

I. Inhaltsverzeichnis

I.	Inhaltsverzeichnis	II
II.	Tabellenverzeichnis	IV
III.	Abbildungsverzeichnis	V
IV.	Glossar	VI
V.	Abkürzungsverzeichnis	VII
1.	Einleitung.....	8
1.1.	Informationen im Internet.....	8
1.2.	Webshop Geschwindigkeit beeinflusst Kaufinteresse.....	8
1.3.	Zielsetzung	9
2.	Content Distribution Networks.....	10
2.1.	Allgemeines.....	10
2.2.	Problem der zentralen Server.....	10
2.3.	Aufbau.....	11
2.4.	Funktionsweise	13
2.5.	Arten von Daten.....	14
2.6.	Wirtschaftliche Vorteile	14
2.7.	Kosten	15
2.8.	Amazon als wichtiger Anbieter	17
2.8.1.	Amazon Simple Storage Service.....	18
2.8.2.	Amazon CloudFront	18
3.	Überblick über die verwendeten Programme und Systeme	22
3.1.	Die Online-Shop Software Magento	22
3.2.	WAPT - Web Application Testing Tool	22
3.3.	OnePica ImageCDN - Magento Extension	23
4.	Einsatz von Amazon CloudFront in Magento.....	24
4.1.	Testumgebung.....	24
4.2.	Testparameter	25
4.3.	Performancetests	25
4.3.1.	Ausgangswerte.....	25
4.3.2.	Magento Konfiguration Javascript und Skin-Ordner.....	25

4.3.3.	Auslagerung vom Skin-Ordner und Javascript.....	27
4.3.4.	Konfiguration Modul - ImageCDN	28
4.3.5.	Auslagerung der Produktbilder.....	30
4.3.6.	Auslagerung der Produktbilder und Javascript	31
4.4.	Auswertung	31
5.	Zusammenfassung und Fazit.....	33
	Literaturverzeichnis.....	34
	Ehrenwörtliche Erklärung	36

II. Tabellenverzeichnis

Tabelle 1: Amazon CloudFront Gebühren für ausgehende Datenübertragung	16
Tabelle 2: Amazon CloudFront Gebühren für Anfragen	16
Tabelle 3: Amazon CloudFront Architektur Schritte	20
Tabelle 4: Performancetest - Ausgangswerte.....	25
Tabelle 5: Performancetest - Auslagerung Javascript, Skin-Ordner Ergebnisse	28
Tabelle 6: Performancetest - Auslagerung Produktbilder Ergebnis.....	31
Tabelle 7: Performancetest - Auslagerung Produktbilder, Javascript Ergebnis	31
Tabelle 8: Vergleich der Ergebnisse aus dem Performancetest.....	32

III. Abbildungsverzeichnis

Abbildung 1: Zentraler Server.....	11
Abbildung 2: CDN - Aufbau	12
Abbildung 3: Top Ten Anbieter im Segment Cloud Infrastructure	17
Abbildung 4: AWS Managment Console	19
Abbildung 5: Amazon CloudFront Architektur	20
Abbildung 6: Magento Demo-Shop	24
Abbildung 7: Magento Konfiguration URL ´s	26
Abbildung 8: Magento Demo - Firebug Requests	27
Abbildung 9: Konfiguration ImageCDN Allgemein	29
Abbildung 10: Konfiguration ImageCDN Amazon Zugangsdaten.....	30

IV. Glossar

Begriff	Erklärung
Apache	Webserver
Conversion-Rate	Ist eine Maßeinheit, welche die Relation von Anzahl der Besucher zur Anzahl gewünschter Aktionen (z.B. Kauf) misst.
Distribution	Zusammenstellung von Dateien, die als Komplettpaket abgebildet werden
Frontend	Bereich einer Webseite, der für die Öffentlichkeit frei zugänglich ist
Latenzzeit	Zeitraum zwischen einer Aktion und dem Eintreten einer verzögerten Reaktion
Usability	Benutzerfreundlichkeit des Designs einer Webseite
SSH	Netzwerkprotokoll, mit dem man eine sichere verschlüsselte Netzwerkverbindung herstellt

V. Abkürzungsverzeichnis

Abkürzung	Langschreibweise
API	Application Programming Interface
FTP	File Transfer Protocol
SFTP	Secure File Transfer Protocol

1. Einleitung

1.1. Informationen im Internet

Heutzutage ist das Internet eines der wichtigsten Systeme, um nach Informationen zu suchen und diese schnell abzurufen. Diese Informationen können Webseiten, Bilder, Textdokumente, Software oder auch Multimediainhalte sein. Im Allgemeinen wird auch von Inhalt (engl. Content) gesprochen. Da das Internet jedem Nutzer zugänglich ist gibt es eine steigende Anzahl von Benutzern, die immer mehr Daten abrufen. Auch die Anzahl der Breitbandangebote bzw. -inhalte nimmt immer mehr zu. So kommt es jetzt schon zu hohen Zugriffszeiten und Überlastungen der Server und Netze.¹

1.2. Webshop Geschwindigkeit beeinflusst Kaufinteresse

Es gibt eine Vielzahl von Faktoren die den Erfolg eines Onlineshop beeinflussen. Dazu zählen selbstverständlich Preise und Versandkosten, aber auch solche Faktoren wie Präsentation, Usability und die Geschwindigkeit des Shops.

„Die Geschwindigkeit einer Website hat großen Einfluss auf die Zufriedenheit des Nutzers. Das Internet präsentiert ein nahezu grenzenloses Angebot von Information und Unterhaltung. Wer nicht schnell genug anbietet verliert den gegenwärtigen Interessenten an einen Konkurrenten. Ein weiterer Effekt, der langfristig schwerwiegende Folgen verzeichnet: Das Markenbild leidet erheblich. Ein langsames und nicht zufrieden stellendes Online-Angebot vermittelt auf Dauer den Eindruck mangelnder Qualität. Die ästhetische Wahrnehmung leidet genauso wie die Glaubwürdigkeit des Angebots. Besonders neue Nutzer, die eine Website zum ersten Mal aufrufen, versuchen ihr Glück bei der ersten Enttäuschung nicht erneut. Sie bevorzugen das Angebot der Konkurrenz. Die Conversion-Rate sinkt, virale Effekte bleiben sowieso aus und der Nutzer behält nur eines in Erinnerung: Frustration.“²

¹ [Sie04] S. 2

² [Kie10]

Die dotSource GmbH ist eine etablierte E-Commerce Agentur und hat es sich zur Aufgabe gemacht, anspruchsvolle E-Commerce-Plattformen nach Kundenwünschen zu entwickeln. Dabei wird u.a. auf Basis der Open Source Onlineshop-Software Magento gearbeitet und durch verschiedene Erweiterungen eine Kundenbindung erreicht. Die Leistungen erstrecken sich von der Entwicklung und dem Design von individuellen Onlineshops bis hin zum Hosting und Betrieb der einzelnen Seiten. Deshalb ist es wichtig, als Full-Service-Internetagentur entsprechende Trends und Lösungen anzubieten.

1.3. Zielsetzung

Das Ziel dieser Projektarbeit besteht darin, mit der Hilfe eines Content Distribution Networks dem Problem der Performance eines Webshops entgegenzuwirken. Dazu wird im ersten Teil der Arbeit auf den genauen Aufbau und der Funktionsweise eines Content Distribution Networks eingegangen. Dabei wird auf den wichtigsten Anbieter in diesem Bereich näher eingegangen. Im zweiten Teil wird anhand des theoretischen Wissens, die praktische Implementierung in einem Magento-Shopsystem dokumentiert. Um die eventuell erreichten Geschwindigkeitsverbesserungen unter realen Bedingungen zu analysieren und auszuwerten, werden anhand eines Beispielshops unterschiedliche Tests vorgenommen.

