

Systeme II



Albert-Ludwigs-Universität Freiburg
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

Christian Schindelhauer
Sommersemester 2007
12. Vorlesungswoche
09.07.-13.07.2007
schindel@informatik.uni-freiburg.de



Kapitel IX

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelbauer

World Wide Web



Das World-Wide Web

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelbauer

- **WWW und HTML**
- **Aufbau einer WWW-Suchmaschine**
- **Googles Pagerank algorithm**
- **Die Struktur des Web**
- **Web-Caching im Internet**



- **1940er Vannevar Bush beschreibt Memex als Maschine, die Text- und Bildinformation in ver“link“t speichert.**
- **1960 Beginn des Xanadu-Projekts durch Ted Nelson**
 - 1965 Ted Nelson prägt die Begriffe hypertext und hypermedia auf der ACM 20th national conference
 - 1998 erste Veröffentlichung von Programmteilen
- **1972 Entwicklung von ZOG an Carnegie-Mellon University**
 - ZOG war eine Text-Datenbank
 - Einträge hatten Titel, Beschreibung und Menü-Punkte, die zu anderen Einträgen führen
 - ZOG-Mitentwickler Donald McCracken und Robert Akscyn entwarfen später KMS, Knowledge Management System
- **1978 Andrew Lippman MIT entwickelte das erste wahre Hypermedia-Produkt *Aspen Movie Map* [video](#)**
- **1984 Hypertextsystem: Notecard von Xerox PARC**
- **1987 Bill Atkinson (Apple) stellt Hypertextsystem HyperCard vor**



Geschichte des Web

➤ 1989 Tim Berners-Lee (CERN)

- Information Management: A Proposal
- World-Wide Web: An Information Infrastructure for High-Energy Physics“

Wichtige Browser:

➤ 1991 Worldwideweb

➤ 1993 Lynx

➤ 1993 Mosaic (Browser)

- 1994 Netscape
 - 1998 Mozilla
 - 2002 Firefox
 - 2002 Camino
- 1995 Microsoft Internet Explorer

➤ 1999 Konquerer

- 2002 Safari



Hypertext Transfer Protocol (HTTP)

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelbauer

- **Kommunikationsprotokoll für das World Wide Web**
- **Zweck: Veröffentlichung und Zugriff auf HTML Hypertext-Seiten**
- **Entwicklung koordiniert durch**
 - World Wide Web Consortium
 - Internet Engineering Task Force
 - RFC 2616 (1999) definiert HTTP/1.1
- **HTTP ist ein Anfrage/Antwort-Protokoll zwischen Clients und Servers**
 - Client
 - z.B.: Web-Browser, Spider,...
 - Web-Server
 - speichert oder erzeugt HTML-Dateien
 - Dazwischen:
 - Proxies, Gateways und Tunnels
 - HTTP-Client erzeugt eine TCP-Verbindung (Default-Port 80)
 - HTTP-Server hört diesen Port ab
- **HTTP-Ressourcen werden durch Uniform Resource Identifiers/Locators (URI/URL) identifiziert, z.B. <http://bundeskanzler.de/index.html>**



HTTP-Befehle

➤ HTTP kennt 8 Methoden

- HEAD
 - Fragt nach einer Antwort identisch zur GET-Anfrage, aber ohne Inhalt (body) - nur Kopf
- GET
 - Standardanfrage zum Erhalt einer Web-Seite
- POST
 - Übermittelt Daten an die Gegenstelle
- PUT
 - Schickt die Web-Seite
- DELETE
 - Löscht eine Ressource
- TRACE
 - Schickt die Anfrage unverändert zurück
- OPTIONS
 - Gibt die HTTP-Methoden des Servers aus
- CONNECT:
 - konvertiert die Anfrage zu einem TCP/IP-Tunnel, in der Regel um SSL-Verbindungen zu ermöglichen



HTTPS

- **https ist eine URI-Schema, das eine sichere HTTP-Verbindung anzeigt**
- **Die Verbindung https: URL zeigt HTTP an,**
 - dass ein spezieller TCP-Port (443) verwendet wird
 - und ein zusätzlicher Verschlüsselungs/Authentifizierungs-Layer zwischen HTTP and TCP.
- **Entwickelt von Netscape Communications Corporation für sicherheitsrelevante Kommunikation, wie Zahlungen, Logins,...**
- **Streng genommen ist HTTPS kein separates Protokoll**
 - Kombination aus HTTP und Secure Socket Layer (SSL) oder Transport Layer Security (TLS)
- **Schützt vor Abhören und Man-in-the-Middle-Angriffen**



TLS/SSL

- **Transport Layer Security (TLS) und sein Vorgänger Secure Sockets Layer (SSL) sind kryptographische Protokolle für sichere Kommunikation wie**
 - HTML, E-Mail, Instant Messaging
- **Sie beinhalten**
 - Public-Key-verschlüsselten Schlüsselaustausch und Zertifikat-basierte Authentifizierung
 - Symmetrische Verschlüsselung für den Datenverkehr
- **Momentan Implementation erlauben die folgenden Protokolle**
 - Public-Key-Kryptographie: RSA, Diffie-Hellman, DSA
 - Symmetrische Verschlüsselung: RC2, RC4, IDEA, DES, Triple DES, AES
 - One-Way-Hash-Funktionen: MD2, MD4, MD5 or SHA.



Dynamic HTML

➤ **Dynamisches HTML beschreibt die Kombination aus**

- HTML, Style-Sheets und Skripten
 - Cascading Style Sheets
 - Skripten wie Javascript
 - ...
- „Animiertes“ HTML: Web-Seite kann auf User-Aktionen reagieren ohne dass der Web-Server reagiert

➤ **Verschiedene Ansätze in**

- Netscape
- MS Internet Explorer



EXtensible Markup Language (XML)

➤ XML

- Beschreibt Daten und die Datenbeschreibung

➤ HTML

- zeigt Daten an und konzentriert sich auf die Darstellung

➤ XML tags

- sind nicht vordefiniert, müssen erstellt werden
- Document Type Definition (DTD) oder
- XML-Schema beschreibt die Daten

➤ XML-Daten können in HTML eingebettet werden

➤ XHTML

- Kombination aus HTML und XML
- Syntaktische Unterschiede:
 - XHTML-Elemente müssen ordentliche geschachtelt sein
 - XHTML-Elements in Kleinbuchstaben
 - ...



Web-Suchmaschinen

➤ durchsuchen das WWW nach Information

➤ Geschichte

- 1993 Aliweb
- 1994 WebCrawler, Infoseek, Lycos
- 1995 AltaVista, Excite
- 1996 Dogpile, Inktomi, Ask Jeeves
- 1997 Northern Light
- 1998 Google
- 1999 AlltheWeb, Teoma
- 2000 Baidu, Info.com, Yahoo! Search
- 2005 MSN Search, Ask.com, AskMeNow
- 2006 wikiseek, Quaero,...

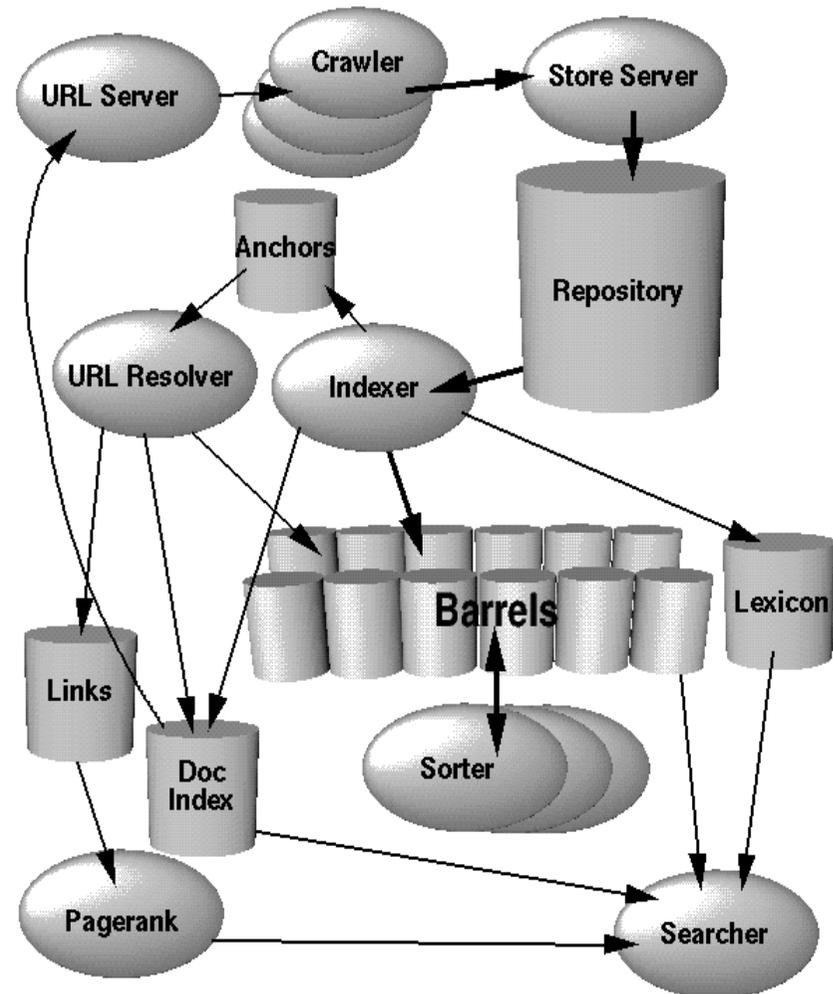
➤ **Gewinner (bis jetzt): Google mit über 75% Marktanteil**



