Routingverfahren zur Lastverteilung in Content-Delivery-Networks

Seminararbeit von
Lilli Winschel

vorgelegt am
Lehrstuhl für Praktische Informatik IV
Prof. Dr. W. Effelsberg
Fakultät für Mathematik und Informatik
Universität Mannheim

Betreuer: Christian Liebig

Dezember 2004

Inhaltsverzeichnis

1 Einleitung

In den letzten Jahren hat das Internet als Medium zur Übertragung vielfältiger Inhalte eine rasante Entwicklung genommen. Mehr und mehr Endnutzer rufen immer umfangreichere Inhalte ab und werden gleichzeitig hinsichtlich der Übertragungsqualität zunehmend anspruchsvoller. Die grundlegende Architektur des Internets kann allerdings verschiedene Dimensionen der Übertragungsqualität (z.B. Latenzzeit, durchschnittliche Übertragungsrate und Konstanz der Übertragung) nicht garantieren. Insbesondere kommerzielle Inhalteanbieter stehen somit vor dem Problem, die so entstehende Lücke zwischen Kundenwünschen und technischen Gegebenheiten zu schließen.

Einen sinnvollen Ansatz zur Lösung dieses Problems bieten Content-Delivery-Networks (CDN). Die Grundidee derartiger Netzwerke besteht darin, Inhalte nicht zentral sondern in Kopie auf zahlreichen sog. Replica-Servern vorzuhalten. Gelingt es nun, Anfragen von Endnutzern auf geeignete Replica-Server in deren Nähe umzuleiten, kann bei gleichzeitiger Verringerung der Netzwerkbelastung und des Hardwarebedarfs der Inhalteanbieter die Übertragungsqualität deutlich verbessert werden.

Eine zentrale Herausforderung bei der Implementierung eines CDN ist die Umleitung der Nutzeranfragen an die Replica-Server. Dabei gilt es, die zwei Fragen zu beantworten, nach welchen Kriterien ein Replica-Server auszuwählen und wie die Anfrage technisch an diesen Server umzuleiten sei. Diese Arbeit stellt einige mögliche Antworten vor.

2 Routing in Content-Delivery-Networks

2.1 Aufbau von Content-Delivery-Networks

Jedes CDN besteht im Kern aus fünf Komponenten [9],

- einem *Ursprungsserver*, auf welchem der Inhalteanbieter die maßgeblichen Inhalte ablegt,
- zahlreichen *Replica-Servern*, welche Kopien dieser Inhalte in der Nähe der Endnutzer vorhalten,
- dem *Distributionssystem*, welches die Inhalte vom Ursprungsserver auf die Replica-Server verteilt,
- dem *Request-Routing-System*, welches die Anfragen der Nutzer auf geeignete Replica-Server umleitet,
- sowie dem *Accounting-System*, welches den Datenverkehr protokolliert und die Nutzung und den Zustand des CDN beschreibende Kennzahlen bereithält.

Routing-Aufgaben fallen in einem CDN offensichtlich dem Distributionssystem und dem Request-Routing-System zu. Beide Komponenten werden daher im folgenden eingehender betrachtet.

2.2 Distributionssystem

Die Aufgabe des Distributionssystems besteht darin, die Inhalte vom Ursprungsserver auf die Replica-Server zu übertragen und dabei die Aktualität und Konsistenz der replizierten Inhalte sicherzustellen. Diese beiden Anforderungen können grundsätzlich auf zwei Arten erfüllt werden.

Zum einen kann das Distributionssystem neue oder geänderte Inhalte proaktiv an alle Replica-Server verteilen, zum anderen kann es aber auch lediglich eine Änderungsnachricht senden, welche die Replica-Server darüber informiert, daß sie über eine veraltete Kopie verfügen und die aktuellen Inhalte im Bedarfsfall beim Urpsrungsserver anzufordern haben [2].

Beiden Vorgehensweisen ist gemein, daß das Distributionssystem für die effiziente Übertragung von Daten vom Ursprungsserver zu den Replica-Servern zu sorgen hat.