2. Content Distribution Networks

2.1. Allgemeines

Ein Content Distribution Network (CDN), oder auch Content Delivery Network genannt, ist für die Auslieferung meist großer statischer Objekte an die jeweiligen Endnutzer verantwortlich. Der Arbeitsablauf beschränkt sich nicht nur auf die Auslieferung, sondern umfasst ebenso die Behandlung der Inhalte. Somit werden auch die Speicherung und die Bereitstellung geregelt.³

Ein Content Distribution Network besteht aus vier verschiedenen Komponenten⁴:

- Bereitstellung von Daten (Distribution Infrastructure)
- Bearbeitung der Anfragen (Request routing)
- Lieferung des Inhalts (Content Delivery)
- Abrechnungssysteme (Account Infrastructure)

Eine detaillierte Betrachtung der einzelnen Komponenten ist nicht Bestandteil dieser Arbeit. Im nachfolgenden Abschnitt werden der allgemeine Aufbau und die Funktionsweise im Gesamten betrachtet.

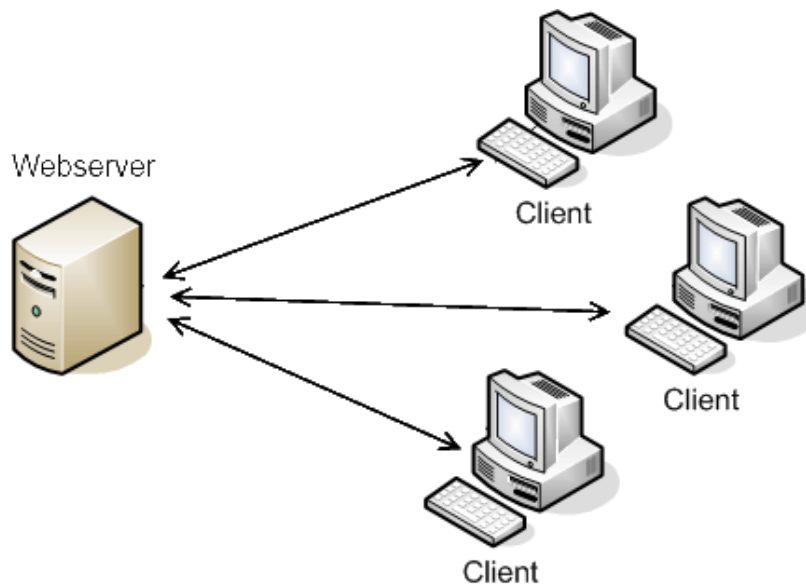
2.2. Problem der zentralen Server

Bei den meisten Anbietern wird der Inhalt von einem zentralen Server aus bereitgestellt. Wie in Abbildung 1: Zentraler Server Abbildung 1 dargestellt, können die Clients ihre Anfragen nur an den einen Server stellen, was unter Umständen zu Problemen führen kann.

³ [Sie04] S. 4

⁴ [Göt04] S. 146

Abbildung 1: Zentraler Server



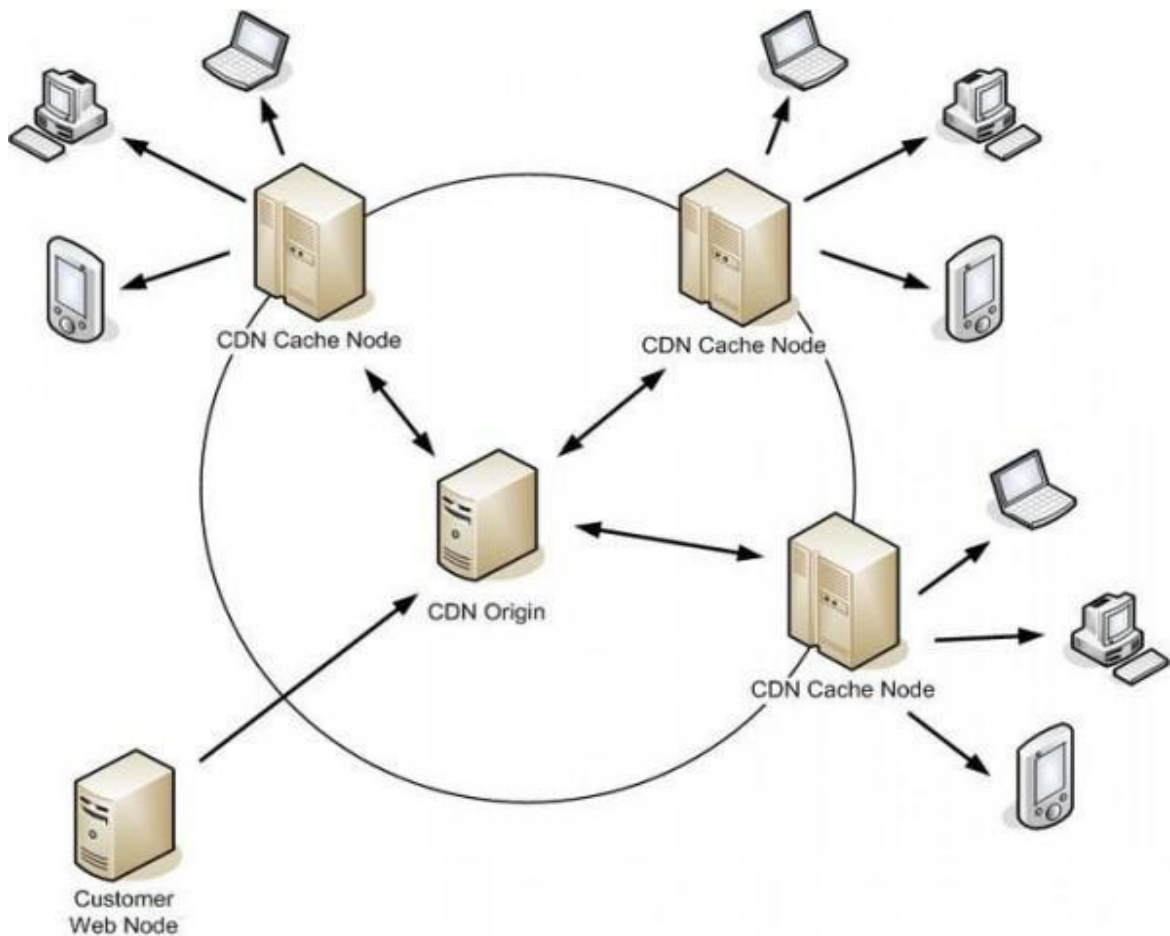
Quelle: eigene Darstellung

Ein Problem kann schon bei höheren Zugriffszahlen auftreten. Wenn mehrere Tausend Clients zur gleichen Zeit versuchen eine Musikdatei herunterzuladen, könnte der Server einen Teil dieser Anfragen nicht mehr bewältigen. Die Folge wäre eine Überlastung des Servers, was dazu führen kann das dieser keine einzige Anfrage mehr bearbeiten kann.

2.3. Aufbau

Damit die Probleme, die durch nur einen zentralen Server auftreten können vermieden werden, setzen CDNs genau an diesem Punkt an. Ein CDN ist wie ein lokales Netzwerk mit verteilten Servern aufgebaut. Es besteht aus einem Ursprungsserver (CDN Origin) und mehreren Cache-Servern (CDN Cache Node). Die folgende Abbildung 2 stellt die typische Anordnung eines CDNs dar.

