

Website Performance Optimierung

Fokus: Frontendoptimierung

form4 GmbH & Co. KG

Jan-Henrik Hempel

Telefon: 030.278784-13

E-Mail: jan-henrik.hempel@form4.de

Website Performance Optimierung

Überblick

- 1 Relevanz
- 2 Requestreduktion und Ladeoptimierung
- 3 Cachingoptimierte Webkonzepte
- 4 Links

Relevanz

... einer guten Website Performance

Eine schnelle Website ist nicht nur eine schöne Sache für den Nutzer, sondern beeinflusst tatsächlich den geschäftlichen Erfolg:

Einige Fakten

- ▶ Google nimmt die Ladezeit als Faktor in das Ranking mit auf
<http://googlewebmastercentral.blogspot.com/2010/04/using-site-speed-in-web-search-ranking.html>
- ▶ Erhöhung der Page Impressions pro Visit und der Verweildauer
- ▶ Bessere Ziel-Conversions
- ▶ Amazon verzeichnet Umsatzeinbußen bei Verlangsamung von Requests von wenigen 100 Millisekunden
Yahoo! hat weniger Suchanfragen pro User bei Verlangsamung von Requests von wenigen 100 Millisekunden
(High Performance Web Sites, Steve Souders, ISBN: 978-0596529307)

Website Performance Optimierung

Überblick

- 1 Relevanz
- 2 Requestreduktion und Ladeoptimierung
- 3 Cachingoptimierte Webkonzepte
- 4 Links

Requestreduktion

... durch Script Merging (CSS und JS)

Eine naheliegende Möglichkeit Requests zu reduzieren, ist das Zusammenlegen von eingebundenen Dateien – insbesondere Stylesheets und Javascript-Dateien.

Maßnahme

- ▶ Mergen von CSS Dateien
- ▶ Mergen von Javascript-Dateien
- ▶ Das Mergen kann durch ein automatisiertes Script erfolgen (z.B. Ant-Deploy)
- ▶ Das Mergen kann per Scriptmerger durch das CMS erfolgen
- ▶ Expire Header nicht vergessen

Effekt

- ▶ Von z.T. ca. 5 bis 10 CSS Dateien kann i.d.R. eine Reduktion auf 2 oder 3 erfolgen
- ▶ Dadurch werden auch die 4er-Blöcke für die Requests entlastet

Requestreduktion

... durch optimierte HTTP Expire Header

Auch wenn eine Datei bereits im Browser Cache (oder Proxy) gespeichert ist, fragt der Browser i.d.R. Den Server, ob die Datei noch aktuell ist.

Ist dies nicht der Fall, gibt der Server einen „304 Not modified“ zurück. Hier erfolgt kein wirklicher Transfer, dennoch entsteht aufgrund der Vielzahl an Dateien auf Websites eine nicht unwesentliche Menge an Requests.

Maßnahme

- ▶ Setzen von Expire Headern für statische Dateien (CSS, JS, Grafiken) per .htaccess
- ▶ z.B. Expires Sat, 01 May 2010 19:32:04 GMT
- ▶ Cache-Control max-age=604800

Vorsicht

- ▶ Die Dateien sollten eine Versionsnummer im Dateiname führen, um Updates zu ermöglichen

Requestreduktion

... durch optimierte HTTP Expire Header

URL	Status	Domain	Size	Timeline
▶ http://www.vhw.de/fernlehrgaenge/	200 OK	vhw.de	32.4 KB (?)	1.01s
▶ GET stylesheet_78997b7621.c	304 Not Modified	vhw.de	9.6 KB	61ms
▶ GET layout.css	304 Not Modified	vhw.de	2.6 KB	66ms
▶ GET navigation.css	304 Not Modified	vhw.de	11.4 KB	67ms
▶ GET box.css	304 Not Modified	vhw.de	12.7 KB	73ms
▶ GET elements.css	304 Not Modified	vhw.de	35.7 KB	87ms
▶ GET jCalendar.css	304 Not Modified	vhw.de	2.8 KB	978ms
▶ GET blog.css	304 Not Modified	vhw.de	4.9 KB	1.08s
▶ GET jquery.js	304 Not Modified	vhw.de	29.1 KB	101ms
▶ GET jquery.form.js	304 Not Modified	vhw.de	22.6 KB	91ms
▶ GET jquery.md5.js	304 Not Modified	vhw.de	9.2 KB	106ms
▶ GET ajax_pi.js	304 Not Modified	vhw.de	1.8 KB	116ms
▶ GET misc.js	304 Not Modified	vhw.de	4.7 KB	128ms
▶ GET jquery.datePicker.min-2.1	304 Not Modified	vhw.de	12.9 KB	145ms
▶ GET date.js	304 Not Modified	vhw.de	11.3 KB	146ms
▶ GET date_de.js	304 Not Modified	vhw.de	586 B	158ms
▶ GET jquery.bgiframe.min.js	304 Not Modified	vhw.de	1.4 KB	169ms
▶ GET vhw_datePicker.js	304 Not Modified	vhw.de	3.5 KB	185ms
▶ GET javascript_a708894199.js	304 Not Modified	vhw.de	951 B	198ms
▶ GET bg_body.png	304 Not Modified	vhw.de	2.4 KB	77ms
▶ GET logo.png	304 Not Modified	vhw.de	18.2 KB	78ms
▶ GET e60af36a40.png	304 Not Modified	vhw.de	63.8 KB	76ms
▶ GET 29b7eac076.jpg	304 Not Modified	vhw.de	3.3 KB	87ms
▶ GET c27ee3782b.jpg	304 Not Modified	vhw.de	19.5 KB	89ms
▶ GET 9a921b7f66.jpg	304 Not Modified	vhw.de	29.4 KB	93ms
▶ GET 448afb8247.jpg	304 Not Modified	vhw.de	21.1 KB	113ms
▶ GET ee5fde7ec4.jpg	304 Not Modified	vhw.de	15.1 KB	115ms
▶ GET bg_header.png	304 Not Modified	vhw.de	30.3 KB	163ms
▶ GET bg_search.png	304 Not Modified	vhw.de	276 B	163ms
▶ GET bt_start.png	304 Not Modified	vhw.de	1.1 KB	183ms
▶ GET ...	304 Not Modified	vhw.de