Die Anatomie einer Web Search Maschine

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelbauer

- “The Anatomy of a Large-Scale Hypertextual Web Search Engine”, Sergey Brin and Lawrence Page,, 1998
- **Design des Prototyps von Google**
 - Stanford University 1998
- **Hauptkomponenten**
 - Web Crawler
 - Indexer
 - Pagerank
 - Searcher
- **Hauptunterschied zwischen Google und anderen Suchmaschinen (1998)**
 - Der Pagerank Algorithmus





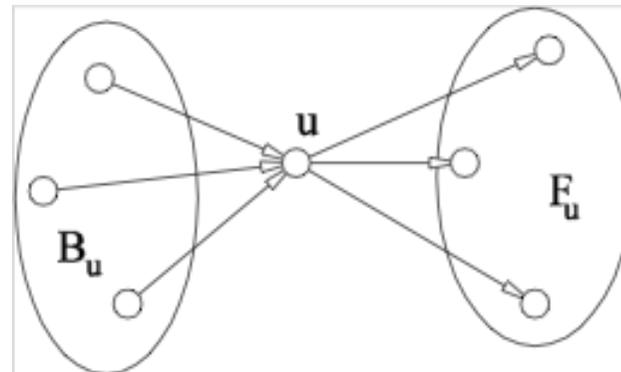
Der vereinfachte PageRank-Algorithmus

➤ Vereinfachter PageRank-Algorithmus

- Rang einer Web-seite $R(u) \in [0, 1]$
- Wichtige Seiten übergeben ihre Gewicht an verlinkte Seite

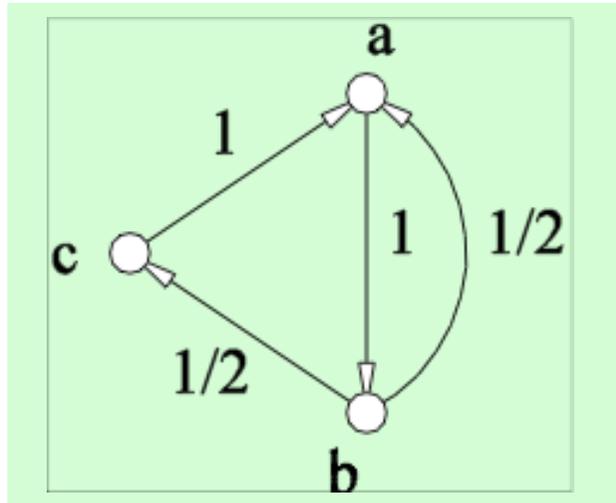
$$R(u) \leftarrow c \sum_{v \in B_u} \frac{R(v)}{|F_v|}$$

- c ist Normalisierungsfaktor so dass $\|R(u)\|_1 = 1$, d.h.
 - die Summe der Pageranks ist 1
- Vorgänger-Knoten B_u
- Nachfolger-Knoten F_u





Vereinfachter Pagerank-Algorithmus mit Beispiel



$$\begin{aligned}x &\leftarrow 0 \cdot x + \frac{1}{2} \cdot y + 1 \cdot z \\y &\leftarrow 1 \cdot x + 0 \cdot y + 0 \cdot z \\z &\leftarrow 0 \cdot x + \frac{1}{2} \cdot y + 0 \cdot z\end{aligned}$$

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} \leftarrow \begin{pmatrix} 0 & \frac{1}{2} & 1 \\ 1 & 0 & 0 \\ 0 & \frac{1}{2} & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

Runde	a	b	c
0	1	1	1
1	3/2	1	1/2
2	1	3/2	1/2
3	5/4	1/2	3/4
⋮	⋮	⋮	⋮
10	1,19..	1,22..	0,59..
⋮	⋮	⋮	⋮
20	1,20..	1,20..	0,60..



Der zufällige Web-Surfer

➤ Algorithmus

- Starte mit (gleichwahrscheinlich) zufälliger Web-Seite
- Wiederhole t Runden oft:
 - Falls kein Link auf der aktuellen Seite vorhanden ist, stoppe und gib nichts aus
 - Wähle gleichwahrscheinlich einen Link der aktuellen Web-Seite
 - Folge diesem Link und gehe auf die Web-Seite
- Gib die aktuelle Web-Seite aus

➤ Lemma

- Die Wahrscheinlichkeit, dass Web-Seite i vom zufälligen Web-Surfer ausgegeben wird, ist gleich der Wahrscheinlichkeit, die der vereinfachte Pagerank-Algorithmus ausgibt (ohne Normalisierung)

➤ Beweis folgt aus der Definition der Markov-Ketten



Aktuelle Probleme

- **Google arbeitet nicht nur nach diesem Prinzip**
 - Manipulation durch Web-Spam
- **Suchergebnisse werden manuell manipuliert**
 - wegen Gerichtsprozessen
- **Suchergebnisse werden personalisiert**
 - nach User/Land, wegen
 - Sprachbarrieren
 - Zensur (jugendgefährdend/politisch)
- **Das Angebot von Google wurde erheblich erweitert**
 - Maps, Videos, Wissenschaftliche Artikel, News-Suche, BLOG-Suche
- **Deep Web**



Der Webgraph

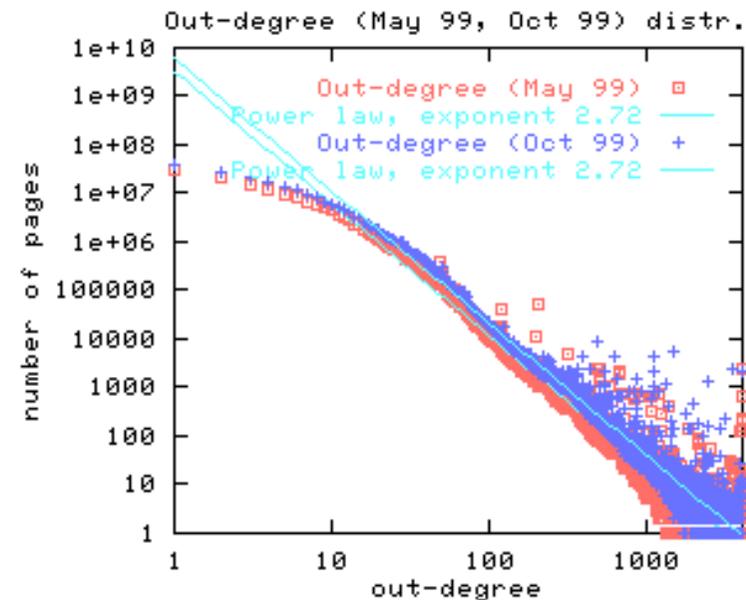
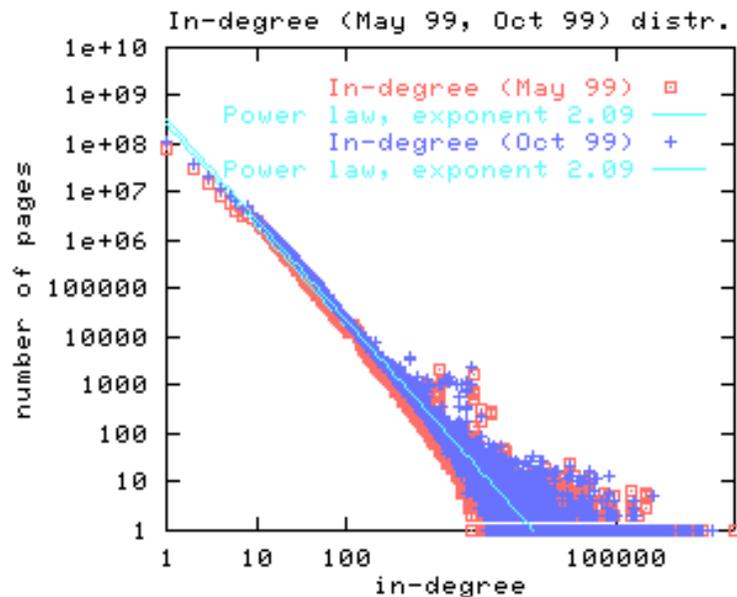
- **G_{www} :**
 - Statische HTML-Seiten sind Knoten
 - Link bezeichnen gerichtete Kanten
- **Ausgrad eines Knoten:**
 - Anzahl der Links einer Web-Seite
- **Eingrad eines Knoten**
 - Anzahl der Links, die auf eine Web-Seite zeigen
- **Gerichteter Pfad von Knoten u nach v**
 - Folge von Web-Seiten, indem man den Links folgt
- **Ungerichteter Pfad ($u=w_0, w_2, \dots, w_{m-1}, v=w_m$) von Seite u nach v**
 - Für alle i gibt es entweder einen Link von w_i nach w_{i+1} oder umgekehrt
- **Starke (schwache) Zusammenhangskomponente**
 - Maximale Knotenmenge in der zwischen allen Knoten dieser Menge ein (un) gerichteter Pfad besteht