Abbildung 2: CDN - Aufbau



Quelle: [FK10]

Die Grundlage eines jeden CDNs ist der Ursprungsserver. Auf diesem speichern die Anbieter ihre Daten, in der Abbildung 2 als *Customer Web Node* bezeichnet.⁵ Die Inhalte können vom Anbieter meist über das File Transfer Protocol (FTP) oder über Secure Shell (SSH) hochgeladen werden.

Durch diese Caching-Struktur stehen die Inhalte näher am Client zum Ausliefern bereit. Niedrigere Latenzzeiten, bessere Erreichbarkeit und eine geringere Last in den Netzverbindungen sind die Vorteile die diese spezielle Anordnung mit sich bringt. Eine vergrößerte Anzahl der Server sorgt für eine Verbreitung der Last. Das heißt, nicht mehr ein einzelner Server muss eine massive Last aushalten, sondern

⁵ [Pet11]

die Last wird stattdessen auf mehrere Server verteilt. Diese Verteilung wird auch als *Burst-Protection* bezeichnet.⁶

2.4. Funktionsweise

Sendet der Client eine Anfrage für eine bestimmte Datei, so wird diese vom Request-Routing-System entgegengenommen. Dieses System trifft die Auswahl eines geeigneten Cache-Servers, zu dem die Anfrage weitergeleitet wird. Das Request-Routing-System kann anhand verschiedener Parameter entscheiden:

- aktuelle Auslastung
- gleichzeitig aktiver Verbindungen
- geografische Gegebenheiten

Es könnte aber auch anhand der Identität entschieden werden. Als Beispiel ist ein Anbieter von Premium-Inhalten zu nennen. Dieser muss seine einzelnen Nutzer zwischen Standard- und Premium-Nutzern unterscheiden. Anhand dieser Merkmale entscheidet das Request-Routing-System selbstständig, welcher Cache-Server am schnellsten antworten kann.⁷

Wird die Anfrage nun an einen bestimmten Cache-Server weitergeleitet, prüft Dieser ob er die angeforderte Datei bereits besitzt. Falls ja, überträgt er die Daten sofort an den Client, dabei kann die Übertragung unverschlüsselt oder mit einer SSL-Verschlüsselung übertragen werden. Hat der Server die Datei noch nicht bei sich gespeichert, leitet er die Anfrage an den Ursprungsserver weiter. Der Ursprungsserver liefert die Datei an den Client aus und leitet gleichzeitig die neue Datei an den Cache-Server weiter. Dieser hat nun die zuvor fehlende Datei gespeichert und kann bei einer erneuten Anfrage den Client direkt bedienen. Somit ist keine weitere Anfrage an den Ursprungsserver mehr nötig.⁸

⁶ [Sie04] S. 4

⁷ [Pet11]

⁸ [FK10]

2.5. Arten von Daten

CDN können vielseitig zum Ausliefern von Daten verwendet werden. Dabei wird meist zwischen zwei Arten unterschieden: Live und On Demand.

Live-Inhalte können z.B. Sportveranstaltungen oder auch verschiedene Konzerte sein. Diese Inhalte werden direkt an die Clients über die Server übertragen. Bei On Demand Inhalten sieht die Auslieferung etwas anders aus, dort werden die Dateien auf einem Server abgelegt und nur auf Anfrage eines Clients ausgeliefert.⁹

2.6. Wirtschaftliche Vorteile

Wie schon in der Einleitung unter Punkt 1.2 angesprochen, kann eine Verbesserung der Geschwindigkeit zu einer positiven Veränderung des Kaufinteresse führen. „Die Toleranz der Nutzer, auf die Ladezeit einer Website zu warten, ist mehr als begrenzt. Google fand gemeinsam mit Bing innerhalb einer Studie heraus, dass bei der Ladezeit einer Website oft Millisekunden über ihren weiteren Erfolg entscheiden. Selbst minimale Anpassungen der Antwortzeiten führten zu signifikanten Veränderungen.“¹⁰

Bei einer Studie kam u.a. heraus, dass schon eine Verlangsamung der Suchergebnisseite von Bing von nur einer Sekunde dazu geführt hat, dass ein Prozent weniger Suchanfragen stattfanden und somit 1,5 Prozent weniger Werbeklicks erfolgten. Ein weiterer Test vom Onlinehändler Amazon ergab auch verblüffende Ergebnisse. Bei dieser Untersuchung wurde herausgefunden, dass eine Verbesserung der Geschwindigkeit von nur 100 Millisekunden zu einer Verkaufssteigerung von einem Prozent geführt hat.¹¹

Auf Grund dieser Ergebnisse kann es sich für Betreiber eines Onlineshops rentieren mit Hilfe eines CDN die Geschwindigkeit zu optimieren. Denn somit können z.B.

⁹ [Sie04] S. 3

¹⁰ [Kie10]

¹¹ [Kie10]

Betreiber große Mengen an Produktbildern ihren Kunden zur Verfügung stellen, da diese in voller Bandbreite für die Kunden nutzbar sind.

„Durch die Geschwindigkeit, mit der Content sichtbar wird und den großen Umfang erfolgt unzweifelhaft auch eine messbare Kundenbindung an die Webseite. Entfallende Wartezeiten und schneller Seitenaufbau generieren ein hohes Maß an Kundenzufriedenheit.“¹²

2.7. Kosten

Bis vor einiger Zeit konnten sich eigentlich nur große Unternehmen ein CDN leisten, da es nur teure Spitzenprodukte von wenigen Anbietern gab z.B. Level3 Communications. Doch mit Amazon Web Services und Google Appspot wurden echte Alternativen geschaffen. Diese richten sich vor allem an professionelle Entwickler aber auch kleinere Firmen.

Das typische Abrechnungsverfahren ist die Vergütung in Abhängigkeit der Nutzungsintensität: „Pay as you use“. Es muss nur für das bezahlt werden, was tatsächlich verbraucht wurde.

Ein gutes Beispiel für dieses Abrechnungsverfahren, ist der Service CloudFront von Amazon. In den folgenden Tabellen sind die Gebühren aufgelistet, die bei der Nutzung anfallen. Dabei werden von Amazon, in den meisten Services, nur die Datenübertragungen und die Anfragen berechnet.

Tabelle 1 stellt die Kosten für die ausgehenden Datenübertragungen dar.

Amazon kann in unterschiedlichen Regionen Kosteneinsparungen erzielen, diese sollen an die Kunden weitergegeben werden. Deshalb werden bei der Berechnung der Kosten immer der jeweilige Standort des Servers berücksichtigt, von dem aus die Inhalte verbreitet werden.¹³

¹² [Pet11]

¹³ [Ama11a]

Tabelle 1: Amazon CloudFront Gebühren für ausgehende Datenübertragung

	USA	Europa	Hongkong & Singapur	Japan
Erste 10 TB pro Monat	\$0,120 / GB	\$0,120 / GB	\$0,190 / GB	\$0,201 / GB
Nächste 40 TB pro Monat	\$0,080 / GB	\$0,080 / GB	\$0,140 / GB	\$0,148 / GB
Nächste 100 TB pro Monat	\$0,060 / GB	\$0,060 / GB	\$0,120 / GB	\$0,127 / GB
Nächste 350 TB pro Monat	\$0,040 / GB	\$0,040 / GB	\$0,100 / GB	\$0,106 / GB
Nächste 524 TB pro Monat	\$0,030 / GB	\$0,030 / GB	\$0,080 / GB	\$0,085 / GB
Nächste 4 PB pro Monat	\$0,025 / GB	\$0,025 / GB	\$0,070 / GB	\$0,075 / GB
Über 5 PB pro Monat	\$0,020 / GB	\$0,020 / GB	\$0,060 / GB	\$0,065 / GB

Quelle: [Ama11a]

Da auch einzelne Anfragen an den Server Last und Rechenzeit verbrauchen, werden diese extra abgerechnet. In Tabelle 2 können die Kosten pro 10.000 Anfragen abgelesen werden.