Insgesamt 47 304 Requests, die lila Balken sind Wartezeiten

Requestreduktion

... durch Sprite Grafiken

Viele Dateien in Websites sind Layoutgrafiken für Buttons, Icons, Ecken, Balken usw. Zumeist handelt es sich um kleine Dateien, die zwar keine hohen Dateigrößen haben, aber durch ihre große Anzahl die Requestanzahl aufblähen.

Maßnahme

- Zusammenfassung von Grafiken in Sprite Dateien

Links und Tools

- CSS Sprite Generator
<http://spritegen.website-performance.org/>

Ladeoptimierung

... durch Script Minifiing (CSS und JS)

Textdateien (bei Websites insbesondere CSS- und Javascript-Dateien) haben ein hohes Potential für Komprimierungen. Ein Teils davon kann durch Entfernung von Spaces, Linebreaks etc. ausgenutzt werden.

Dadurch können ca. 20 % bis 40 % der Dateigrößen eingespart werden. Relevant ist dies insbesondere, da Styles und Scripte immer auch bei der Einstiegsseite mitgeladen werden müssen, wenn noch kein Cache vorhanden ist.

Maßnahme

- Verkleinerung von Dateien

Links und Tools

- Online Javascript compressor
<http://javascriptcompressor.com/>
- CSS Drive CSS Compressor
<http://www.cssdrive.com/index.php/main/csscompressor/>

Ladeoptimierung

... durch HTTP-Kompression

Insbesondere Textdateien (HTML, CSS, Javascript) haben ein hohes Potential für Kompressionen. Zur Optimierung der Transferzeiten kann hierfür eine HTTP-Kompression eingesetzt werden.

Maßnahme

- ▶ HTTP-Kompression
- ▶ Accept-Encoding: gzip, deflate

Ladeoptimierung

... durch Verteilung auf Subdomains

Regulär lädt ein Browser nur vier Requests parallel von einem Host. Die heutigen Bandbreiten lassen i.d.R aber mehr parallele Downloads zu, ohne dass es Einbußen gibt – zumal wenn es sich primär um CSS-, JavaScript- und Bilddateien handelt.

Maßnahme

- ▶ Verschiedene Dateien oder Dateitypen können auf Subdomains verteilt werden, z.B.
 - ▶ `css.domain.com`
 - ▶ `js.domain.com`
 - ▶ `img.domain.com`
 - ▶ `video.domain.com`
- ▶ Der ausliefernde Server kann dabei der selbe sein – es geht primär um die Erhöhung der parallelen Downloads.

Website Performance Optimierung

Überblick

- 1 Relevanz
- 2 Requestreduktion und Ladeoptimierung
- 3 Cachingoptimierte Webkonzepte
- 4 Links

Cachingoptimierte Webkonzepte

Verwendung statischer und parametrisierter Ansichten

Um in allen Ebenen eine hohe Performance und Skalierbarkeit zu erreichen, ist einer der Schlüssel möglichst alles daran zu setzen, dass möglichst viele Einzelseiten einer Website statisch sind.

Denn dann können verschiedenste Optimierungstools eingesetzt werden:

- ▶ Static File Cache
- ▶ Proxy, CDN
- ▶ Squid
- ▶ ...

Eine oft unterschätzte aber naheliegende Möglichkeit ist die klare Parametrisierung aller Seiten und Varianten, wie z.B.

- ▶ Sprachvarianten
- ▶ Unterschiedliche Schriftgrößen oder Kontrastansichten
- ▶ Filterfunktionen, Tagnavigationen, Tagcloudansichten und Zielseiten
- ▶ ...