Ein- und Ausgradverteilung

➤ Der Ein- und Ausgrad gehorchen einem Potenzgesetz (power law)

- d.h. die Häufigkeit von Eingrad i ist proportional zu $\sim 1/i^\alpha$



➤ Ergebnisse von

- Kumar et al 97: 40 Millionen Web-Seiten
- Barabasi et al 99: Domain *.nd.edu + Web-Seiten in Abstand 3
- Broder et al 00: 204 million Web-Seiten (Scan Mai und Oktober '99)



Pareto Verteilung = Verteilung nach Potenzgesetz

➤ **Diskrete Pareto (power law) Verteilung für $x \in \{1,2,3,\dots\}$**

$$P[X = x] = \frac{1}{\zeta(\alpha) \cdot x^\alpha}$$

mit konstanten Faktor $\zeta(\alpha) = \sum_{i=1}^{\infty} \frac{1}{i^\alpha}$

– auch bekannt als Riemannsches Zeta-Funktion

➤ **“Heavy tail”-Eigenschaft**

- nicht alle Momente $E[X^k]$ sind definiert
- Der Erwartungswert existiert genau dann wenn $\alpha > 2$
- Varianz und $E[X^2]$ existieren genau dann wenn $\alpha > 3$
- $E[X^k]$ ist genau dann definiert wenn $\alpha > k+1$



Pareto-Verteilung

➤ Beispiele für Potenzgesetze (= Pareto Verteilungen)

- Pareto 1897: Einkommensverteilung in der Bevölkerung
- Yule 1944: Word-Häufigkeit in Sprachen
- Zipf 1949: Größe von Städten
- Länge von Molekülketten
- Dateilängen von Unix-Dateien
-

- Zugriffshäufigkeit auf Web-Seiten
- Länge von Telefonanrufen
- ...

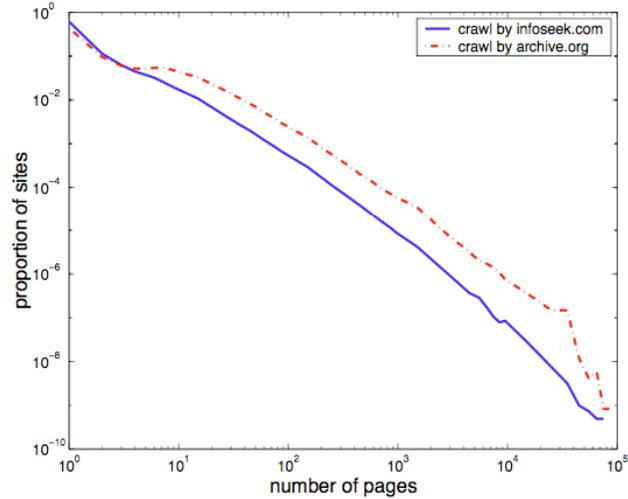


Zipf's Law and the Internet

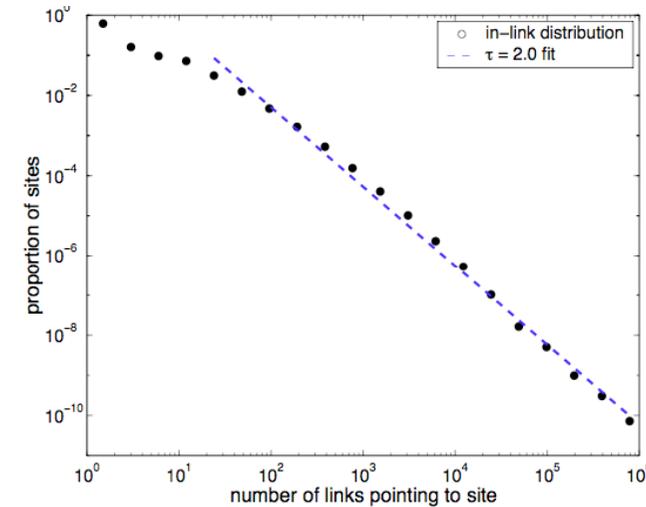
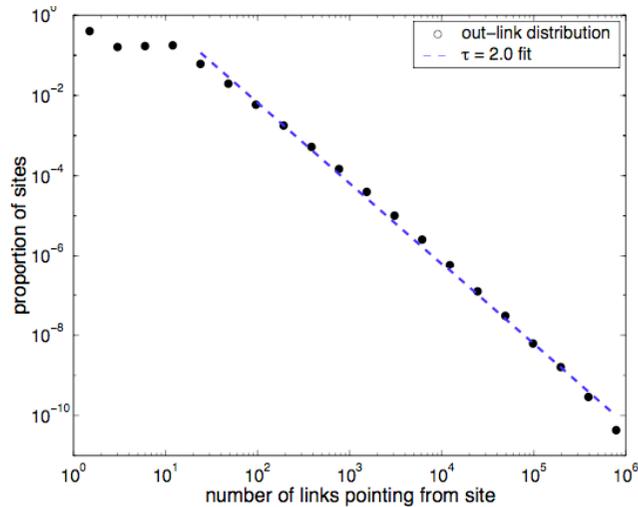
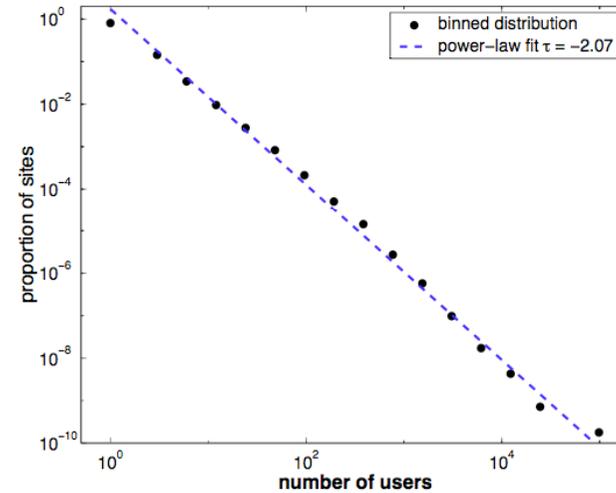
Lada A. Adamic, Bernardo A. Huberman,
2002

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelbauer

a)



b)