Tabelle 2: Amazon CloudFront Gebühren für Anfragen

	USA	Europa	Hongkong & Singapur	Japan
Pro 10.000 HTTP-Anfragen	\$0,0075	\$0,0090	\$0,0090	\$0,0095
Pro 10.000 HTTPS-Anfragen	\$0,0100	\$0,0120	\$0,0120	\$0,0130

Quelle: [Ama11a]

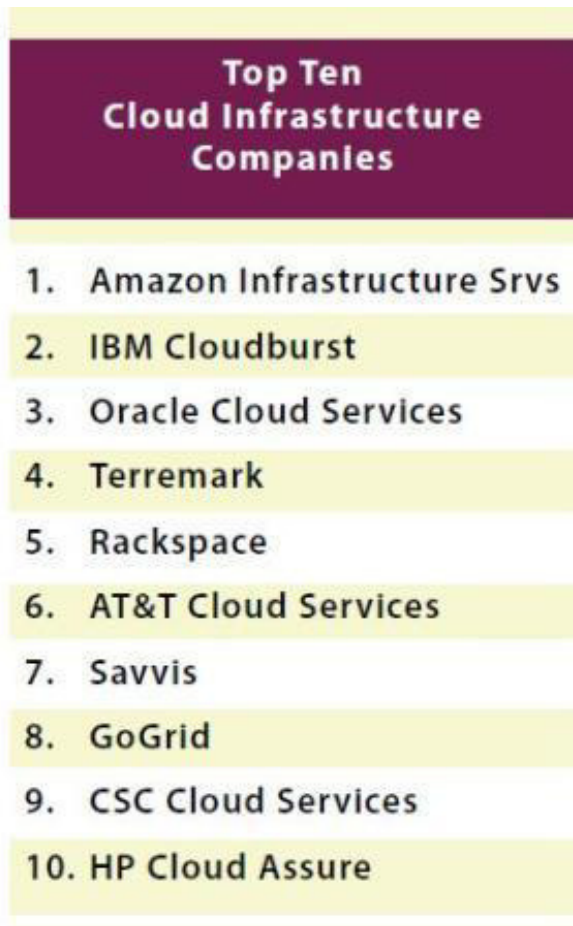
Des Weiteren fallen keine Einrichtungs- oder Bearbeitungsgebühren an.

2.8. Amazon als wichtiger Anbieter

Da sich das Cloud-Computing zu einem starken Trend entwickelt hat, gibt es mittlerweile viele verschiedene Anbieter. Diese können sich aber stark durch Angebot und Leistung ihrer Netze und Rechenzentren unterscheiden.

Laut einer Studie von BTC Logic sind die best aufgestellten Anbieter Amazon und IBM. Amazon belegte den ersten Platz in der Kategorie *Cloud Infrastructure*, wie in Abbildung 3 zu sehen. In dieser Kategorie wurden alle Anbieter gelistet, die Cloud Infrastrukturen als skalierbare, sichere und kosteneffiziente Lösung anbieten.¹⁴

Abbildung 3: Top Ten Anbieter im Segment Cloud Infrastructure



Quelle: [Eri11] S. 1

¹⁴ [Eri11] S.1

Nach Ansicht des Autors dieser Arbeit, kann Amazon Web Services sehr gut als ersten Einstieg in den Bereich Cloud Computing und Content Distribution genutzt werden. Nach nur einer kurzen Anmeldung steht das komplette System für einen ersten Test bereit. Zudem können verschiedene Service im ersten Jahr kostenlos genutzt werden.

2.8.1. Amazon Simple Storage Service

Da Amazon den Service S3 (Simple Storage Service) als Grundlage für viele weitere Angebote nutzt, sollte dieser kurz betrachtet werden.

Amazon S3 ist ein Speicher im Internet. In Diesem ist es möglich eine beliebige Anzahl an Objekten abzuspeichern und diese von jedem Ort im Internet aus wieder abzurufen.

Alle Objekte werden in einem so genannten Bucket gespeichert. Ein Bucket kann man mit einem handelsüblichen Ordner auf dem Computer vergleichen. Für jeden Bucket kann separat eine Region ausgewählt werden. Die Region gibt an wo die Server zum Speichern der Daten stehen, so kann z.B. die Verzögerungszeit der Anfragen optimiert werden. Dieser Service wurde bewusst mit nur einem Minimum an Funktionen ausgestattet, um ihn einfach und intuitiv benutzen zu können. Deshalb ist nur das Speichern, Lesen und Löschen von Dateien möglich. Die Anzahl der Dateien ist unbeschränkt, nur die Größe muss sich im Bereich von 1 Byte bis 5 Terabyte bewegen.¹⁵

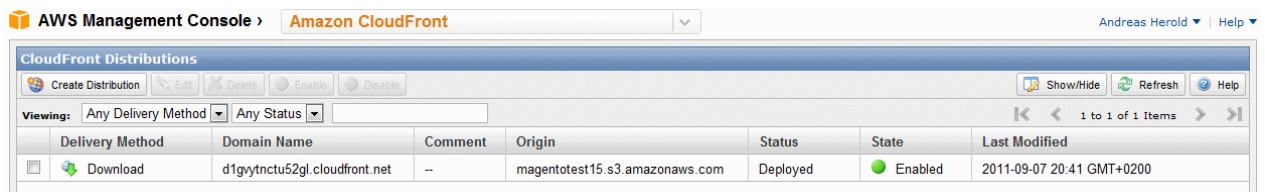
2.8.2. Amazon CloudFront

Amazon stellt für alle Funktionen ein Application Programming Interface (API) bereit, die für die Verwaltung der Dateien und Ordner verwendet werden kann. Für den einfachen Einstieg bietet Amazon aber auch eine Web-basierte Schnittstelle an, die über den Internet-Browser genutzt werden kann. Mit dieser *AWS Management Console* können die einzelnen Services von Amazon verwaltet werden. Abbildung 4 zeigt ein Screenshot von der Web-basierten Schnittstelle im Bereich

¹⁵ [Ama11c]

Amazon CloudFront. Dort kann mit wenigen Mausklicks eine neue Distribution angelegt werden.

