HTW Dresden
Fachbereich Informatik

Leistungsuntersuchung von Webserveranwendungen und Optimierung auf Basis von Drupal 5 (am Beispiel vo

Bachelorarbeit

„Leistungsuntersuchung von Webserveranwendungen und Optimierung auf Basis von Drupal 5 (am Beispiel von ITsax.de)“

Autor: Schneider, Martin

Seminargruppe 08/041/02

Betreunder Professor: Prof. Vogt

Datum: 28. August 2011

Inhaltsverzeichnis

Teil I.

Einleitung

1. Motivation

Der Autor dieser Arbeit absolvierte sein Pflichtpraktikum in der pludoni GmbH und war bis zum heutigen Tag als Werksstudent tätig.

1.1. pludoni GmbH

“pludoni kommt aus dem Esperanto und bedeutet weitergeben.” Das Unternehmen wurde 2008 von Dr. Jörg Klukas gegründet und hat als Ziel den regionalen Mittelstand bei der Kommunikation und Bewerbersuche durch Communities zu unterstützen. Die von pludoni gemanagten Communities heißen ITsax.de, ITmitte.de und MINTsax.de. Sie richten sich an Technologie-Firmen aus Sachsen beziehungsweise Mitteldeutschland und versuchen über zahlreiche Dienstleistungen einen Mehrwert für die beteiligten Unternehmen zu schaffen, im Allgemeinen gehören dazu:

- Aggregation, Veröffentlichung und Weitergabe von Stellenanzeigen
- Organisation von regelmäßigen Community-Treffen zum Austausch der Unternehmen,
- Unterstützung und Beratung bei der Gestaltung Suchmaschinen-optimierter Inhalte und
- Infrastruktur zur gegenseitigen Empfehlung von Fachkräften und Lernenden.

1.2. Ziel

Alle Communities sind zum heutigen Zeitpunkt durch das Content Management System Drupal 5 realisiert. Drupal 5 ist mittlerweile schon vier Jahre alt und nicht mehr auf dem neuesten Stand was Web Performance betrifft. Es gibt Drupal mittlerweile schon in der Version 7.7 und es sind viele Verbesserungen in Sachen Geschwindigkeit eingeflossen. Leider ist es nicht trivial ein Upgrade durchzuführen, da viele Strukturen sich stark verändert haben. Aus diesem Grund wurde der Autor damit beauftragt die Möglichkeiten einer Drupal 5 Optimierung am Beispiel der ältesten Community

ITsax.de zu ergründen und wenn vorhanden durchzuführen. Dabei soll ein Leitfaden entwickelt werden, anhand dessen Drupal 5 und allgemein Webseiten effizient optimiert werden können.

Teil II.

Theoretische Grundlagen

2. Web Performance

> "Human beings don't like to wait. We don't like waiting in line at a store, we don't like waiting for our food at a restaurant, and we definitely don't like waiting for Web pages to load."
>
> Alberto Savoia

Web Performance ist in der letzten Zeit ein immer präsenteres Thema geworden. Aber was bedeutet Web Performance eigentlich? Zum einen gibt es die subjektive Performance, die ein Besucher beim Browsen einer Webseite empfindet. Zum anderen kann man auch eine objektive Aussage über die Geschwindigkeit mit der eine Webseite generiert, ausgeliefert und beim Besucher im Browser angezeigt wird, treffen.

2.1. Usability

Web Performance hat einen großen Einfluss auf die Bedienbarkeit von Webseiten. Ohne die starke Verbesserung ihrer Performance hätte Google mit GMail und ihren anderen Web Applikationen keinen Erfolg erzielen können. Nur wenn eine Anwendung auch in einer akzeptablen Geschwindigkeit auf Benutzerinteraktionen reagieren kann, hat sie eine Chance auf dem Markt zu bestehen. Für Unternehmen gibt es noch weitere Faktoren, die evaluiert werden müssen:

- langsamere Webseiten führen zu Vertrauensverlust
- empfundenem Qualitätsverlust
- niedrigere Konversionsraten
- höhere Bailoutraten
- Nutzerfrustration

2.2. Google Ranking

Für (Internet-)Firmen ist es von immenser Bedeutung gefunden zu werden. Ein Großteil der Internetnutzer sucht Angebote über die Google-Suche. Google hat mit ihrer Initiative "Let's make the web faster" für viel Entwicklung und Aktivität im Bereich Web Performance gesorgt und arbeitet zielgerichtet weiter in diese Richtung. Dazu gehört seit 2008 Google AdWords und seit 2010 die Google Suche selbst. Dies stellt viele Unternehmen vor die Aufgabe, ihre Webseite zu optimieren und zu beschleunigen, um im Google Ranking konkurrenzfähig zu bleiben. Wie genau Google die Webseiten testet, ist nur zum Teil bekannt, da diese Informationen zu Googles Geschäftsgeheimnissen gehören. Es finden sich aber einige Fakten, die bei der Optimierung für die Google Suche, im Bereich Web Performance, helfen können. Bekannt ist dass: - der Google Suchmaschinen Bot nichts mit der Geschwindigkeitsmessung zu tun hat. - Google nur Daten von echten Nutzern, die die Google Toolbar in ihrem Browser installiert haben, nutzt. - Daten nur für Internet Explorer ab Version 6 und Firefox ab Version 2 verfügbar sind - die Onload Zeit gemessen wird. - das verzögerte Nachladen von Inhalten zu einem besseren Ergebnis führt, da die Onload Zeit verbessert wird.

