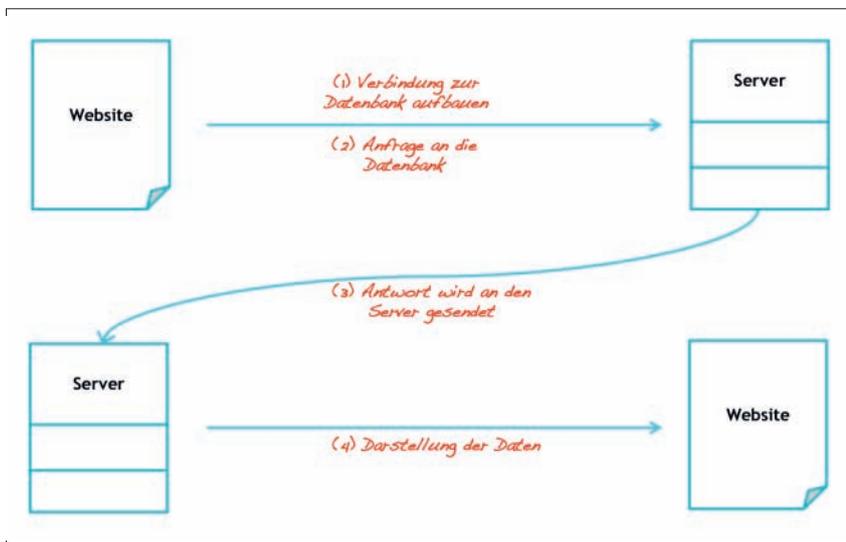


Abbildung 13: Vereinfachte Darstellung einer Webserver-Anfrage (Quelle: eigene Bilderstellung eology)

mit dynamischen Anfragen arbeitet.

HTTP-Requests lassen sich allerdings auch schon beim Erstellen von CSS- und Javascript-Dateien reduzieren. Auch wenn es wegen der Übersicht einfacher ist, Navigation und Content-Anweisungen in zwei separate Dateien zu packen, sind es dann immer noch zwei Requests. Packt man diese stattdessen in eine einzelne Datei, hat man lediglich noch einen einzigen Request. Es spricht absolut nichts dagegen, eine einzelne .js-Datei mit einer Funktionsbibliothek einzubinden, welche dann mittels Skript-Aufruf im Quellcode geladen wird. Wo allerdings sollten die Sachen im Quellcode stehen, um den besten Effekt zu erhalten?

Grundsätzlich stehen CSS-Anweisungen im Header und Javascript-An-

weisungen kurz vor dem schließenden Body-Tag. Zwar kann man diese komplett in den Header setzen, allerdings birgt das einen entscheidenden Nachteil. Javascript-Dateien werden erst fertig geladen, bevor die Seite weitergerendert wird. Bei fünf zu ladenden Javascripts, was in Zeiten von jQuery, mootools und Prototype Standard ist, muss der Browser demnach fünfmal das Laden des HTML-Quelltextes abbrechen, um die Dateien einzulesen. Beim Rendern der Seite wird man auch noch keine Web-2.0-Features benötigen, von daher können diese problemlos erst am Ende der Seite platziert werden. Damit geht nicht nur das Laden schneller, der vermeintliche Besucher sieht auch sofort den Text und weiß, ob er das findet, was er gesucht hat.

Weiterführende Möglichkeiten

Der eine oder andere wird das Wort „minify“ bestimmt schon einmal gehört haben. Minify ist ein Tool, mit dem CSS- und Javascript-Dateien auf recht einfache Weise kombiniert und optimiert werden können. Hier werden beispielsweise Javascript-Dateien kombiniert und auf eine einzelne Datei reduziert.

Ebenso werden Leerzeichen und Leerzeilen entfernt. Genauso wird mit CSS-Dateien verfahren. Dieses Skript lässt sich problemlos überall einbauen. Für die meisten Content-Management-Systeme gibt es bereits fertige Plugins und Addons. Sogar für das Shop-Ungeheuer Magento gibt es ein Minify-Skript namens „Foomans Speedster“. Ein weiterer Vorteil des minify-Pakets ist das Cachen der Dateien. So können diese noch wesentlich schneller geladen und ausgeliefert werden. Für weitere Informationen sollte man code.google.com/p/minify/ einen Besuch abstatten.

Fazit

Auch wenn die Website-Geschwindigkeit nicht das alleinige Heilmittel für gute Rankings darstellt, sollte man sich diesem Thema ausführlich widmen – denn jedes wichtige, positive Signal an Google zahlt auf gute Positionen ein. Und auch wenn nicht sofort Rankingverbesserungen erkennbar sind, werden es einem auf jeden Fall die Website-Besucher danken. Denn einen langsamen Seitenaufbau mag wohl niemand. Da Google in letzter Zeit viele neue Features zum Thema Page Speed veröffentlicht hat, ist sicherlich davon auszugehen, dass der Faktor Ladegeschwindigkeit in Zukunft noch stärker bei der Rankingbewertung gewichtet wird. Wer tiefer in diese Materie einsteigen möchte, sollte sich am besten gezielt mit Themen wie Opcache, Datenbanktuning oder gzip beschäftigen.¶