Abbildung 4: AWS Management Console

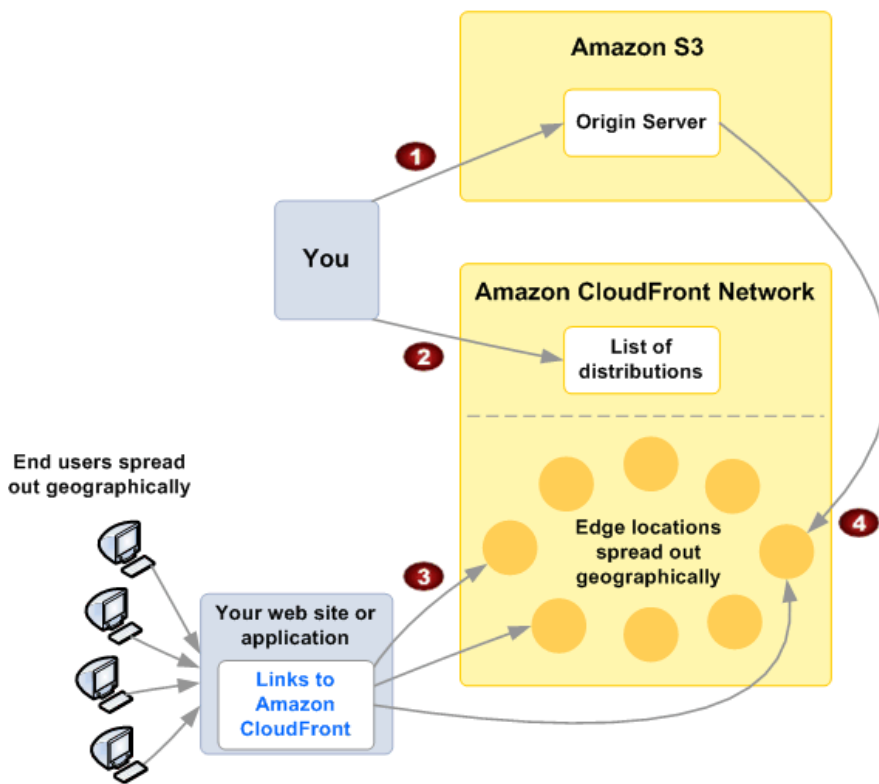


Quelle: <https://console.aws.amazon.com/>, eigener Screenshot

Um eine neue Distribution anlegen zu können, werden zuerst die zu verteilenden Inhalte benötigt. Diese müssen, wie unter Punkt 2.3 beschrieben, auf einem Ursprungsserver bereitgestellt werden. Amazon nutzt dazu ihren Service *Amazon S3* wie schon in Abschnitt 2.8.1 kurz beschrieben.

In der nachfolgenden Abbildung 5 ist die Architektur von Amazon CloudFront in Verbindung mit Amazon S3 dargestellt. Die einzelnen Schritte, die mit Zahlen in der Abbildung gekennzeichnet sind, werden in der Tabelle 3 ausführlich erklärt.

Abbildung 5: Amazon CloudFront Architektur



Quelle: [Ama11b] S.6

Tabelle 3: Amazon CloudFront Architektur Schritte

Schritt	Beschreibung
1	Die gewünschten Dateien werden auf dem Ursprungsserver verwaltet (Ablegen, Löschen). Bei CloudFront wird Amazon S3 als Ursprungsserver verwendet.
2	In CloudFront muss eine neue Distribution angelegt werden, dabei muss angegeben werden, welcher Bucket von S3 genutzt werden soll. Jede Distribution bekommt eine eigene URL, die zum Abrufen der Dateien verwendet werden muss.
3	Der CloudFront Link wird auf der Webseite eingebunden bzw. an die Endnutzer verteilt. CloudFront liefert die Dateien aus, nicht Amazon S3! Die Endnutzer werden anhand der geografischen Gegebenheiten an den am nächsten gelegenen Edge-Server weitergeleitet.

4

Sollte der Server die Datei noch nicht besitzen, fordert er sie automatisch vom Ursprungsserver (Amazon S3) an.

Übersetzt und mit Änderungen entnommen aus: [Ama11b] S.7

3. Überblick über die verwendeten Programme und Systeme

3.1. Die Online-Shop Software Magento

Die erfolgreiche E-Commerce-Software Magento basiert auf der Skriptsprache PHP (Hypertext Preprocessor) und auf dem Zend Framework. Der Erfolg dieser Software ist dem modularen Aufbau und seinem flexiblen Datenmodell zuzuschreiben. Dadurch ist es möglich über ein „Baukastenprinzip“ das vorhandene System einfach zu erweitern und somit in die eigenen Prozesse zu integrieren. Durch die eigenen programmierten Erweiterungen, auch Extension genannt, bleibt das System weiterhin updatefähig, da keine Systemdateien überschrieben werden müssen.

Durch eine eigene Community werden sehr viele fertige Erweiterungen auf der Webseite kostenlos zur Verfügung gestellt. Diese können über den eingebauten „Magento-Connect Manager“ einfach installiert werden. Somit ist es möglich Funktionen nachzurüsten, die in der Standardinstallation nicht vorhanden wären.¹⁶

3.2. WAPT - Web Application Testing Tool

Wapt ist ein Programm mit dem Belastungs-, Stress- und Performance-Tests für Webseiten und Anwendungen mit Web-Interface möglich gemacht werden. Um reale Bedingungen für den Test einer Webseite zu schaffen, versucht das Programm verschiedene Nutzer zu simulieren. So wird eine Reihe von verschiedenen Techniken verwendet, um reale Bedingungen so genau wie möglich zu simulieren. Als Beispiel können diese virtuellen Besucher, unterschiedliche Seiten besuchen, automatisch Formulare ausfüllen, auf Schaltflächen klicken oder übernehmen andere übliche Tätigkeiten von realen Besuchern. Jeder virtueller Besucher hat dabei seine eigenen speziellen Parameter, wie z.B. Cookies, Namen oder auch Passwörter die verwendet werden sollen.¹⁷

Für jeden einzelnen Test können die Testparameter neu definiert werden, also die zeitliche Länge, Anzahl der virtuellen Besucher oder auch die Anzahl der Profildurchläufe. Dadurch ist es möglich mit einem erstellten Profil verschiedene

¹⁶ [Zen10], S. 9

¹⁷ [Wap11]

Konstellationen nacheinander testen zu können. Die Ergebnisse können später ausgewertet und miteinander verglichen werden.

3.3. OnePica ImageCDN - Magento Extension

OnePica ImageCDN ist ein kostenloses Community Modul von Magento. Es kann in einem Magento-Shop über den Extension-Key installiert werden.

Der Link zur Modul-Seite: <http://www.magentocommerce.com/magento-connect/One%20Pica/extension/1279/one-pica-image-cdn>

Diese Erweiterung macht es möglich, die in Magento hochgeladenen Produkt- und Kategoriebilder auf einem separaten Server zu speichern. Dazu sind auch einige CDN-Anbieter extra integriert, z.B. Amazon S3 und Coral CDN. Es kann aber auch jeder andere beliebige Anbieter verwendet werden, sobald er mindestens FTP oder Secure File Transfer Protocol (SFTP) unterstützt.

Sobald das Modul aktiviert und entsprechend konfiguriert wurde, werden alle Kategorie- und Produktbilder die im Frontend des Shops zu sehen sind über das jeweilige CDN ausgeliefert. Das betrifft auch die verkleinerten bzw. vergrößerten Cache Bilder, die sonst bei einem Aufruf auf dem eigenen Server generiert werden.

Der Vorteil dieser Erweiterung ist, dass keine sonstigen Einstellungen am Server gemacht werden müssen. Es wird komplett auf *Apache rewrites* verzichtet, da die Bilder direkt auf das CDN verlinkt sind. Auch an die Verwendung von SSL-Seiten wurde gedacht. So entscheidet das Modul automatisch ob die Bilder mit HTTP oder eben mit HTTPS angefordert werden müssen.

4. Einsatz von Amazon CloudFront in Magento

4.1. Testumgebung

Um herauszufinden inwieweit eine Performanceoptimierung durch Nutzung eines CDN möglich ist, sind vergleichbare Testergebnisse zur realen Bedingungen nötig. Aus diesem Grund wurden die Tests an einem Magento Demo-Shop, auf den Produktivservern der dotSource GmbH durchgeführt. Auf diesen Servern werden mehrere Kundenprojekte der Firma dotSource betrieben. Aus diesem Grund ist diese Umgebung für die nachfolgenden Tests sehr gut geeignet. Der extra dafür eingerichtete Shop ist eine Magento Standard-Installation mit den für diese Version bereitgestellten Beispiel-Daten. Es handelt sich dabei um die Magento Version 1.5.0.1. Durch die Beispiel-Daten sind alle typischen Konfigurationen und einige Kategorien und Produkte vorhanden. Die Startseite des Demo-Shops ist in der folgenden Abbildung 6 zu sehen.