2.3. Serverlast

Eine umfassende Verbesserung der Auslieferungszeiten von Webseiten hat direkten Einfluss auf die Serverperformance, wie man leicht ermitteln kann. Wenn eine Webseite nur noch die Hälfte der Zeit benötigt, um ausgeliefert zu werden, hat man doppelt soviel Auslieferungskapazität zur Verfügung. Das spart Kosten für Server und in geringerem Maße Traffic. Bestimmte Techniken ermöglichen auch einen Großteil an Prozessorlast einzusparen, der dann für andere Aufgaben zur Verfügung steht.

3. Performance Bottlenecks

Performance Bottlenecks oder auch Performance Flaschenhälse genannt, beschreiben Schwachpunkte in einem Datenverarbeitungssystem. Auf Web Performance angewendet, kann man das in clientseitige Probleme und serverseitige Probleme unterteilen.

3.1. Clientseitige Bottlenecks

3.1.1. Netzwerkverbindung

Die Schwachstelle Netzwerkverbindung kann auch auf Serverseite auftreten, aber da kann mit mehr Servern und besseren Anbindungen Abhilfe geschafft werden. Viel problematischer wird das auf der Clientseite da selten bekannt ist mit welcher Anbindung Nutzer eine Webseite besuchen werden.Da in Deutschland der Breitbandausbau noch lange kein Stadium erreicht hat, der es Webseitenbetreibern erlauben würde auf die Größe der Webseite die sie an ihre Betrachter ausliefern keinen Wert zu legen. Der Breitbandausbau peilt für 2011 eine 100 prozentige Versorgung mit 1 Mbit/s an und so sind noch viele Nutzer sind mit langsamen Internetzugängen im Internet unterwegs.

3.1.2. Browser

Als Schnittstelle zwischen Mensch und Webseite ist der Browser ein besonders kritischer Punkt und muss bei Web Performanceanalysen besonders beachtet werden. Zum einen müssen die verschiedenen Browserfamilien, mit ihren Eigenheiten und Problemen und zum anderen die Tatsache das ein Browser nur so schnell arbeiten kann, wie der Computer auf dem er ausgeführt wird zulässt, beachtet werden. Problemzonen sind:

- Anzahl von zu berechnenden DOM-Elementen
- Javascript Ausführungszeiten
- Anzahl benötigter HTTP Zugriffe
-

3.1.3. DOM Elemente

3.2. Serverseitig

3.2.1. Datenbankabfragen

3.2.2. Skriptausführung

3.2.3. Skriptkompilierung

4. Technologiestack und Optimierungsmöglichkeiten

Im Rahmen dieser Bachelorarbeit werden verschiedenste Technologien verwendet und auf ihnen aufbauende Werkzeuge zu Hilfe genommen. Um zu verstehen wo Probleme lokalisiert sind und wie solche Schwachstellen zu finden sind muss man sich mit dem vorhandenem Technologiestack auseinandersetzen und ihn analysieren.

Server Ein Server ist ein Computer, der Informationen und Dienste für andere Computer zur Verfügung stellt

Betriebssystem Die Software die auf dem Server läuft. In der pludoni GmbH wird das Linuxderivat Debian eingesetzt.

Datenbank Als Datenbank wird eine strukturierte Sammlung von Daten bezeichnet. MySQL ist hier die erste Wahl.

Web Server Der Web Server ist für die zuverlässige Auslieferung von Webseiten zuständig. Er beantwortet die Anfragen die die Nutzer mit dem Browser stellen. Apache2 wird dafür in der pludoni GmbH eingesetzt.

PHP PHP ist eine Programmiersprache, die es Web Entwicklern ermöglicht dynamische Webseiten zu programmieren.

Drupal Drupal ist ein Framework, welches in PHP geschrieben ist und viele direkt nutzbare Funktionen mitbringt. Dazu gehören unter anderem Administrationsoberflächen, Newsaggregatoren, Veröffentlichungsabläufe für Artikel und Blogbeiträge sowie Suchmaschinenoptimierungen. Außerdem ermöglicht Drupal die Installation weiterer, durch Nutzer entwickelte, Module. Dadurch wird eine fast unbegrenzte Funktionsvielfalt ermöglicht. Im vorliegenden Fall wird Drupal in der Version 5 eingesetzt.