Erfolgt diese Übertragung über das Internet, stellt sich unmittelbar die Frage nach dem effizienten Routing. Eine mögliche Vorgehensweise besteht darin, zwischen dem Ursprungsserver und allen Replica-Servern jeweils einen Unicast-Kanal aufzubauen. Da aber diese Vorgehensweise offensichtlich Netzwerkbandbreite verschwendet, ist der Einsatz von Multicasting vorzuziehen [9]. Bei dieser Vorgehensweise erstellt das Distributionssystem in der Anwendungsschicht einen Multicast-Baum, indem es zunächst einen möglichst alle Verbindungen zwischen den Replica-Servern umfassenden Graphen konstruiert und anschließend einen optimalen Spannbaum ermittelt [2, 4]. Beide Vorgehensweisen leiden allerdings unter der nur begrenzt vorhersehbaren Übertragungsqualität des Internets.

Ein technisch grundlegend anderer Ansatz ist die Verteilung von Inhalten an die Replica-Server über Satelliten. Die Einrichtung einer solchen Lösung ist zwar mit erheblichem Aufwand verbunden, ermöglicht es aber, in den von den Satelliten abgedeckten Regionen beliebig viele Replica-Server zu bedienen; sie bietet somit Potential für erhebliche Kostenersparnisse. Ein weiterer Vorteil von Satellitenübertragungen besteht in deren extrem hoher und kalkulierbarer Qualität. Insbesondere zeitkritische Inhalte können also problemlos verteilt werden.

2.3 Request-Routing-System

Die Aufgabe des Request-Routing-Systems ist die Umleitung von Nutzeranfragen an die Replica-Server. Diese Umleitung erfolgt in zwei Schritten:

Zunächst wird unter Zuhilfenahme von Informationen des Accounting-Systems der für die Beantwortung der Anfrage geeignetste Replica-Server ausgewählt; dieser Schritt wird in Abschnitt 3 ausführlich beschrieben. Anschließend wird die vom Endnutzer initiierte Anfrage rein technisch auf den

gewählten Replica-Server umgeleitet [12]; die hierfür geeigneten Verfahren werden in Abschnitt 4 diskutiert.

3 Auswahl des Replica-Servers

3.1 Kennzahlen

Ein CDN kann seinen Zweck offensichtlich nur dann erfüllen, wenn sein Request-Routing-System regelmäßig in der Lage ist, den "besten" Replica-Server auszuwählen. Grundlage dieser Entscheidung sind Kennzahlen, welche die Eignung eines Replica-Servers für die Beantwortung einer Anfrage beschreiben und in geeigneter Weise zu einem eindimensionalen, abstrakten Eignungsmaß aggregiert werden können [2, 8, 11].

Geeignete Kennzahlen lassen sich im wesentlichen in drei Gruppen einteilen,

- *serverseitige* Kennzahlen (z.B. CPU-Auslastung, Anzahl der aktiven Verbindungen, I/O-Last), anhand derer der Replica-Server mit der geringsten durchschnittlichen Belastung ermittelt werden kann,
- *clientseitige* Kennzahlen (z.B. Identität und Präferenzen des anfragenden Nutzers), welche etwa die Bereitstellung unterschiedlicher Qualitäten für normale und Premium-Kunden ermöglichen [10],
- und die *Netzwerkverbindung* zwischen Client und Server beschreibende Kennzahlen (z.B. geographische Nähe, Latenzzeit, durchschnittliche Übertragungsrate, Konstanz der Übertragungsrate, Paketverlustrate, Round-Trip-Time), anhand derer der Replica-Server mit der besten Verbindung zum Client bestimmt werden kann.

Welches Gewicht die einzelnen Kennzahlen erhalten sollten, hängt natürlich zum Teil von den zu übertragenden Inhalten ab. So ist beispielsweise bei

der Übertragung von Streaming-Media-Inhalten die Eignung der Netzwerkverbindung besonderes wichtig, während dynamisch generierte Inhalte eine Betonung serverseitiger Kennzahlen sinnvoll erscheinen lassen.

3.2 Meßmethoden

Aufgrund der dezentralen Struktur des Internets erfordert die Bestimmung der oben genannten Kennzahlen allerdings geeignete Meßmethoden. Diese lassen sich im wesentlichen in drei Kategorien einteilen: Passive-Measurement, Active-Probing und Feedback-Information; sie werden im folgenden genauer vorgestellt.