Abbildung 6: Magento Demo-Shop

Quelle: eigener Screenshot

4.2. Testparameter

Alle Tests werden mit dem in Abschnitt 3.2 beschriebenen Tool durchgeführt. Nach jedem einzelnen Test erstellt WAPT einen ausführlichen Testbericht. Die Daten aus diesen Berichten werden in dieser Arbeit zu Vergleichszwecken verwendet.

Jeder Lasttest wird mit einer unterschiedlichen Anzahl von virtuellen Nutzern durchgeführt. Der Autor dieser Arbeit hat sich für 20, 50, 100 und 200 Nutzer entschieden. Diese bewegen sich im Zeitraum von 5 Minuten je Test auf der Webseite. Damit ein großer Bereich des Shops abgedeckt werden kann, werden die virtuellen Nutzer in 2 verschiedene Profile eingeteilt. Beide Profile klicken sich dabei durch den Shop und schauen sich Kategorien, Produkte und Produktbilder im Großformat an oder führen auch mal eine Suche nach einem Produkt durch. Während des Testes werden keine Bestellungen im Shop getätigt.

4.3. Performancetests

4.3.1. Ausgangswerte

Um Vergleichswerte zu erhalten wurde ein Test ohne Veränderungen und ohne Einsatz eines CDN durchgeführt. Alle CSS-, Javascript- und Bilddateien, darunter auch Produktbilder, befinden sich auf den gleichen Server wie die Magento-Installation.

Tabelle 4: Performancetest - Ausgangswerte

Anzahl der Nutzer	20	50	100	200
Ø Reaktionszeit (mit Elemente), Sek.	0,77	0,75	0,78	1,62
Ø Reaktionszeit, Sek.	0,60	0,58	0,61	1,31
Ø Seiten pro Sekunde	7	17,8	35	56,1
Ø Zugriffe pro Sekunde	30,9	78,3	154	258

4.3.2. Magento Konfiguration Javascript und Skin-Ordner

Die Konfiguration im Magento-Adminpanel ist durch wenig Aufwand zu realisieren. Es muss nur die CloudFront URL von Amazon für die beiden Verzeichnisse

eingestellt werden. Ein Beispiel für die korrekte Konfiguration ist in Abbildung 7 zu sehen. Bevor die URL's eingetragen werden sollte auf jeden Fall sichergestellt sein das sich alle Dateien auf dem Amazon Server befinden. Wenn neue hinzukommen werden diese auch dort abgelegt.

Abbildung 7: Magento Konfiguration URL's

Web

Url Options

Search Engines Optimization

Unsecure

Base URL	<input type="text" value="http://cdn-demo.dotsource.de/"/>	[STORE VIEW]
Base Link URL	<input type="text" value="{{unsecure_base_url}}"/>	[STORE VIEW]
Base Skin URL	<input type="text" value="http://d1gvytnctu52gl.cloudfront.net/skin/"/>	[STORE VIEW]
Base Media URL	<input type="text" value="{{unsecure_base_url}}media/"/>	[STORE VIEW]
Base JavaScript URL	<input type="text" value="http://d1gvytnctu52gl.cloudfront.net/js/"/>	[STORE VIEW]

▲ **Warning!** When using CDN, in some cases JavaScript may not run properly if CDN is not in your subdomain

Quelle: eigener Screenshot

Sobald die Konfiguration gespeichert wurde, sind die Änderungen aktiv und alle Dateien (CSS, Javascript usw.) werden nur noch von den CloudFront Servern geladen. Wie das bei den ganzen Requests aussieht, kann man in Abbildung 8 sehen. Es wird effektiv nur noch der HTML-Code der Seite und eventuell vorhandene Produktbilder vom eigenen Server geladen. Der Rest wird von den Amazon Servern angefordert.

Abbildung 8: Magento Demo - Firebug Requests

⊕ GET	200 OK	cdn-demo.dotsource.de	4.3 KB	213.174.43.23:80
⊕ GET styles.css	200 OK	d1gvytnctu52gl.cloudfront.net	88.9 KB	216.137.61.137:80
⊕ GET widgets.css	200 OK	d1gvytnctu52gl.cloudfront.net	4.9 KB	216.137.61.137:80
⊕ GET print.css	200 OK	d1gvytnctu52gl.cloudfront.net	1.6 KB	216.137.61.137:80
⊕ GET prototype.js	200 OK	d1gvytnctu52gl.cloudfront.net	126.9 KB	216.137.61.137:80
⊕ GET ccard.js	200 OK	d1gvytnctu52gl.cloudfront.net	747 B	216.137.61.137:80
⊕ GET validation.js	200 OK	d1gvytnctu52gl.cloudfront.net	36.5 KB	216.137.61.137:80
⊕ GET builder.js	200 OK	d1gvytnctu52gl.cloudfront.net	4.6 KB	216.137.61.137:80
⊕ GET effects.js	200 OK	d1gvytnctu52gl.cloudfront.net	37.8 KB	216.137.61.137:80
⊕ GET dragdrop.js	200 OK	d1gvytnctu52gl.cloudfront.net	30.5 KB	216.137.61.137:80
⊕ GET controls.js	200 OK	d1gvytnctu52gl.cloudfront.net	34 KB	216.137.61.137:80
⊕ GET slider.js	200 OK	d1gvytnctu52gl.cloudfront.net	10.1 KB	216.137.61.137:80
⊕ GET js.js	200 OK	d1gvytnctu52gl.cloudfront.net	18.3 KB	216.137.61.137:80
⊕ GET form.js	200 OK	d1gvytnctu52gl.cloudfront.net	11.4 KB	216.137.61.137:80
⊕ GET menu.js	200 OK	d1gvytnctu52gl.cloudfront.net	4.3 KB	216.137.61.137:80
⊕ GET translate.js	200 OK	d1gvytnctu52gl.cloudfront.net	1.6 KB	216.137.61.137:80
⊕ GET cookies.js	200 OK	d1gvytnctu52gl.cloudfront.net	2.6 KB	216.137.61.137:80
⊕ GET bkg_body.gif	200 OK	d1gvytnctu52gl.cloudfront.net	72 B	216.137.61.137:80
⊕ GET logo.gif	200 OK	d1gvytnctu52gl.cloudfront.net	2.1 KB	216.137.61.137:80
⊕ GET ph_callout_left_top.gif	200 OK	d1gvytnctu52gl.cloudfront.net	28.4 KB	216.137.61.137:80
⊕ GET ph_callout_left_rebel.jpg	200 OK	d1gvytnctu52gl.cloudfront.net	13.7 KB	216.137.61.137:80
⊕ GET home_main_callout.jpg	200 OK	d1gvytnctu52gl.cloudfront.net	35.2 KB	216.137.61.137:80
⊕ GET free_shipping_callout.jpg	200 OK	d1gvytnctu52gl.cloudfront.net	14.3 KB	216.137.61.137:80
⊕ GET best_selling_img01.jpg	200 OK	d1gvytnctu52gl.cloudfront.net	1.8 KB	216.137.61.137:80

Quelle: eigener Screenshot

4.3.3. Auslagerung vom Skin-Ordner und Javascript

Bei diesem zweiten Test werden alle benötigten Javascript-Dateien und der gesamte Skin-Ordner auf das Amazon CloudFront verschoben. Der Skin-Ordner beinhaltet alle CSS-Dateien und Bilder, die für das Design der Seite verantwortlich sind. Das sind z.B. Logos oder Icons die im Design verwendet werden.