Client Im Browser der Nutzer kommt am Ende HTML an, dies wird durch Drupal generiert und vom Webserver ausgeliefert. Dabei werden Bilder verarbeitet Javascript Programme ausgeführt und andere Elemente wie Flashapplikationen und Videos berechnet.

4.1. Server

4.2. PHP

4.3. Drupal (5)

Drupal ist ein Content-Management-System, dass heißt es ermöglicht dem Nutzer eine dynamische Webseite zu erstellen ohne spezielle Programmierkenntnisse zu besitzen.

4.4. Benchmark - Werkzeuge

4.5. Debugger / Profiler

5. Methoden der Web-Performance Optimierung

Die Methoden zur Leistungssteigerung können wie im Technologiestack untergliedert werden und jedes Element muss für sich betrachtet werden.

Server Der einzige Parameter der am Server verbessert werden kann ist seine Leistung, dass heißt Datendurchsatz und Rechengeschwindigkeit. Dies geschieht durch CPU Upgrades oder Arbeitsspeichererweiterung. Weitergehend kann man die Festplattenzugriffsgeschwindigkeiten durch RAID Verbünde und neue Technologien wie SSD Speicher verbessern.

Betriebssystem Ansatzpunkte für Verbesserungen auf Betriebssystemebene sind Memory-Mapping, dass heißt Speicherbereiche die normalerweise auf der Festplatte liegen werden in den Arbeitsspeicher umgelagert um die Latenzen zu verringern. Dies wird genutzt um Caches zu beschleunigen, die normalerweise von der Festplatte lesen.

Datenbank Auf Datenbankebene kann im Fall von MySQL nur beschränkt optimiert werden. Zum einen sind Indizes anzulegen bei häufig genutzten Tabellen und zum anderen kann man MyISAM statt InnoDB nutzen um Performanter zu sein.

Web Server Optimierungen am Webserver sind sehr schwierig da die Webserver Performance hauptsächlich von der Serverleistung abhängt. Die Anzahl der gleichzeitigen Zugriffe wird maßgeblich durch den verfügbaren Arbeitsspeicher und die verfügbare Bandbreite eingeschränkt.

PHP

Drupal

HTML

Teil III.

Praxisteil

6. Versuchsaufbau

Der Optimierungsversuch kann leider nicht auf dem Livesystem durchgeführt werden, da das die Stabilität beeinträchtigen würde. Deswegen wurde ein Entwicklungssystem genutzt um die Verschiedenen Methoden zu analysieren. Dass Entwicklungssystem ist eine neuere Anschaffung der pludoni GmbH und dadurch im Vergleich zum Livesystem leistungsfähiger. Dadurch wird sich das Endergebnis, wenn es nach dem Abschluss der Arbeit auf das Livesystem übertragen wurde in seinen Kenndaten unterscheiden. Trotz alledem werden die Verbesserungen relativ zum Testsystem proportional ausfallen.

- Testplattform: pludoni Server eq4
- Prozessor: Intel® Core™ i7-920
- Arbeitsspeicher: 8 GB DDR3 RAM
- Festplatten: 2 x 750 GB SATA-II HDD (Software-RAID 1)
- Netzwerkverbindung: 100Mbit
- Server Software: Apache/2.2.16
- Server Hostname: itsax.it-jobs-und-stellen.de
- Server Port: 80

7. Testverfahren

Die rein serverseitigen Optimierungen wurden mit ab getestet. ab, das für apache bench steht, wurde entwickelt um Apache Server zu testen. Es analysiert wie lange ein Server braucht um Webseiten auszuliefern und wieviele Anfragen er pro Sekunde befriedigen kann, ohne das es zu Ausfällen kommt. ab führt 50 Anfragen hintereinander auf die Ressource aus, und berechnet dann anhand der Ergebnisse die durchschnittliche Antwortzeit. Alle Änderungen die erst im Frontend, also im Browser, sichtbar

werden wurden mit der Analyse von webpagetest.org untersucht. Dabei wurden folgende Testparameter genutzt:

- 10 konsekutive Tests (maximum)
- Standort des Clients: Frankfurt a. M.
- Browser: IE9
- Connection: DSL 1,5 Mbps / 50ms RTT
- Only First View

Browsertests sind eine sehr komplizierte Angelegenheit, da es zum einen sehr viele Verschiedene Browser gibt und zum Anderen diese Browser in den verschiedensten Umgebungen ausgeführt werden. Ein Netbook zum Beispiel wird eine Webseite immer langsamer Darstellen, als ein Hexacore der neuesten Generation. Aus diesem Grund ist es von Vorteil eine Standardisierte Umgebung zu nutzen. Besonders die verschiedenen Analysen die automatisch durchgeführt werden sind positiv hervorzuheben:

- Bewertung der Website nach den Google Page Speed Kriterien
- Inhaltsanalyse nach MIME Typen
- Performance Review in dem jedes Element der Homepage für sich betrachtet wird
- Wasserfalldiagramm in dem schnell Problemzonen erkannt werden können
- Videovergleich des Seitenaufbaus mit anderen Webseiten

Andere Analyse Verfahren wie Profiling und Debugging auf Codeebene wurden nur sporadisch eingesetzt, da sie den Anwendungsfall Webseite nur eingeschränkt betreffen. Durch Caching kann der Server zum großen Teil vollständig entlastet werden und die Codeausführungszeit kann eliminiert werden. Bei Anwendungsfällen die kein Caching erlauben, wie zum Beispiel eine Suche oder eine Applikation, ist es sehr hilfreich diese Tools zu verwenden. Sie ersparen viel Arbeit, da die Problemstellen schnell ausgemacht werden können. Von Hand diese Arbeiten durchzuführen ist in kleinen Projekten sehr mühsam und in großen Projekten fast unmöglich.

8. Ausgangszustand

Sreeram Ramachandran ein Software Ingenieur der Firma Google hat eine Analyse über 4.2 Milliarden Seiten veröffentlicht. Diese ist im Rahmen der Initiative "Let's make the web faster" entstanden und zeigt häufige Fehlerquellen und ungenutztes Potential auf. Die durchschnittliche Webseite hat laut Ramachandran 320 kB Größe, 44 verschiedene Ressourcen und es werden nur 66% der komprimierbaren Inhalte tatsächlich komprimiert. Itsax.de hat 106 Ressourcen und 444 kB an Daten. Schon anhand dieser zwei Zahlen lässt sich eine vergleichsweise schwache Leistung vorhersehen. Besonders die Anzahl an verschiedenen Ressourcen deutet auf Missstände hin, da Parallelisierung von Zugriffen nur bis zu einem bestimmten Grad möglich ist. Die Time to First Byte(TtFB) von 674ms bezeichnet die Zeit die vergeht bis der Webbrowser die ersten Daten empfängt, dass bedeutet aber noch nicht das der Nutzer schon Inhalte präsentiert bekommt. Die Inhalte werden erst angezeigt nachdem die Time to Start Render(TtSR) vergangen ist, der Nutzer muss demnach ungefähr zwei Sekunden warten bis die Webseite im Browser anfängt sich aufzubauen. Die Load Time(LT) bezeichnet dann die Zeit die vergeht bis die Seite komplett dargestellt wird und der Benutzer sie ohne Einschränkungen bedienen kann. Es können aber auch nach der LT noch Inhalte nachgeladen werden, wie zum Beispiel gestreamte Videos oder andere asynchrone Inhalte. Diese nachgeladenen Inhalte wirken sich aber nicht mehr Negativ auf die User Experience aus, solange sie im Rahmen bleiben und nicht wichtige Teile der Webseite wie zum Beispiel das Hauptmenü noch per Flash geladen werden müssen. Die Anzahl der DOM Elemente bezeichnet alle vom Browser zu verarbeitenden Objekte und ist ein Indikator für die Komplexität der Webseite, je mehr Elemente also vorhanden sind desto länger muss der Browser die Positionierung und Darstellung berechnen. Mit 855 DOM Elementen ist ITsax.de da schon auf der komplexeren Seite, was zu einem Großteil Drupal anzurechnen ist, welches durch seine gute Konfigurierbarkeit Einbußen in Die Inhalte auf der Seite Itsax.de sind in Abbildung ? dargestellt, einmal im Bezug auf Größe und einmal aufgeschlüsselt nach der Anzahl der benötigten Requests um die Inhalte vom Server anzufordern. Die folgenden Ergebnisse wurde mit dem Analysetool webpagetest.org ermittelt. Leicht zu erkennen ist, dass der HTML Anteil sowohl bei der Größe als auch bei Anzahl der HTTP Anfragen eine Untergeordnete Rolle spielt. Nicht vergessen werden darf aber die Zeit die der Server benötigt um den HTTP Code zu generieren. Dies kann man sehr gut in einem Wasserfalldiagramm erkennen, in dem der Start des Ladevorgangs hervorgehoben wurde. Bevor der Initiale GET Request abgeschlossen ist, weiss der Browser noch nicht welche Ressourcen er

laden muss und es können noch keine anderen Aktionen ausgeführt werden.