3.2.1 Passive-Measurement

Unter Passive-Measurement versteht man die Beobachtung des tatsächlich stattfindenden Datenverkehrs zwischen Clients und Replica-Servern. Zunächst wird die Übertragungsqualität jeder Netzwerkverbindung gemessen; anschließend werden diese Meßdaten gesammelt und so aufbereitet, daß sie vom Request-Routing-System verwendet werden können. Auf diese Weise können beispielsweise die Latenzzeit, die Paketverlustrate, die Übertragungsrate und die Round-Trip-Time (RTT) ermittelt werden [8].

Als konkretes Beispiel für Passive-Measurement wird nachfolgend die Messung der RTT durch einen Router mit Netzwerkunterstützung beschrieben. Wie Abbildung 1 zeigt, mißt ein derartiger Router die RTT autonom, indem er die Anfrage- und Antwortpakete beobachtet. Dabei werden sowohl Verzögerungen auf dem Server als auch im Netzwerk zwischen dem Router und dem Server erfaßt.

Um die RTT zu messen, werden Zeitstempel verwendet. Das Anfragepaket eines Clients, welches den Router erreicht, wird mit einem Zeitstempel versehen und an den ausgewählten Server weitergeleitet. Der Server sendet das Antwortpaket zusammen mit diesem Zeitstempel an den Router zurück, der

nun die RTT leicht errechnen kann. Zu beachten ist hierbei, daß lediglich die RTT zwischen Router und Server gemessen wird, während diejenige zwischen Client und Router unberücksichtigt bleibt.

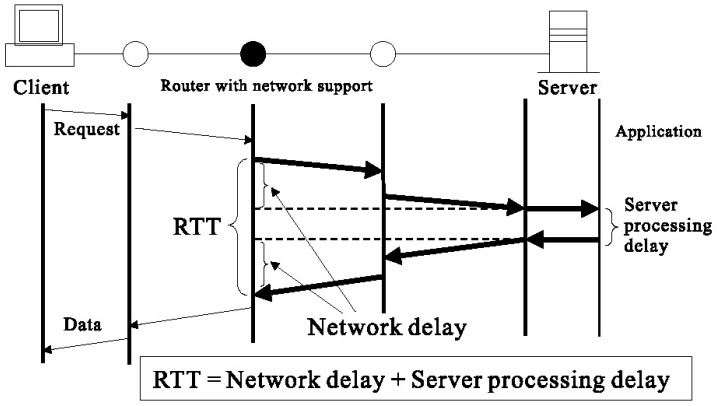

Abbildung 1: Passive-Measurement der Round-Trip-Time [8]

Der Hauptvorteil des Passive-Measurement ist, daß die Messung keinen zusätzlichen Datenverkehr erzeugt. Ein Nachteil ist hingegen, daß mit dieser Meßmethode gewonnene Informationen vergangenheitsorientiert sind und deshalb veralten sein können. Insbesondere in einem dynamischen Umfeld trifft das Request-Routing-System also möglicherweise Fehlentscheidungen, wenn es sich allein auf diese Informationen verläßt.

3.2.2 Active-Probing

Beim Active-Probing wird die Verbindung zwischen früheren oder potentiellen Clients sowie den Replica-Servern anhand einer oder mehrerer Methoden aktiv getestet. Die Messung wird dabei von den Replica-Servern initiiert [2, 3].

Ein Möglichkeit, Active-Probing zu implementieren, ist die Verwendung von ICMP-Echos, besser bekannt als Pings. Dabei sendet ein Replica-Server eine

ICMP-Echo-Request an einen Client, welcher dann eine korrespondierende ICMP-Echo-Reply an den Server zurückschickt. So kann auf einfache Weise ein aktuelles Bild der Verbindungsqualität zwischen Clients und Servern gewonnen werden [1].