Die Ergebnisse aus dem Programm WAPT, für diesen Test sind in der Tabelle 5 abgebildet. Vergleicht man diese Resultate mit den Ausgangswerten aus Tabelle 4 kann man feststellen dass es keine Verbesserungen im Bereich der Performance gibt. Im Gegenteil, die Webseite ist zum Teil langsamer geworden, z.B. schon bei nur 50 gleichzeitigen Nutzern kommt eine Steigerung von 0,15 Sekunden zu Stande.

Tabelle 5: Performancetest - Auslagerung Javascript, Skin-Ordner Ergebnisse

Anzahl der Nutzer	20	50	100	200
Ø Reaktionszeit (mit Elemente), Sek.	0,7	0,9	0,96	2,15
Ø Reaktionszeit, Sek.	0,52	0,74	0,75	1,82
Ø Seiten pro Sekunde	7,14	16,8	33,1	48,9
Ø Zugriffe pro Sekunde	31,6	74,3	145	225

Die Ursache für dieses Ergebnis könnte dadurch zu Stande kommen, dass zu wenige Elemente vom eigenen Server geladen werden. Der Browser kann pro Domain nur eine begrenzte Anzahl an parallelen Requests schicken. Liegt demnach die Anzahl der gleichzeitig zu ladenden Elemente höher, als die maximale Anzahl an möglichen Requests, wird der Seitenaufbau verlangsamt.¹⁸

4.3.4. Konfiguration Modul - ImageCDN

In den nachfolgenden beiden Tests, geht es vorläufig um Produktbilder. Diese sollen zu Amazon CloudFront ausgelagert werden. Um das in Magento zu realisieren, wird das Magento Modul ImageCDN benutzt. In diesem Abschnitt wird die dafür notwendige Konfiguration beschrieben.

Nach der Installation des Moduls über Magento Connect, müssen noch wenige Einstellungen gemacht werden, damit das Modul richtig funktioniert.

Die Einstellungsmöglichkeiten findet man im Adminpanel. Hierzu muss zuerst in die Store-Konfiguration gewechselt werden: System -> Konfiguration. Dort befindet sich auf der linken Seite ein Tab mit der Aufschrift: Image CDN. Nach dem man auf diese Seite gewechselt ist, hat man alle Einstellungsmöglichkeiten des kompletten Modul vor sich.

Für die nachfolgenden Tests wurde das Modul, wie in Abbildung 9 zu sehen, konfiguriert.

¹⁸ [Mai11]

Abbildung 9: Konfiguration ImageCDN Allgemein

Image CDN

General Settings	
Current Adapter	<input type="text" value="Amazon S3/CloudFront"/> [STORE VIEW]
File Result Cache	<input type="text" value="In database (recommended)"/> [STORE VIEW]
Cache Time-to-Live	<input type="text" value="1440"/> [STORE VIEW] <small>▲ in minutes</small>
Verify File Size	<input type="text" value="Yes"/> [STORE VIEW] <small>▲ Simply ensures file size is greater than zero. May not work in rare cases.</small>
Image Compression	<input type="text" value="-- Use default --"/> [STORE VIEW] 1 is best quality/biggest file, 9 is worst quality/smaller <small>Compression is only available to jpeg and png formats. After changing this setting you may need to delete all of the previously created images on the remote server. This can easily be done by going to System > Cache Management and clicking on the Clear button next to Image Cache.</small>

Quelle: eigener Screenshot

Anschließend müssen noch die Zugangsdaten für den Amazon S3 Server und die CloudFront URL angegeben werden. Diese befinden sich in dem zu verwendeten Amazon Web Services Account. Wie in Abbildung 10 zu sehen, müssen die Zugangsdaten nur eingetragen werden.

Abbildung 10: Konfiguration ImageCDN Amazon Zugangsdaten

Amazon S3/CloudFront

To use CloudFront, simply enter your CloudFront domain in the base URL fields instead.

Access Key ID	<input type="text"/>	[STORE VIEW]
Secret Access Key	<input type="text"/>	[STORE VIEW]
Bucket	<input type="text" value="magentotest15"/>	[STORE VIEW]
	<small>▲ Must already be created and be dedicated for this Magento install only.</small>	
Base URL	<input type="text" value="http://d1gvytnctu52gl.cloudfront.net"/>	[STORE VIEW]
	<small>▲ Example: http://[bucket].s3.amazonaws.com</small>	
Secure Base URL	<input type="text" value="https://d1gvytnctu52gl.cloudfront.net"/>	[STORE VIEW]
	<small>▲ Example: https://[bucket].s3.amazonaws.com</small>	

Quelle: eigener Screenshot

4.3.5. Auslagerung der Produktbilder

Im vorletzten Test geht es nur um die Produktbilder. Die meisten Webseitenbetreiber wollen ihren Kunden große und hochauflösende Produktbilder bieten. Aber diese Bilder brauchen ihren Platz und die nötige Zeit zum Anzeigen beim Kunden, da meist mehrere Produktbilder zur gleichen Zeit geladen werden müssen. Deshalb sind nach Ansicht des Autors die Produktbilder die besten Seitenelemente, die auf CDNs ausgelagert werden können. Den Upload und die Auslieferung der Bilder übernimmt das Magento Modul ImageCDN. Die Konfiguration dafür ist in Abschnitt 4.3.4 beschrieben.

Wenn man die Ergebnisse der Geschwindigkeitsmessung beim Auslagern der Produktbilder in Tabelle 7 mit den Werten der vorhergehenden Tests vergleicht, bemerkt man schnell, dass bei diesem Test die Reaktionszeit merklich zurückgegangen ist. Schon bei 50 Nutzern auf der Webseite hat sich die Reaktionszeit um 0,08 Sekunden verbessert. Bei 100 Nutzern sogar um 0,23 Sekunden. Das ist der erste Test in dem der Einsatz eines CDN messbare Performance Verbesserung gebracht hat.

Tabelle 6: Performancetest - Auslagerung Produktbilder Ergebnis

Anzahl der Nutzer	20	50	100	200
Ø Reaktionszeit (mit Elemente), Sek.	0,61	0,67	0,55	1,38
Ø Reaktionszeit, Sek.	0,51	0,57	0,46	1,23
Ø Seiten pro Sekunde	7,32	18,1	37,6	60,5
Ø Zugriffe pro Sekunde	15,2	37,5	77,4	126

4.3.6. Auslagerung der Produktbilder und Javascript

In diesem letzten Test werden die Produktbilder und die Javascript Dateien zu Amazon CloudFront ausgelagert. Der Skin-Ordner, bleibt auf dem eigenen Server bestehen. So sind die Anfragen auf beide Server fast gleich verteilt.

Die Ergebnisse des Tests sind in Tabelle 7 aufgelistet. Dabei muss festgestellt werden, dass es nur kleine Abweichungen zum vorhergehenden Test gibt. Aber trotzdem sind bei 20, 50 und 200 Nutzern leichte Verbesserungen zu erkennen.