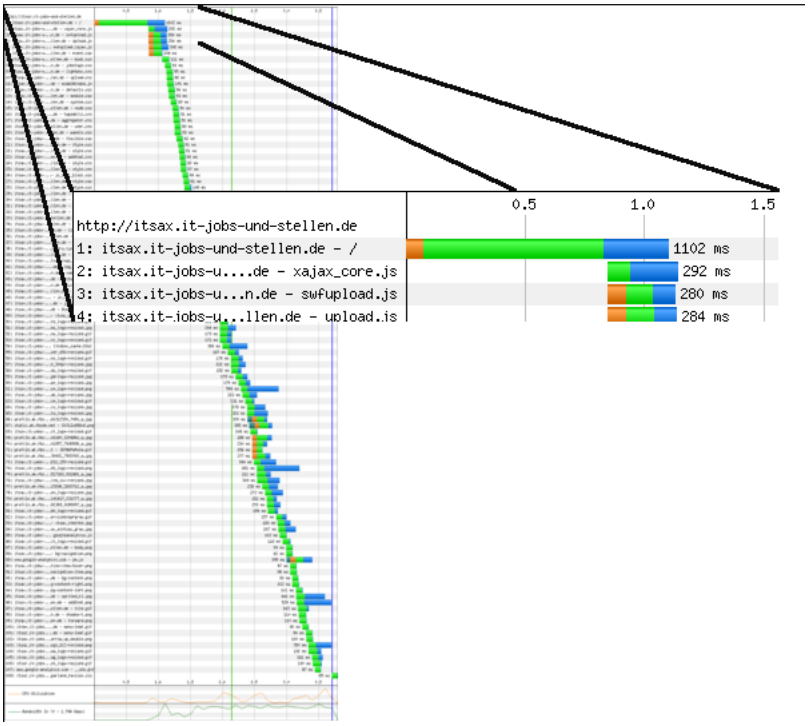

Abbildung 1: Wasserfalldiagramm: Ausgangszustand

Load Time: 3.728 Time to First Byte: 0.674s Time to Start Render: 2.002s #DOM Elements: 855

Typ	**Anzahl**
text/css	24
image/gif	23
image/jpeg	22
javascript	20
image/png	15
text/html	2

Typ	**Byte**
image/jpeg	142251
image/png	101428

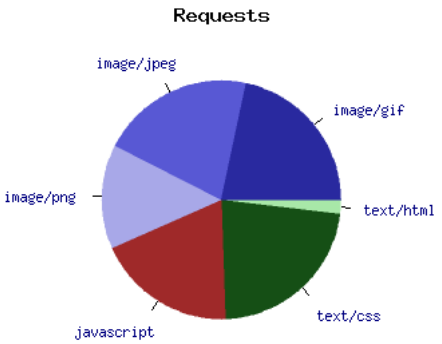

Abbildung 2: HTTP Anfragen

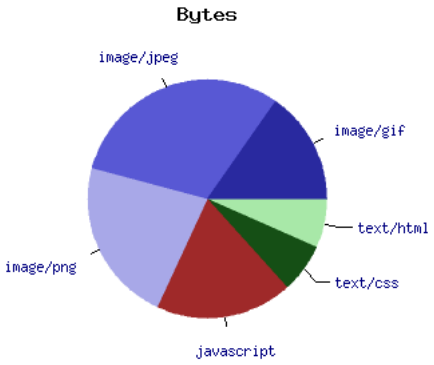

Abbildung 3: Größe der Inhalte

javascript 84758
image/gif 73378
text/css 30222
text/html 29917

Listing 1: Ausgabe von ab

```
1  Server Software:        Apache/2.2.16
2  Server Hostname:        itsax.it-jobs-und-stellen.deDocument Path:
                 /
3  Document Length:        65218 bytes
4
5  Concurrency Level:      1
6  Time taken for tests:   18.182 seconds
7  Complete requests:      50
8
9  Write errors:           0
10 Total transferred:      3266726 bytes
11 HTML transferred:       3239226 bytes
12 Requests per second:    2.75 [#/sec] (mean)
13 Time per request:       363.640 [ms] (mean)
14 Time per request:       363.640 [ms] (mean, across all concurrent
       requests)
15 Transfer rate:          175.46 [kBytes/sec] received
```

9. Implementierung und Test der einzelnen Methoden

Im folgenden Abschnitt werden die eingesetzten Methoden dargestellt und ihre Auswirkungen auf die Webperformance der Seite itsax.de dargestellt. In diesem Abschnitt werden alle Methoden die vom Autor, nach der Analyse des Ausgangszustands, für möglicherweise sinnvoll befunden wurden getestet. Die Methode wird jeweils mit dem Startwert verglichen und der Aufwand wird eingeschätzt.