Pareto
Verteilung

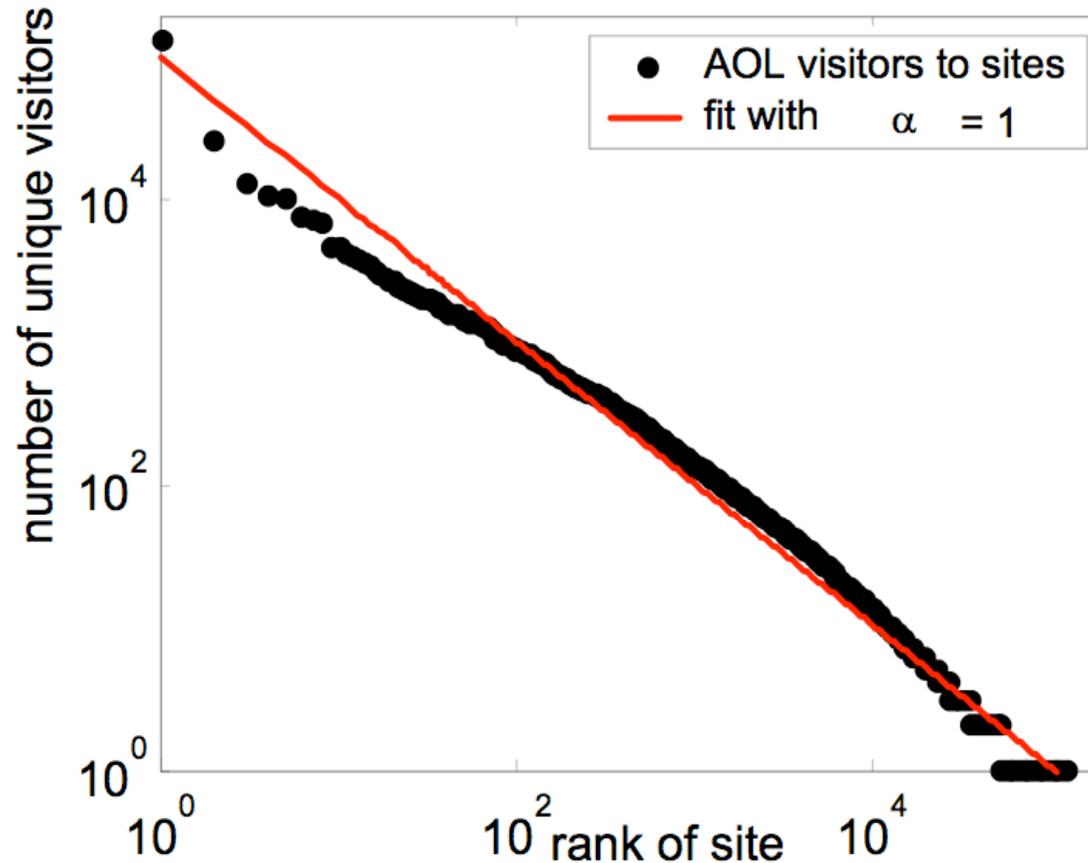
Figure 1. Fitted power law distributions of the number of site a) pages, b) visitors, c) out links, and d) in links, measured in 1997.



Zipf's Law and the Internet

Lada A. Adamic, Bernardo A. Huberman,
2002

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelbauer



Zipf Verteilung

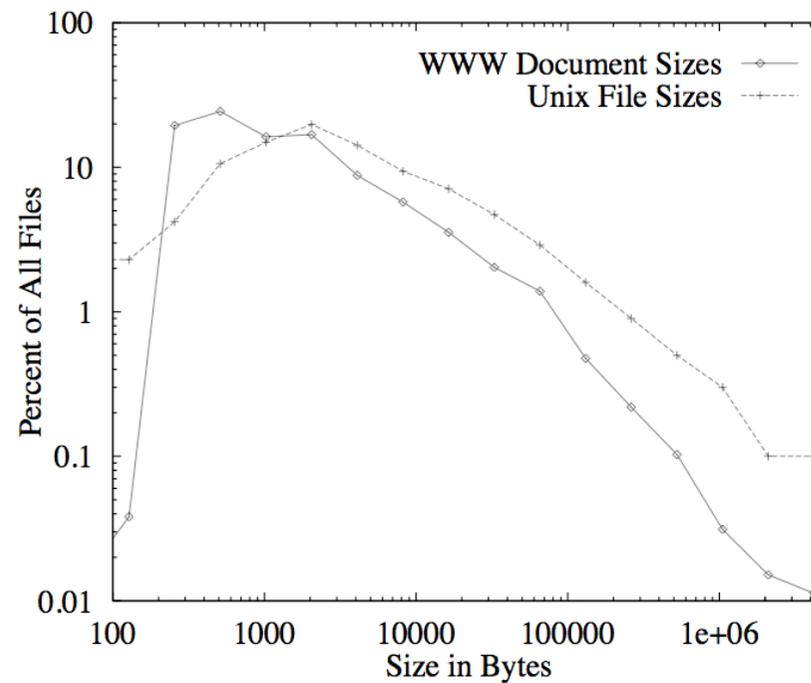
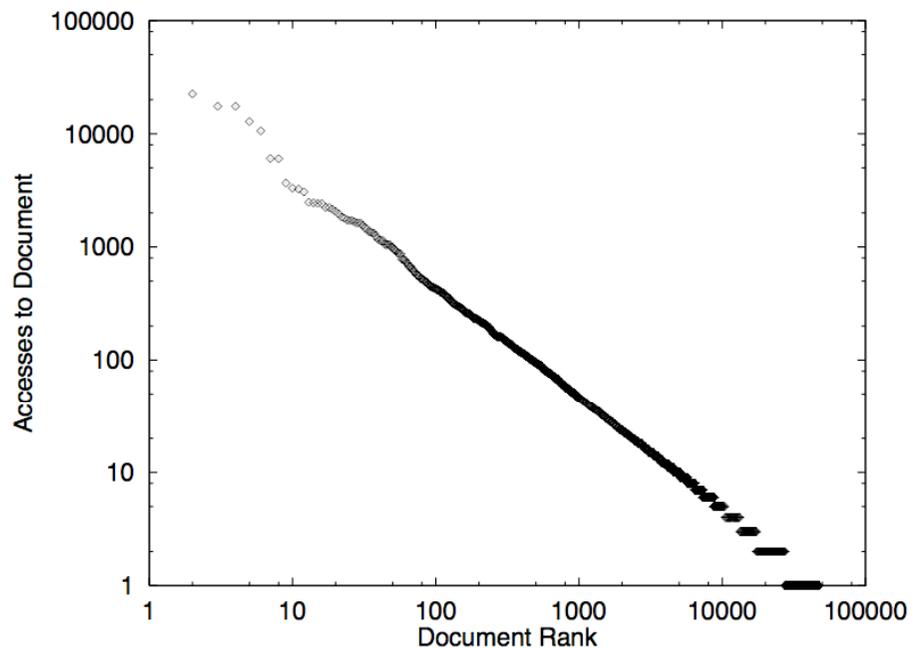
Figure 2. Sites ranked by the number of unique AOL visitors they received Dec. 1, 1997. AOL (America Online) is the largest Internet service provider in the United States. The fit is a Zipf distribution $n_r \sim r^{-1}$



Heavy-Tailed Probability Distributions in the World Wide Web

Mark Crovella, Murad, Taqqu, Azer Bestavros. 1996

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

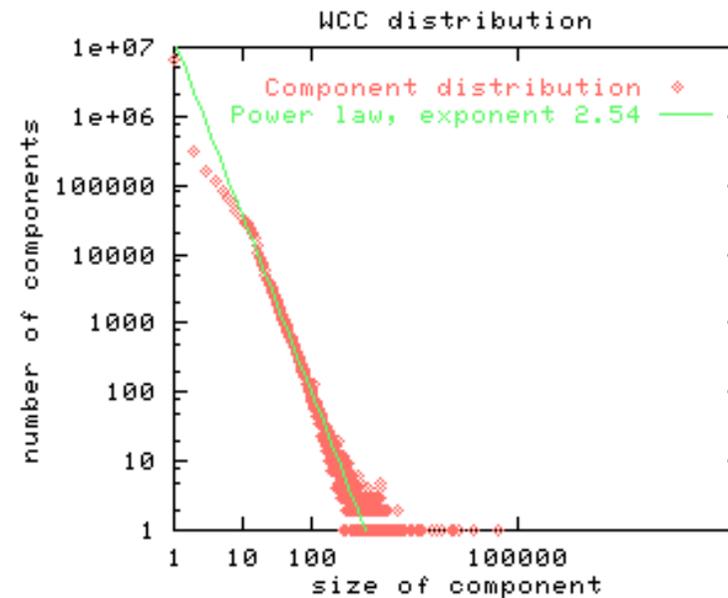
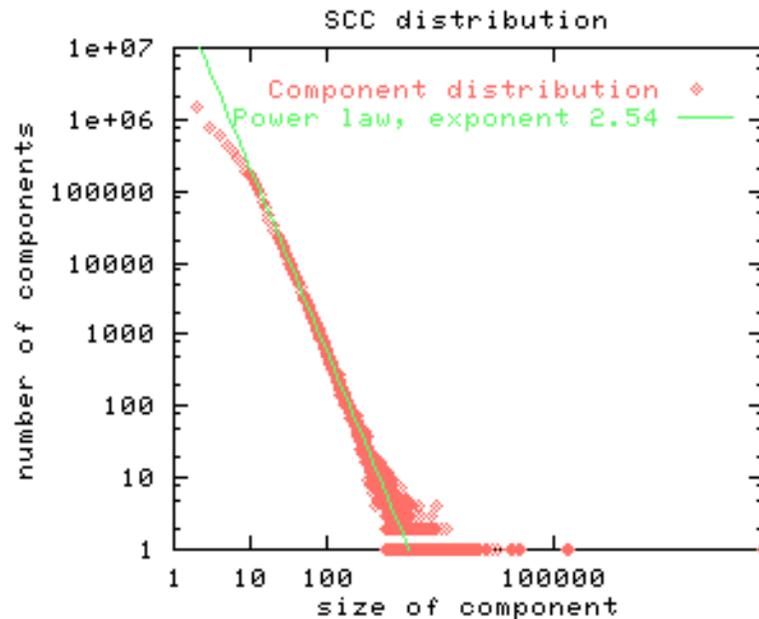