Ein gravierender Nachteil dieser Technik besteht in dem zusätzlichen Datenverkehr; der Frequenz der Messungen sind also gewisse Grenzen gesetzt. Des weiteren treten bei den auf Grundlage von ICMP-Echos gewonnenen Informationen Verzerrungen auf, da ICMP-Echo-Pakete in der Regel deutlich kleiner als Inhalte transportierende Datenpakete sind und zudem von Routern oftmals nur mit geringer Priorität weitergeleitet werden. Auch lösen ICMP-Echos in einigen Intrusion-Detection-Systemen Alarmmeldungen aus und werden von manchen Firewalls vollständig blockiert.

3.2.3 Feedback-Information

Feedback-Information-Methoden legen den Fokus auf die Ermittlung serverseitiger Kennzahlen, indem sie z.B über HTTP-Anfragen die Replica-Server periodisch mit anwendungsspezifischen Anfragen testen. Oft sind auf diese Weise gewonnene Informationen jedoch unpräzise. Alternativ können auf den Replica-Servern installierte Agenten deren Auslastung überwachen und regelmäßig an das Accounting-System berichten [2, 3].

4 Verfahren zur Request-Redirection

Verfahren zur Request-Redirection leiten die von den Endnutzern initiierten Anfragen rein technisch auf die Replica-Server um. Sie unterscheiden sich dabei hinsichtlich des Orts, an dem die Entscheidung über das Ziel der Umleitung gefällt wird: auf dem Client, im Netzwerk oder auf dem Server [2, 6].

4.1 Clientseitige Request-Redirection

Den einfachsten Fall stellt die Request-Redirection auf dem Client dar. Hier wird dem Endnutzer eine Liste von Replica-Servern präsentiert, aus der er willkürlich einen auswählt. Offensichtlich ist dieses Verfahren extrem einfach zu implementieren, hat aber auch den erheblichen Nachteil, daß das CDN fast keine Kontrolle über die Zuordnung von Clients zu Replica-Servern hat.

4.2 Request-Redirection im Netzwerk

Die wichtigsten Verfahren zur Request-Redirection im Netzwerk sind DNS-basiertes Request-Routing und Anycasting. Ein grundsätzlicher Vorteil beider Verfahren ist, daß ihre Implementierung keine Änderungen an Clients und Servern erfordert. Dagegen stehen der hohe Aufwand für die Implementierung einer neuen Netzwerkinfrastruktur und die etwaige Instabilität dieser Verfahren.

4.2.1 DNS-basiertes Request-Routing

DNS-basiertes Request-Routing ist bei den kommerziellen CDNs derzeit das am häufigsten eingesetzte Verfahren. Sein Ansatzpunkt sind die im Internet allgegenwärtigen DNS-Server, deren Aufgabe die Übersetzung von Domainnamen in IP-Adressen ist [6, 12, 13].

Die grundsätzliche Idee besteht darin, Anfragen des Clients an einen lokalen DNS-Server über die übliche DNS-Server-Hierarchie an einen speziellen, vom CDN bereitgestellten DNS-Server weiterzuleiten, welcher die IP-Adresse des ausgewählten Replica-Servers zurückgibt. Der DNS-Server des CDNs hat zudem die Möglichkeit, auf Anfrage die IP-Adressen gleich mehrerer Replica-Server an den lokalen DNS-Server zurückzusenden, welche jener dann mittels der Round-Robin-Methode an eventuell kurze Zeit später anfragende Clients vergeben kann.

Das DNS-basierte Request-Routing weist trotz seiner Verbreitung zahlreiche Nachteile auf: Das vielstufige Weiterleiten von Anfragen innerhalb der DNS-Server-Hierarchie resultiert in einer hohen Round-Trip-Time, was zur Folge hat, daß dieses Verfahren schlecht skaliert. Des weiteren hat DNS-basiertes Request-Routing Schwierigkeiten mit der Bedienung einer großen Anzahl an Inhalteanbietern. Und schließlich gehen beim Weiterleiten der Anfrage an zwischengelagerte DNS-Server Informationen über den Client verloren, die entsprechend nicht in die Auswahl des Replica-Servers einfließen können.