9.1. APC

Als allererste Maßnahme wurde ein PHP Opcode Cache ausprobiert. Opcode Caches speichern die Ergebnisse vorangegangener Operationen und kann so beim nächsten Aufruf direkt die Ergebnisse ausliefern, wenn Anfrage gleichgeblieben ist. Solche Systeme sind immer zu empfehlen, da sie normalerweise keinerlei Probleme bereiten und alle

PHP Anfragen beschleunigen.Aktuell gibt es mehrere verschiedene sogenannte PHP Beschleuniger unter ihnen nimmt aber APC eine führende Position ein. Hauptsächlich weil es der zuverlässigste unter den nicht-kommerziellen Caches ist.Ab PHP Version 5.4 wird er in PHP schon enthalten sein. Die Installation von APC ist auf einigermaßen aktuellen Linux Servern auch denkbar einfach. Das PHP Pear Framework, was oft schon installiert ist, erlaubt es mit einigen wenigen Befehlen einen funktionierenden Opcode Cache zu installieren.

Listing 2: Installation von APC

```
1 apt-get install php-pear
2 pecl install apc
```

Nach der Installation wurde die Antwortzeit des Apacheservers gemessen und es kam zu folgendem Ergebnis. Statt 363 ms wurden nur noch 233 ms benötigt, eine Verbesserung um 36% und es können 4,28 Anfragen je Sekunde bearbeitet werden gegenüber vorher nur 2,75. Alles unter der Prämisse das es keine multiplen Anfragen gibt.

Listing 3: Ausgabe von ab

```
1  Server Software:        Apache/2.2.16
2  Server Hostname:        itsax.it-jobs-und-stellen.de
3  Server Port:            80
4
5  Document Path:          /
6  Document Length:        64842 bytes
7
8  Concurrency Level:      1
9  Time taken for tests:   11.681 seconds
10 Complete requests:      50
11
12 Write errors:           0
13 Total transferred:      3261730 bytes
14 HTML transferred:       3234230 bytes
15 Requests per second:    4.28 [\#/sec] (mean)
16 Time per request:       233.620 [ms] (mean)
17 Time per request:       233.620 [ms] (mean, across all concurrent
       requests)
18 Transfer rate:          272.69 [kBytes/sec] received
```

9.2. Drupal Cache

Listing 4: Ausgabe von ab

```
1  Server Software:        Apache/2.2.16
2  Server Hostname:        itsax.it-jobs-und-stellen.de
3  Server Port:            80
4
5  Document Path:          /
6  Document Length:        65005 bytes
7
8  Concurrency Level:      1
9  Time taken for tests:   2.082 seconds
10 Complete requests:      50
11 Failed requests:        0
12 Write errors:           0
13 Total transferred:      3276900 bytes
14 HTML transferred:       3250250 bytes
15 Requests per second:    24.02 [\#/sec] (mean)
16 Time per request:       41.638 [ms] (mean)
17 Time per request:       41.638 [ms] (mean, across all concurrent requests
       )
18 Transfer rate:          1537.09 [kBytes/sec] received
```

9.3. JS Aggregation und Minifizierung mit dem Javascript Aggregation Modul

Aggregation und Minifizierung sind Verfahren, die in der Webperformance Optimierung häufig eingesetzt werden. Für das Drupal 5 System gibt es fertige Module die diese Aufgabe übernehmen. Das Modul interveniert innerhalb des Drupal Kerns und ersetzt die Javascripts, die vorher direkt so ausgegeben wurden wie sie die Module lieferten, durch eine einzelne, das heisst aggregierte Version, die optional minifiziert werden kann. Diese Minifizierung wird natürlich genutzt und spart einige kB. Minifizierung wird ermöglicht durch die Nutzung von JSmin https://github.com/rgrove/jsmin-php/. JSmin ist ein Filter der unter anderem Kommentare entfernt und mehrere Leerzeichen zu einem zusammenfasst. Durch die Nutzung dieser Methode konnten 11 HTTP Abfragen und 11 kB an Datenvolumen eingespart werden. Load Time: 3.658s Time to First Byte: 0.595s Time to Start Render: 1.915s #DOM Elements: 844 #Requests: 95 Bytes In: 432 kB

9.4. CSS Aggregation und Komprimierung

Analog zur Javascript Optimierung kann man auch die CSS Aggregation betrachten. Dabei werden durch Zusammenfassung der einzelnen CSS Dateien HTTP Abfragen eingespart. Wie man an der gesunkenen Anzahl der Requests sehen kann, wurden 21 Abfragen nur durch Zusammenfügen der einzelnen CSS Dateien zu einer einzigen eingespart. Durch die Bereinigung beziehungsweise Komprimierung werden dabei außerdem 17 kB an überflüssigen Zeichen entfernt. Load Time: 3.577s Time to First Byte: 0.649s Time to Start Render: 1.577s #DOM Elements: 834 #Requests: 85 Bytes In: 427 kB