Cachingoptimierte Webkonzepte

Berechnungen per Javascript im Frontend

Cachingkonzepte werden gerne sehr schnell ausgehebelt, sobald userspezifische Darstellungen im Frontend erfolgen sollen.

Typische Punkte sind z.B.:

- ▶ Loginstatus
- ▶ Merkzettel- oder Warenkorbblink mit Anzahl der Produkte
- ▶ Lokalisierung nach Land, Bundesland, Postleitzahl o.ä.

Solche Informationen können zumeist wunderbar per Javascriptberechnung mit vorgeladenen Daten im Frontend erfolgen und somit zur Serverentlastung führen.

Cachingoptimierte Webkonzepte

Verwendung von Cookies anstatt Server-Sessions

Spezielle Werte oder Attribute von Usern müssen häufig nicht in der Serversession gespeichert werden. Leider erfolgt die Speicherung und die Auswertung von Cookie-Werten allzu oft per PHP.

Oft reicht die Speicherung per Javascript im Cookie.

Dadurch bleiben Seiten statisch und unabhängig von serverseitigen Berechnungen.

Cachingoptimierte Webkonzepte

Nachladen per AJAX

Einige dynamische Funktionen von Websites sollen auf allen Seiten an zentraler Stelle zu sehen sein. Zumeist bleibt jedoch der größte Teil der Einzelseiten statisch und nur einige nutzen die userindividuellen Einstellungen in der Anzeige des Inhalts.

Das Problem dabei ist, dass durch ein einziges userindividuelles Element, sofort alle Einzelseiten nicht mehr statisch sind.

Diese Problem kann durch das Nachladen von dynamischen Informationen per AJAX optimiert werden.

Dynamische Informationen können per Cookie/Javascript auf alle Seiten statischen Seiten gerendert werden. Dort wo Inhalte dynamisch werden, können die entsprechenden Daten per AJAX nachgeladen werden.

Typische Funktionen hierfür sind:

- ▶ Loginstatus
- ▶ Merkmalt
- ▶ Warenkörbe
- ▶ Userspezifische Anzeige von Kontaktdaten oder Ansorechpartnern (z.B. nach Land)

Cachingoptimierte Webkonzepte

Cookie Auswertung per Rewrite Engine

Die Auswertung von Cookie-Werten bei wiederkehrenden Usern wird häufig per serverseitigen Script innerhalb eines CMS realisiert.

Allerdings kann eine solche Auswertung auch per Apache Rewrite Engine erfolgen und Redirects auf statische Seiten erzeugen.

So schnell ist kein CMS-Rendering.

Anwendungen hat diese Funktions typischerweise bei

- ▶ der Auswertung der durch den User zuletzt verwendeten Sprache mit Redirect
- ▶ der Auswertung der durch den User zuletzt verwendeten Schrift- oder Kontrasteinstellung
- ▶ Spezieller Redirects in userspezifisch gespeicherte Themenbereichs o.ä.

Javascript vs. Barrierefreiheit

Zugänglichkeit der Informationen

Viele optimierte Konzepte setzen auf Javascript, um Logik und Berechnungen im Frontend zu realisieren und damit den Server und die Client-Server-Kommunikation zu entlasten.

- ▶ Je nach Zielgruppe einer Website browsen ca. 1 % bis 5 % der User ohne Javascript
- ▶ Barrierefreie Websites müssen jedoch auch ohne Javascript funktionieren, egal ob es sich nur noch um 1 % der User handelt.

Aber es gilt:

- ▶ Primär muss die Information erreichbar sein! Häufig ist das auch ohne die Javascript-Features gegeben. Evtl. fehlt ein Merkmalszettel oder eine intuitive Google Maps Navigation, aber solange alle Informationen erreichbar sind, ist das nicht so schlimm.
- ▶ Alternativ können Javascript Fallbacks eingebaut werden. Wenn diese dann per serverseitiger Berechnung erfolgen entstehen keine wesentlichen Einbußen. Die Seite wird ggf. in 2 ½ anstatt in 1 Sekunde geladen. Und der Server muss für 1-5 % der User mehr rechnen.

Website Performance Optimierung

Überblick

- 1 Relevanz
- 2 Requestreduktion und Ladeoptimierung
- 3 Cachingoptimierte Webkonzepte
- 4 **Links**

Links und Bücher

... zur Frontend Optimierung

- ▶ Best Practices for Speeding Up Your Web Site
<http://developer.yahoo.com/performance/rules.html>
- ▶ High Performance Web Sites
Steve Souders
ISBN: 978-0596529307
- ▶ Online Javascript compressor
<http://javascriptcompressor.com/>
- ▶ CSS Drive CSS Compressor
<http://www.cssdrive.com/index.php/main/csscompressor/>
- ▶ CSS Sprite Generator
<http://spritegen.website-performance.org/>

Vielen Dank.

form4 GmbH & Co. KG

Jan-Henrik Hempel

Tel.: 030.278784-13

E-Mail: jan-henrik.hempel@form4.de