Tabelle 7: Performancetest - Auslagerung Produktbilder, Javascript Ergebnis

Anzahl der Nutzer	20	50	100	200
Ø Reaktionszeit (mit Elemente), Sek.	0,6	0,57	0,59	1,34
Ø Reaktionszeit, Sek.	0,49	0,47	0,48	1,18
Ø Seiten pro Sekunde	7,47	18,9	37,9	61
Ø Zugriffe pro Sekunde	15,5	39,2	77,5	126

4.4. Auswertung

Nachdem die vier einzelnen Tests betrachtet wurden, kann eine bezüglich der Geschwindigkeitsverbesserung gemacht werden. In Tabelle 8 sind alle durchschnittlichen Reaktionszeiten, mit den Seitenelementen, für die einzelnen Tests aufgelistet. Die jeweils besten Ergebnisse sind grün und die schlechtesten sind rot hinterlegt. Es wird nur die Reaktionszeit aufgelistet, weil es der bedeutendste Wert ist. Denn der Nutzer wartet so viele Sekunden bis die komplette

Seite mit allen Seitenelementen geladen und aufgebaut ist, dazu gehören natürlich auch Bilder und diverse Icons.

Die Resultate zeigen, dass der langsamste Seitenaufbau durch die Auslagerung von Javascript und dem Skin-Ordner zu Stande kommt. Dieser Test hat sogar eine Verschlechterung der Ladezeit zur Folge gehabt, im Vergleich zum normalen Seitenaufbau ohne diverse Anpassungen.

Tabelle 8: Vergleich der Ergebnisse aus dem Performancetest

	Ø Reaktionszeit in Sekunden (mit Seitenelementen)			
	20	50	100	200
Anzahl Nutzer	20	50	100	200
Ohne Auslagerung	0,77	0,75	0,78	1,62
Auslagerung von Javascript + Skin	0,7	0,9	0,96	2,15
Auslagerung Produktbilder	0,61	0,67	0,55	1,38
Auslagerung Produktbilder + Javascript	0,6	0,57	0,59	1,34

Das beste Ergebnis wird mit der Auslagerung von Produktbildern und statischen Javascript-Dateien erzielt. So wird für eine gute Verteilung der Anfragen zwischen CDN und eigenem Server realisiert. Die größeren und ladungsintensiven Bilder werden in das schnelle CDN ausgelagert und die kleinen Icons bzw. CSS-Dateien bleiben auf dem eigenem Server.

Aber auch nur die Auslagerung von Produktbildern führt zu einer Verbesserung der Geschwindigkeit beim Seitenaufbau. Das kann schon bei 100 Nutzern zu einer Verbesserung von 0,23 Sekunden führen.

5. Zusammenfassung und Fazit

Im ersten Teil dieser Arbeit wurde der Aufbau und die Funktionsweise eines Content Distribution Network erläutert und ein Zusammenhang zwischen der Performance eines Onlineshop und der wirtschaftlichen Einflussnahme hergestellt. Anschließend wurde das erlangte theoretische Wissen an einem Magento Demo-Shop praktisch angewandt und verschiedene Tests zur Auswertung durchgeführt.

Durch die Auswertung der ausgeführten Tests, kann festgehalten werden, dass eine Performanceverbesserung durch Nutzung eines CDN möglich ist. Es muss aber genau entschieden und festgelegt werden welche Elemente einer Webseite ausgelagert werden können. Denn wie die Auswertung gezeigt hat, rentiert es sich am meisten bei größeren statischen Dateien.

Aber auch die Kosten durch die Nutzung eines CDN dürfen nicht vernachlässigt werden. Vor einer Nutzung sollte man die eventuellen Kosten pro Monat errechnen und ins Verhältnis zur eventuellen Verkaufssteigerung setzen, um herauszufinden ob sich ein Umstieg lohnt. Bei einem hohen und weltweiten Besucheraufkommen sind die großflächig verteilten Server ein großer Vorteil bei der Auslieferung der Inhalte. Es lohnt sich deshalb vorerst für größere Webseiten mit einem internationalen Besucherstamm, da man an diese die eigenen Inhalte schnell und stabil ausliefern kann.

Doch auch in der Zukunft wird es auf dem Gebiet des Cloud-Computing weitere Veränderungen und Verbesserungen geben, deshalb sollte man dieses Thema nicht aus den Augen verlieren.

Literaturverzeichnis

- [Kie10] Kiellisch, T.: „Website-Performance: Geschwindigkeit des Internetauftritts messen und optimieren“, o.O., 2010,
<http://www.onpulson.de/themen/2305/website-performance-geschwindigkeit-des-internetauftritts-messen-und-optimieren/>
Abruf: 05.09.2011
- [Zen10] Zenner, R., Kopp, V.: „Magento - Das Handbuch für Entwickler“, 1. Auflage, Köln, O'Reilly Verlag GmbH & Co. KG, 2010
- [FK10] Fries, E., Kambach P.: „Wie Content Delivery Networks Medieninhalte geschickt verteilen: Schneller surfen“, Düsseldorf, 2010, <http://t3n.de/magazin/content-delivery-networks-medieninhalte-geschickt-225211/>
Abruf: 10.09.2011
- [Göt04] Göttinger, T.: „Content Distribution Networks“, in Racz, P. (Hrsg.): Internet Economics III, Institut für Informationstechnische Systeme, Neubiberg, 2004, S. 133 - 156
- [Pet11] Peters, M.: „Content Delivery Network (CDN) nutzen im eCommerce“, o.O., 2011,
<http://www.e-commerce-blog.de/2011/01/17/content-delivery-network-cdn-nutzen-im-ecommerce/2922/>
Abruf: 06.09.2011
- [Ama11a] o.V.: „Amazon CloudFront“, o.O., 2011,
<http://aws.amazon.com/de/cloudfront/>
Abruf: 06.09.2011
- [Eri11] Eriksdotter, H.: „Die Top 10 der Cloud-Anbieter in 7 Kategorien“, Hamburg, 2011,
http://www.channelpartner.de/channelcenter/cloud_computing/2383359/index.html
Abruf: 11.09.2011
- [Ama11b] o.V.: „Amazon CloudFront: Developer Guide“, Amazon Web Services LLC, 2011,

http://s3.amazonaws.com/awsdocs/CF/latest/cf_dg.pdf

Abruf: 11.09.2011

[Ama11c] o.V.: „Amazon Simple Storage Service“, o.O., 2011,

<http://aws.amazon.com/de/s3/>

Abruf: 18.09.2011

[Sie04] Siemer, A.: „Seminar: Innovative Netztechnologien

Content Distribution Networks“, o.O., 2004,

http://www.siemer.tv/wp-content/uploads/2006/10/in_seminar_ausarbeitung.pdf

Abruf: 06.09.2011

[Wap11] o.V.: „WAPT Performance Testing Tool“, o.O., 2011,

<http://www.loadtestingtool.com/product.shtml>

Abruf: 18.09.2011

[Mai11] Maier, F.: „CSS Sprites“, o.O., 2011,

<http://www.graccem.com/css/css-sprites>

Abruf: 18.09.2011

Ehrenwörtliche Erklärung

Ich erkläre hiermit ehrenwörtlich,

1. dass ich meine Studienarbeit mit dem Thema:

Performanceoptimierung in Magento durch Nutzung eines Content Distribution Network

ohne fremde Hilfe angefertigt habe,

2. dass ich die Übernahme wörtlicher Zitate aus der Literatur sowie die Verwendung der Gedanken anderer Autoren an den entsprechenden Stellen innerhalb der Arbeit gekennzeichnet habe und
3. dass ich meine Studienarbeit bei keiner anderen Prüfung vorgelegt habe.

Ich bin mir bewusst, dass eine falsche Erklärung rechtliche Folgen haben wird.

Ort, Datum

Unterschrift