9.5. Drupal Boost Module

Das Boost Modul ist ein Cache für statische Seiten, er funktioniert aber nur für Gastnutzer der Webseite. ITsax.de wird aber fast ausschließlich von Gastnutzern besucht, da nur Partner und Administratoren einen erweiterten Zugang benötigen. Durch diese guten Vorraussetzungen kann das Boost Modul komplette Seiten cachen. Dabei ist HTML, XML, Ajax, CSS und Javascript caching und gzipping möglich. Die genutzen Techniken sind sehr performant und bauen auf einem Dateisystemcache auf, das bedeutet jede Seite wird komplett auf der Festplatte abgelegt. Alle Arten von serverseitigen Prozessen werden so nach dem initialem Seitenaufruf, der das Caching auslöst, nicht mehr durchlaufen. Dies sieht man sehr gut an der Time to First Byte. Von den 172 ms werden ca. 50ms für den DNS Lookup benötigt und 70ms für die erste HTTP Verbindung. Der Server braucht demnach nur ca. 50 ms um die Seite auszuliefern. Dies hat natürlich einen sehr Positiven Einfluss auf die Gesamtperformance und macht sich besonders beim Endnutzer bemerkbar, da die Seite spürbar schneller anfängt sich im Browser aufzubauen. Load Time: 3.233s Time to First Byte: 0.172s Time to Start Render: 1.473s #DOM Elements: 856 #Requests: 106 Bytes In: 444 kB

9.6. Bildkompression mit jpegoptim und OptiPNG

Bilder und Grafiken bieten oft großen Optimierungsspielraum. Zum einen erfolgt dies durch die richtige Auswahl der Dateiformate und zum anderen durch Komprimierung der Bilder. Da es auf www.itsax.de nicht nur statische Inhalte gibt, sondern auch durch Communitymitglieder und Communitymanager eingestellte Inhalte verwaltet werden müssen, sollte eine nachträgliche Qualitätsoptimierung der hochgeladenen Bilder durchgeführt werden. Um dies umzusetzen, sind die Programme OptiPNG und

jpegOptim zu empfehlen. Da es sich bei beiden Programmen um Kommandozeilenprogramme handelt, kann man ihre Anwendung leicht automatisieren. Mit dem Linuxbefehl find, der praktischerweise eine Möglichkeit, Befehle auszuführen, besitzt, kann man direkt die entsprechenden Dateien an die Optimierer übergeben. Diese Aktionen können dann über einen Cronjob periodisch jede Nacht ausgeführt werden.

9.6.1. Verlustfreie Kompression

Technisch wird eine verlustfreie Kompression durch Verfahren wie die Huffman-Codierung oder die Lauflängenkodierung umgesetzt. Im Fall des PNG Formats wird die Komprimierungsmethode Deflate genutzt. Außerdem werden Vorfilter, in Form von Prädikativer Kodierung, eingesetzt. Diese berechnen die wahrscheinlichen Farbwerte und es müssen nur die Abweichungen gespeichert werden.

Die Befehle sehen dann wie folgt aus:

Listing 5: Optimieren mit find

```
1 find . -name "*.png" -exec optipng -o7 {} \;
2 find . -name "*.jpg" -exec jpegoptim {} \;
```

Auf jeden Fall sollte eine verlustfreie Komprimierung durchgeführt werden, da die Bilder in diesem Fall nur an Dateigröße verlieren und die Bildqualität unberührt bleibt. Load Time: 3.640 Time to First Byte: 0.636s Time to Start Render: 1.894s #DOM Elements: 855 #Requests: 106 Bytes In: 429 kB

9.6.2. Verlustbehaftete Kompression

Da im Web kein besonders hoher Detailgrad gewährleistet werden muss, ist auch eine Kompression erwünscht die auf Kosten der Bildqualität die Bildgröße verringert.

find . -name ”*.pngexec optipng -o7 find . -name ”*.jpgexec jpegoptim

Load Time: 3.255s Time to First Byte: 0.669s Time to Start Render: 1.908s #DOM Elements: 855 #Requests: 106 Bytes In: 371 kB

9.7. Drupal 5 Fehler bei umgefärbten Themes

Das Framework Drupal 5 benutzt Themes zur Gestaltung der Oberfläche. Um diese farblich anpassen zu können, wurde das Color Modul installiert, welches Themeveränderungen ermöglicht. Aufgrund der Tatsache, dass das Theme nur kopiert wird und im Anschluss die Farben geändert werden, entstehen bei diesem Vorgang unnötige Duplikate, die beim Laden der Seite mitgeschleppt werden. Um diese zu entfernen, wird

einfach das Standardtheme durch das Modifizierte ersetzt. Dafür müssen nur noch einige Pfade in der style.css angepasst werden und man spart in dem Fall von itsax.de 4 kB, was immerhin ca 1% der übertragenen Datenmenge ausmacht. händisch gemergte styles style auf standard setzen vorher die images und das geänderte stylesheet kopieren Load Time: 3.626s Time to First Byte: 0.629s Time to Start Render: 1.890s #DOM Elements: 854 #Requests: 104 Bytes In: 440 kB