4.2.2 Anycasting in der Anwendungsschicht

Die Grundidee von Anycasting besteht darin, eine Gruppe von Replica-Servern, welche aus Nutzersicht äquivalente Inhalte bereitstellen, unter einem sog. Anycast-Domain-Name (ADN) zusammenzufassen. Anycasting ist dem bekannten Multicasting insofern darin ähnlich, daß einer Adresse mehrere Server zugeordnet werden. Da aber diese Server äquivalent sind, findet ein Datenfluß tatsächlich nur zwischen zwei Punkten statt – ein Merkmal, welches Anycasting mit Unicasting teilt [2, 6, 14].

Die Implementierung von Anycasting in der Anwendungsschicht erfordert es, den Clients die Adresse eines sog. Anycast-Resolvers mitzuteilen, welcher eine ADN auf einen geeigneten Replica-Server abzubilden vermag. Insofern haben Anycast-Resolver grundsätzlich dieselbe Aufgabe wie DNS-Server, sind aber zudem explizit dafür ausgelegt, Anfragen innerhalb eines verteilten Systems geeignet umzuleiten.

4.3 Serverseitige Request-Redirection

Die gebräuchlichsten serverseitigen Request-Redirection-Verfahren sind HTTP-302-Redirection und URL-Rewriting. Hierbei wird die Client-Anfrage von dem Server, an den sie ursprünglich gerichtet war, gänzlich bzw. teilweise auf einen besseren Server umgeleitet.

Ein Vorteil dieser Verfahren ist, daß der Server üblicherweise sehr detailierte Informationen über den Zustand des CDNs besitzt und somit eine fundierte Entscheidung treffen kann. Nachteile bestehen hingegen darin, daß jede Anfrage zunächst von ein- und demselben Server zu bearbeiten ist und daß dieser Server modifiziert werden muß [2, 6].

4.3.1 HTTP-302-Redirection

Bei der HTTP-302-Redirection gibt der Server, an den eine Anfrage ursprünglich gerichtet war, lediglich eine dynamisch generierte Umleitung auf einen geeigneten Replica-Server zurück. Der Client empfängt diese Umleitung und iniitiert automatisch eine erneute Anfrage an den benannten Replica-Server, welcher schließlich die angeforderten Inhalte liefert.

Der Hauptvorteil dieser Verfahrens ist seine relativ leichte Implementierbarkeit; der Hauptnachteil besteht darin, daß es die Anzahl der für die Übertragung der Inhalte erforderlichen Verbindungen verdoppelt.

4.3.2 URL-Rewriting

Beim URL-Rewriting werden in einem zentralen HTML-Dokument befindliche URLs eingebetteter Inhalte so modifiziert, daß sie auf einen geeigneten Replica-Server verweisen. Dabei kann die Auswahl des Replica-Servers pro Objekt getroffen werden. Offensichtlich ist dieses Verfahren der Request-Redirection dann besonders sinnvoll, wenn die Inhalte aus einer einfachen Basisstruktur mit umfangreichen zusätzlichen Inhalten bestehen.

Weiterhin kann zwischen proaktivem und reaktivem URL-Rewriting unterschieden werden. Beim proaktiven Rewriting werden die URLs der eingebetteten Objekte festgeschrieben, bevor die Inhalte auf den Ursprungsserver geladen werden. Das bedeutet, daß für das Rerouting keine clientspezifischen Informationen genutzt werden und daß keine Anpassungen an die aktuelle Lastverteilung vorgenommen werden können. Beim reaktiven URL-Rewriting

werden die URLs der eingebetteten Objekt hingegen erst festgelegt, nachdem die Anfrage eines Clients beim Ursprungsserver eingegangen ist. Somit ist reaktives URL-Rewriting flexibler als proaktives, aber gleichzeitig auch schwieriger zu implementieren.

5 Zusammenfassung

Diese Arbeit beschreibt Verfahren zum Routing in CDNs. Die zentrale Aufgabe dieser Verfahren besteht darin, für die schnelle Bearbeitung von Nutzeranfragen zu sorgen, indem sie diese Anfragen an geeignete Replica-Server des CDNs weiterleiten. Dabei gilt es zu klären, nach welchen Kriterien ein Replica-Server auszuwählen und wie die Anfrage technisch an diesen Server umzuleiten ist. Bezüglich der ersten Frage wurden in Abschnitt 3 diverse Kennzahlen angegeben, welche entscheidungsrelevante Eigenschaften von Clients, Servern und Netzwerkverbindungen beschreiben; bezüglich der zweiten Frage wurden in Abschnitt 4 Verfahren zur Request-Redirection vorgestellt, wobei die Umleitung grundsätzlich beim Client, auf dem Server oder im Netzwerk erfolgen kann.