Größe der Zusammenhangskomponenten

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

- **Starke und schwache Zusammenhangskomponenten unterliegen einer Pareto-Verteilung**

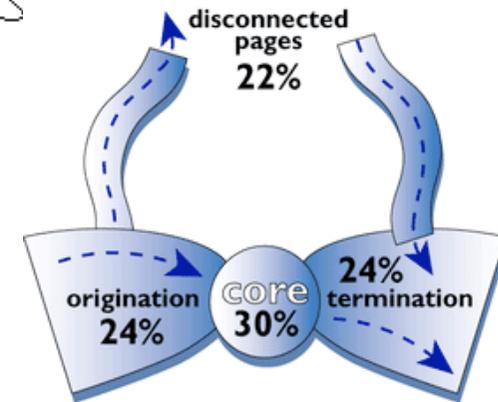
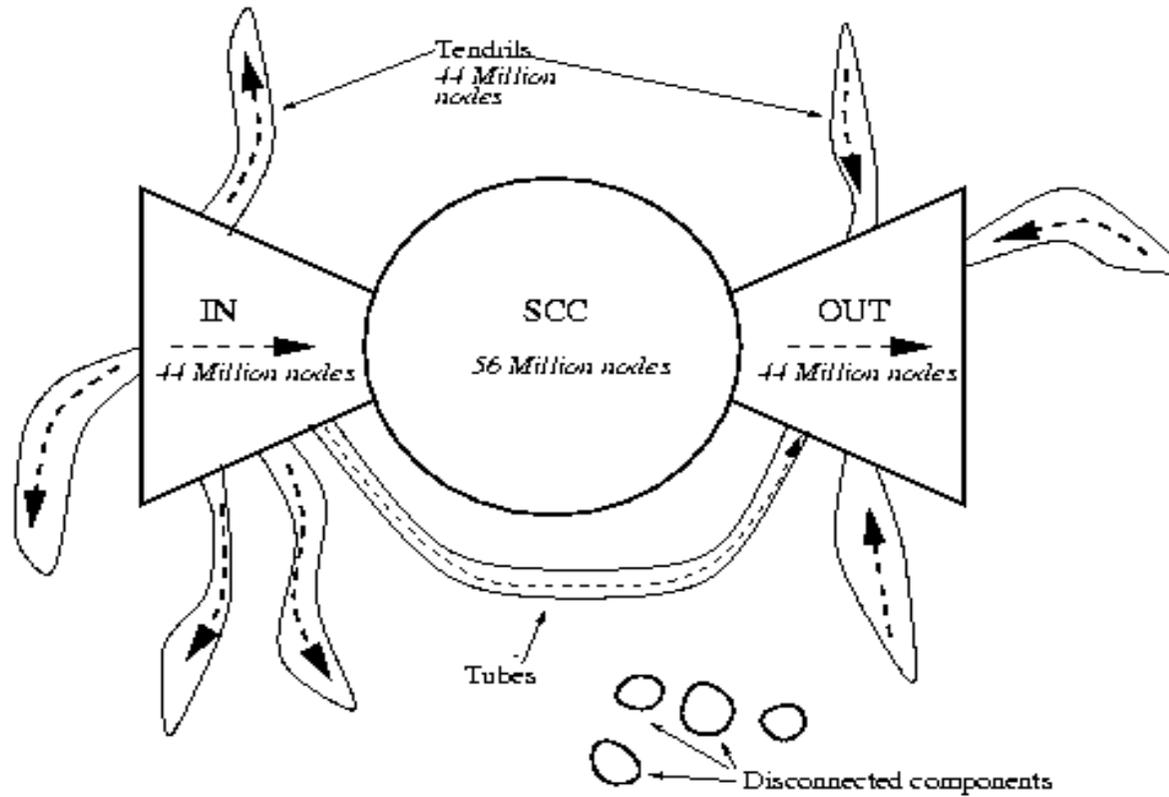


- **A. Broder, R. Kumar, F. Maghoul, P. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins, and J. Wiener. “Graph Structure in the Web: Experiments and Models.” In Proc. of the 9th World Wide Web Conference, pp. 309–320. Amsterdam: Elsevier Science, 2000.**
 - Größte schwache Zusammenhangskomponente hat 91% aller Web-Seiten
 - Größte starke Zusammenhangskomponente hat 28% aller Webseiten
 - Durchmesser ist ≥ 28



Der Web-Graph (1999)

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer





Zipf's Law and the Internet

Lada A. Adamic, Bernardo A. Huberman,
2002

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

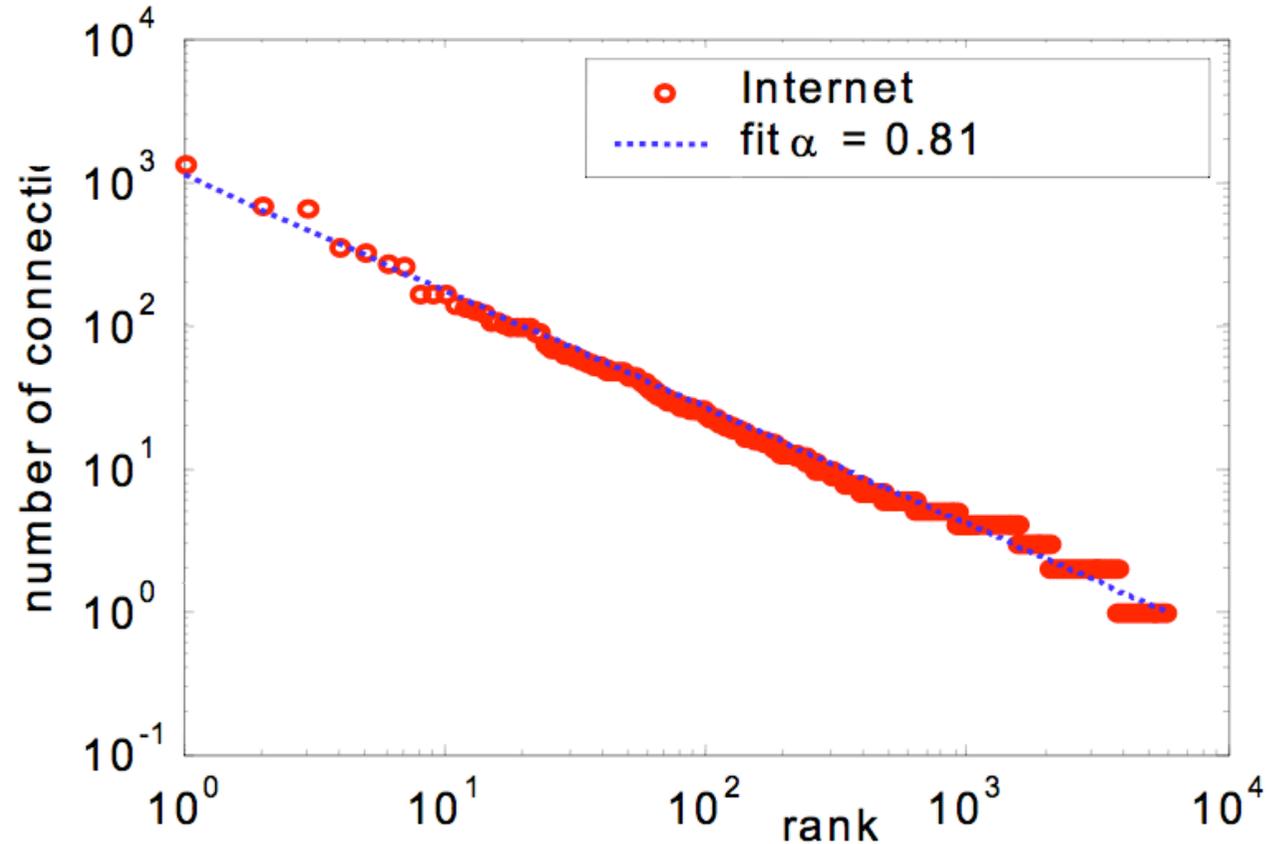


Figure 3. The connectivity of the internet backbone at the autonomous system (AS level). Each AS is itself a network corresponding to a single ISP, business entity or educational institution.