9.8. Theme Bilder Spriten

Spriting wurde ursprünglich in der Videospielentwicklung verwendet, um Bilder in den Grafikspeicher zu laden. In der Webentwicklung ist es eine effektive Technik, um Bilder ohne mehrfachen Overhead zu laden. Beim Spriting wird aus vielen einzelnen Bildern ein einziges Bild erstellt, dass anstelle der vielen Bilder geladen wird. Um die Bilder dann noch einzeln Anzeigen zu können, werden CSS Befehle genutzt, die es ermöglichen, die Größe und die Position eines Bildausschnittes anzuzeigen. Load Time: 3.707 Time to First Byte: 0.669s Time to Start Render: 1.968s #DOM Elements: 855 #Requests: 103 Bytes In: 443 kB

9.9. Module von Startseite entfernen

Um zu überprüfen, welchen Einfluss verschiedene, im Netzwerkgraphen auffällige, Module auf die Gesamtperformance haben, werden sie testweise komplett deaktiviert. So kann man entscheiden, bei welchen Modulen zusätzlicher Programmieraufwand lohnenswert ist.

9.9.1. Partnerslideshow

Das Deaktivieren der Partnerslideshow hat gravierenden Einfluss auf die Gesamtperformance. Das Modul lädt im Aktiven zustand alle Bilder die es benötigt und bremst damit den gesamten Ladevorgang aus. Die Load Time verringert sich um 27,7%. Der größte Teil des Geschwindigkeitszuwachses ist der Verringerung der Übertragungsmenge zuzuschreiben. Load Time: 2.695s Time to First Byte: 0.636s Time to Start Render: 1.838s #DOM Elements: 790 #Requests: 80 Bytes In: 276 kB

9.9.2. Facebook Widget

In diesem Fall ist der Einfluss auf die Performance geringer, aber mit einer Verbesserung von 6,7% auf jeden Fall vorhanden. Das Widget besteht aus neun kleinen Fotos

und dem Facebookrahmen. Den größten Effekt hat hier die Verringerung der Anzahl der HTTP Anfragen. So können wichtigere Elemente schneller geladen werden. Load Time: 3.478s Time to First Byte: 0.668s Time to Start Render: 1.966s #DOM Elements: 779 #Requests: 95 Bytes In: 406 kB

9.9.3. Jobleiste deaktiviert

Die Jobleiste hat einen geringfügig größeren Einfluss auf die Ladezeiten als das Facebook Widget. Die Ursache ist wahrscheinlich in den zusätzlichen Bibliotheken zu suchen, die zu diesem Modul gehören. Darunter sind Ajax Bibliotheken und anderer ThirdParty Code. Load Time: 3.367s Time to First Byte: 0.617s Time to Start Render: 1.709s #DOM Elements: 822 #Requests: 94 Bytes In: 411 kB

9.10. Umprogrammierung der Module

Um die langsamen Module weiterhin nutzen zu können, muss eine Lösung gefunden werden, die es ermöglicht, Inhalte nachzuladen, nachdem die Seite komplett geladen wurde. Um diese Asynchronität zu erreichen, wird mit Hilfe eines Timeout-Befehls, das Laden der nicht priorisierten Inhalte verzögert. Programmiert wird dieses Verhalten mit Javascript, genauer JQuery. Besonders betrachtet werden muss dabei, ob eine vorhandene Dynamik erhalten bleiben soll. Dazu gehören zum Beispiel zufällige Elemente oder Elemente mit besonders häufigen Aktualisierungen.

9.10.1. Partnerslideshow

Die Partnerslideshow Load Time: 2.749s (3.794) Time to First Byte: 0.626s Time to Start Render: 1.842s #DOM Elements: 855 #Requests: 82 (107) Bytes In: 284 kB (395)

9.10.2. facebook fenster

Load Time: 3.411s (4.258) Time to First Byte: 0.576s Time to Start Render: 1.848s #DOM Elements: 857 #Requests: 95 (106) Bytes In: 406 kB (444)

9.10.3. Partneranzeigen

Load Time: 3.871s (5.759s) Time to First Byte: 0.669s Time to Start Render: 2.255s #DOM Elements: 857 #Requests: 105 (108) Bytes In: 417 kB (448)

10. Endergebnis

Die Ladezeiten haben sich gravierend Verbessert, weniger als die Hälfte der Zeit wird benötigt bis die Seite benutzbar ist.

Load Time: 1.354s (2.913s) Time to First Byte: 0.173s Time to Start Render: 0.952s #DOM Elements: 789 #Requests: 25 (67) Bytes In: 138 kB (314)