CDNs adressieren ein Grundproblem, dessen Lösung sehr bedeutend für die Weiterentwicklung des Internets ist. Bedauerlicherweise verletzten aber die Routing-Verfahren von CDNs als proprietäre, nachträglich aufgesetzte Mechanismen die ursprüngliche Architektur des Internets sowie seine Philosophie als offenes System in vielfältiger Weise. Beispielsweise erfordert DNS-basiertes Request-Routing den Zugriff von DNS-Servern auf Routing-Informationen, was einen Verstoß gegen das Schichtmodell darstellt [5]. Aus diesem Grund wurden etwa mit TRIAD oder KAESAR Ansätze zur Lösung des Grundproblems vorgelegt, welche mit der Grundstruktur des Internet in Einklang stehen [5, 7]. Ob sich diese Ansätze aber langfristig durchsetzen werden, bleibt abzuwarten.

Literatur

[1] Auerbach, K. (2004) Limitations of ICMP Echo for network measurement. InterWorking Labs, Scotts Valley, CA.

[2] Bartolini, N., E. Casalicchio und S. Tucci (2004) A Walk through Content Delivery Networks, in: M. Calzarossa und E. Gelenbe (Hrsg.), *MASCOTS 2003, LNCS 2965*, Springer, Heidelberg, 1–25.

[3] Brekne, T., M. Clemetsen, P. Heegaard, T. Ingvaldsen und B. Viken (2002) Requirements for a Measurement Architecture. R&D Scientific Document N 62/2002, Telenor Communication, Fornebu, Norway.

[4] Chen, Y., R. Katz und J. Kubiatowicz (2002) Dynamic Replica Placement for Scalable Content Delivery, in: P. Druschel, F. Kaashoek und A. Rowstron (Hrsg.), *IPTPS 2002, LNCS 2429*, Springer, Heidelberg, 306–318.

[5] Cheriton, D. und M. Gritter (2001) TRIAD: A New Next-Generation Internet Architecture. Computer Science Department, Stanford University.

[6] Kabir, M., E. Manning und G. Shoja (2002) Request-Routing Trends and Techniques in Content Distribution Network, in: *Proceedings ICCIT 02, Dhaka, Bangladesh, December 2002*, 315–320.

[7] Kim, H. und K. Chon (2001) KAESAR: A Scalable Information Dissemination Architecture on the Internet. Department of Electrical Engineering and Computer Science, Korea Advanced Institute of Science and Technology.

[8] Miura, H. und M. Yamamoto (2002) Content Routing with Network Support Using Passive Measurement in Content Distribution Networks. in: *Proceedings of the IEEE International Conference on Computer Communications and Networks, Miami, FL, October 2002.*

[9] Peng, G. (2003) CDN: Content Distribution Network, Technischer Report, Computer Science Department, Stony Brook University.

[10] Rangarajan, S., S. Mukherjee, P. Rodriguez (2003) A Technique for User Specific Request Redirection in a Content Delivery Network. Konferenzbeitrag, Workshop on Web Content Caching and Distribution, New York.

[11] Ratnasamy, S., M. Handley, R. Karp und S. Shenker (2002) Topologically-Aware Overlay Construction and Server Selection. Konferenzbeitrag, Annual Joint Conference of the IEEE Computer and Communications Societies, New York.

[12] Vakali, A. und G. Pallis (2003) Content Delivery Networks: Status and Trends, *IEEE Internet Computing*, November/Dezember 2003, 68–74.

[13] Zari, M., H. Saiedian und M. Naeem (2001) Understanding and Reducing Web Delays. *IEEE Computer*, 34(12), 30–37.

[14] Zegura, E., M. Ammar, Z. Fei und S. Bhattacharjee (2000) Application-Layer Anycasting: A Server Selection Architecture and Use in a Replicated Web Service. *IEEE/ACM Transactions on Networking*, 8(4), 455–466.