Pareto-Verteilung des Grades von ASen im Internet

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

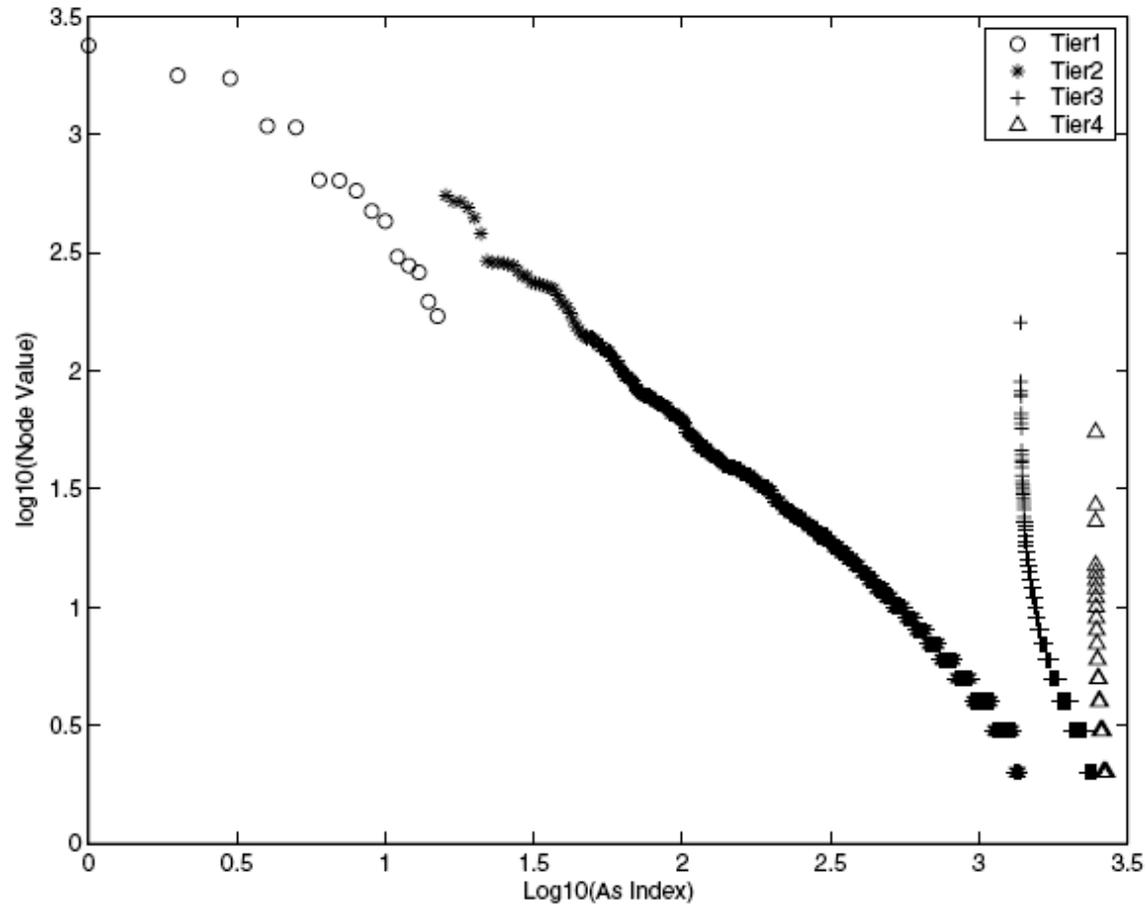


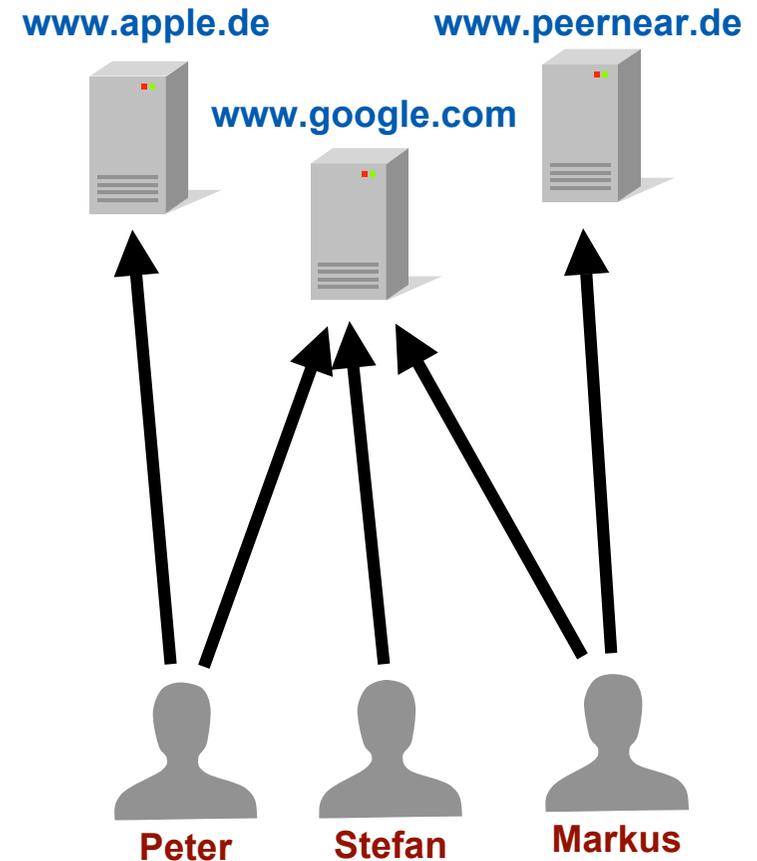
Fig. 2. Degree of ASes in different tiers

“Comparing the structure of power-law graphs and the Internet AS graph”, Sharad Jaiswal, Arnold L. Rosenberg, Don Towsley, INCP 2004



WWW-Lastbalancierung

- Für Surfen im Web typisch:
 - Web-Server bieten Web-Seiten an
 - Web-Clients fordern Web-Seiten an
- In der Regel sind diese Mengen disjunkt
- Eingehende Anforderungen belasten Web-Server hinsichtlich:
 - Übertragungsbandbreite
 - Rechenaufwand (Zeit, Speicher)





Lastanforderungen

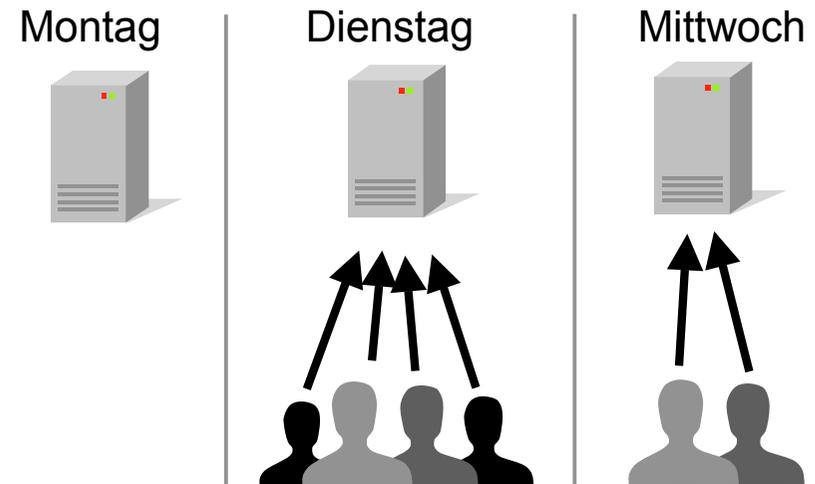
➤ Einige Web-Server haben immer hohe Lastanforderungen

- Z.B. Nachrichten-Sites, Suchmaschinen, Web-verzeichnisse
- Für permanente Anforderungen müssen Server entsprechen ausgelegt werden



➤ Andere leiden unter hohen Fluktuationen

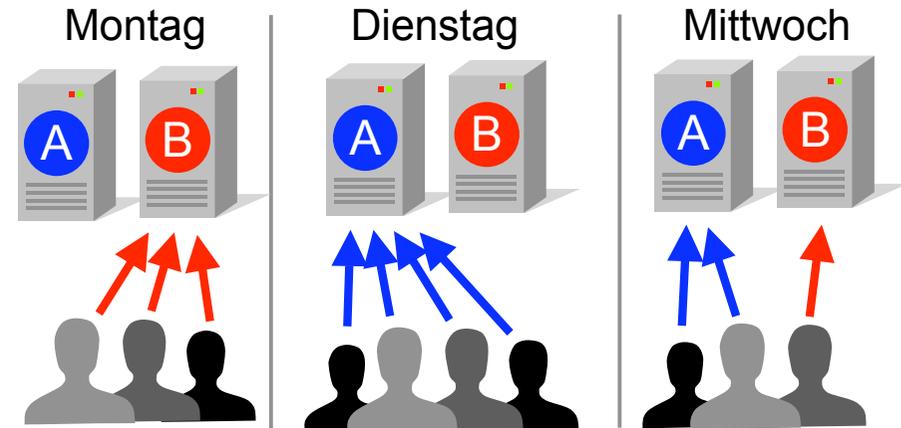
- z. B. bei besonderen Ereignissen:
 - uni-freiburg.de (550-Jahr-Feier)
 - apple.com (iPhone-Verkaufsstart)
- Server-Erweiterung nicht sinnvoll
- Bedienung der Anfragen aber erwünscht





Lastbalancierung im WWW

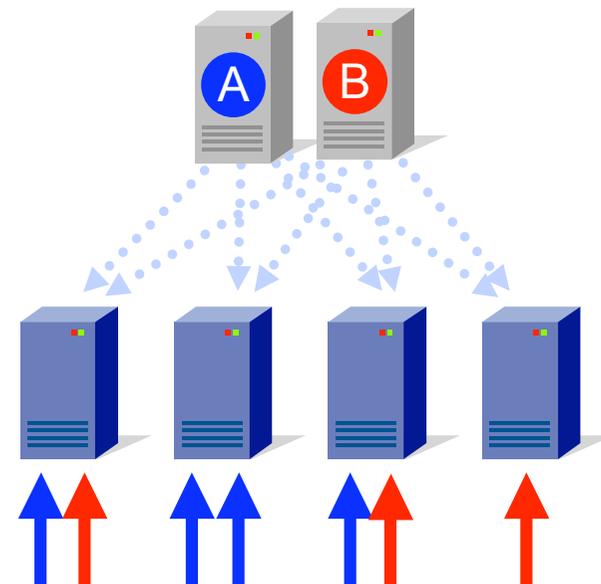
- **Fluktuationen betreffen meistens einzelne Server**



- **(Kommerzielle) Lösung**

- Dienstleister bieten Ausweich-(Cache-)Server an
- Viele Anforderungen werden auf diese Server verteilt

- **Aber wie?**



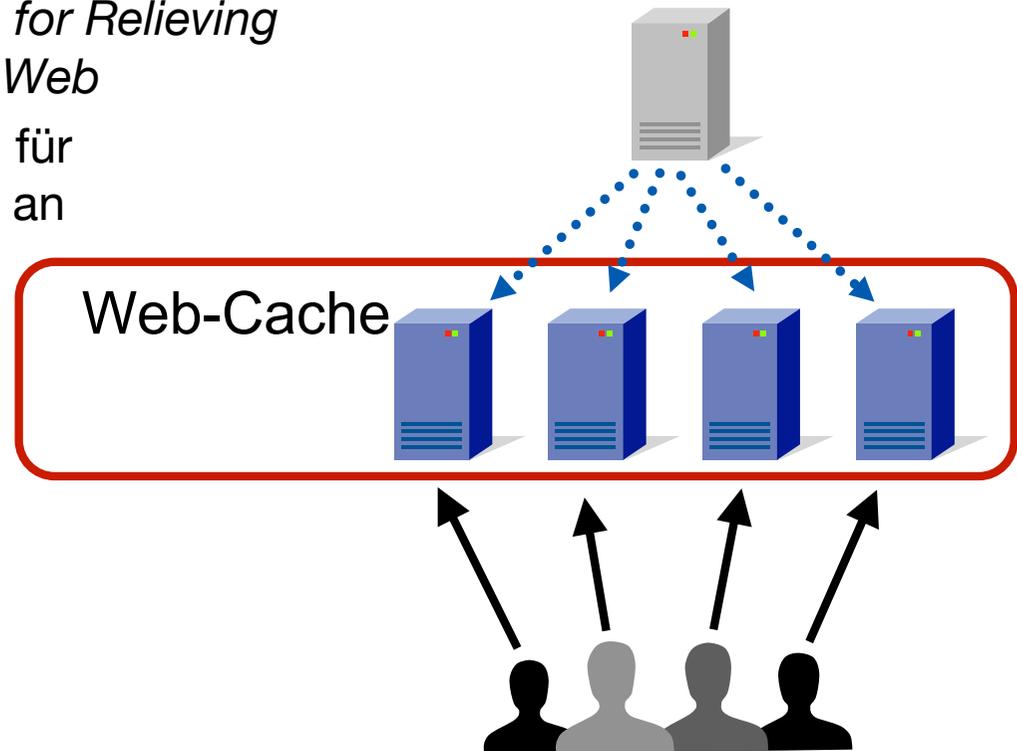


Web-Caching

➤ **Leighton, Lewin, et al. STOC 97**

- *Consistent Hashing and Random Trees: Distributed Caching Protocols for Relieving Hot Spots on the World Wide Web*
- Passen bestehende Verfahren für dynamische Hash-Funktionen an WWW-Anforderungen an

➤ **Leighton und Lewin (MIT) gründen Akamai 1997**





Ausgangssituation

➤ Ohne Lastbalancierung:

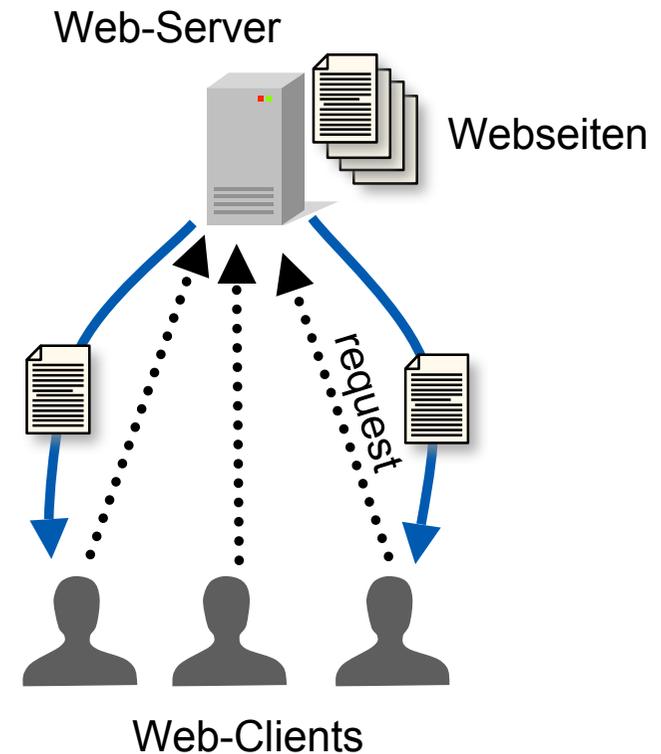
- Jeder Browser (Web-Client) belegt einen Web-Server für eine Web-Site

➤ Vorteil:

- Einfach

➤ Nachteil:

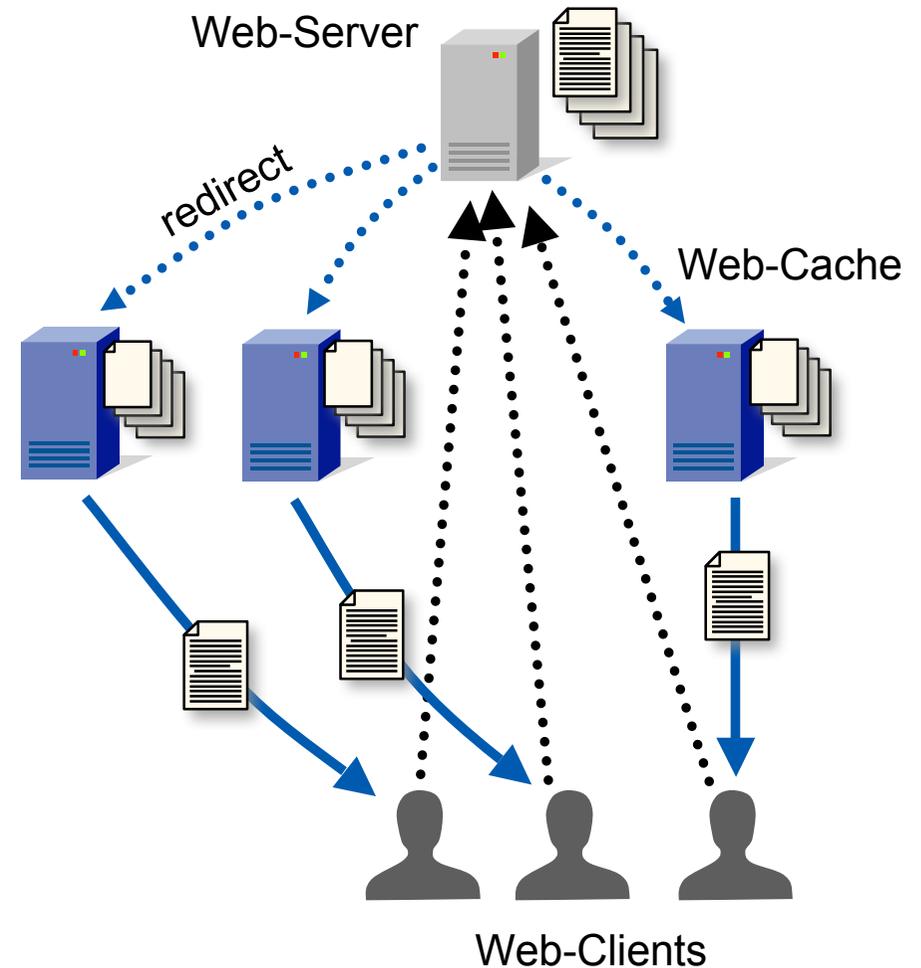
- Der Server muss immer für den Worst-Case ausgelegt werden





Site Caching

- **Ganze Web-Site wird auf verschiedene Web-Caches kopiert**
 - **Browser fragt bei Web-Server nach Seite**
 - **Web-Server leitet Anfrage auf Web-Cache um (redirect)**
 - **Web-Cache liefert Web-Seite aus**
-
- **Vorteil:**
 - Gute Lastbalancierung für Seitenverteilung
 - **Nachteil:**
 - Bottleneck: Redirect
 - Großer Overhead durch vollständige Web-Site-Replikationen

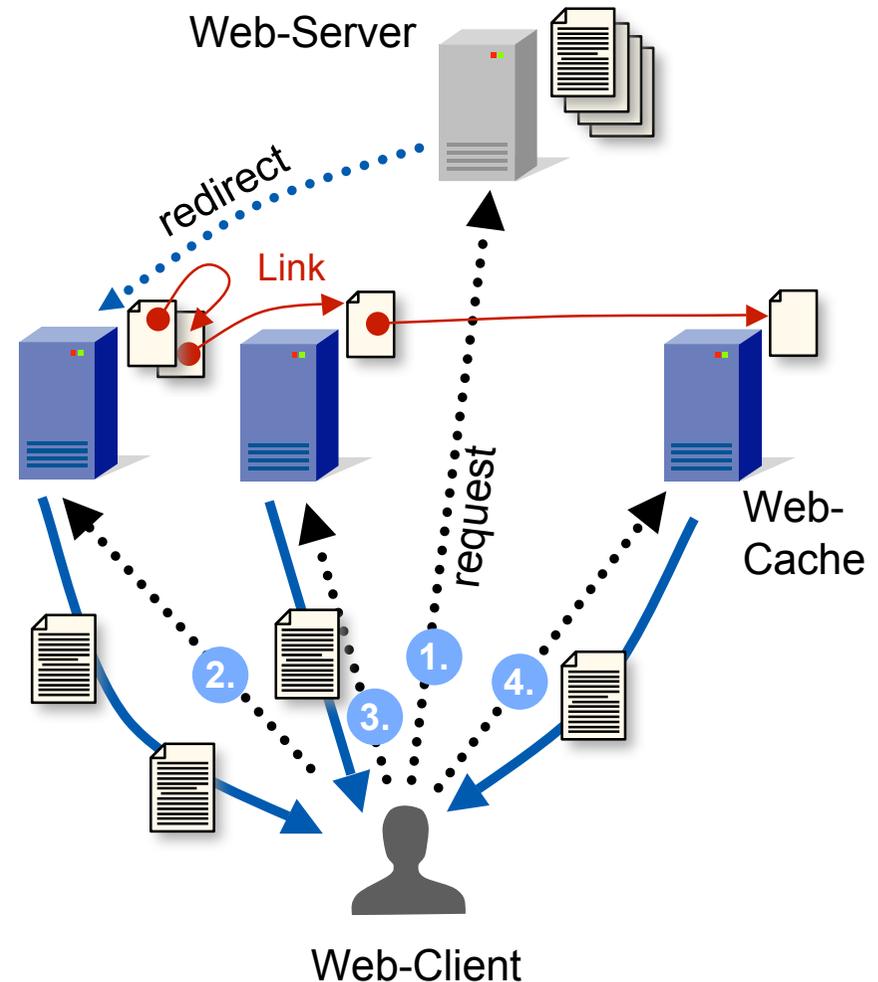




Proxy Caching

- Jede Web-Seite wird auf einige (wenige) Web-Caches verteilt
- Nur Startanfrage erreicht Web-Server
- Links verweisen auf Seiten im Web-Cache
- Dann surft der Web-Client nur noch auf den Web-Cache

- Vorteil:
 - Kein Bottleneck
- Nachteil:
 - Lastbalancierung nur implizit möglich
 - Hohe Anforderung an Caching-Algorithmus



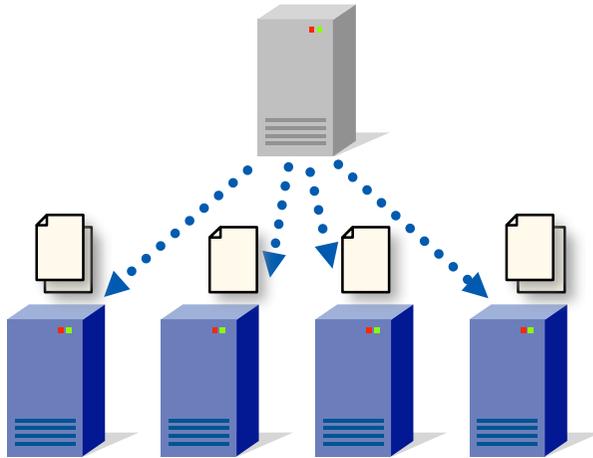


Anforderungen an Caching-Algorithmus

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

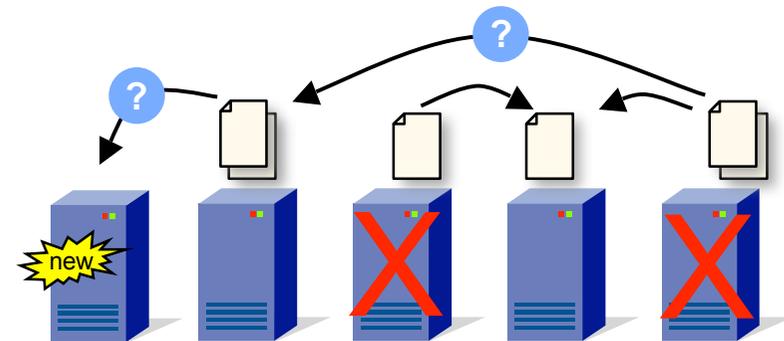
1. Balance

Gleichmäßige Verteilung der Seiten



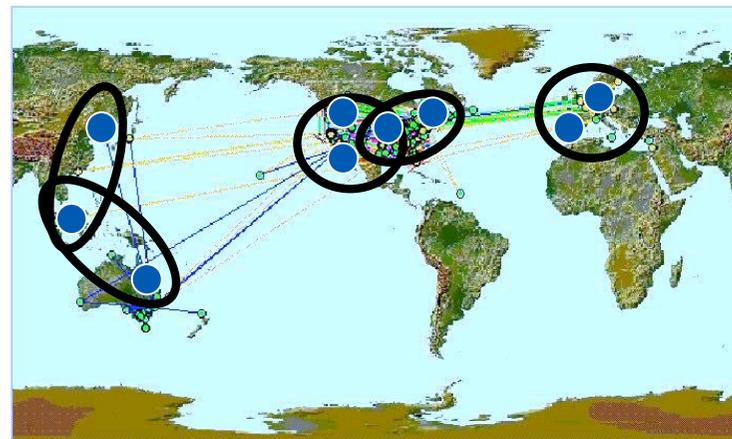
2. Dynamik

Effizientes Einfügen/Löschen von neuen Web-Cache-Servern



3. Views

Web-Clients „sehen“ unterschiedliche Menge von Web-Caches





Web 2.0

- **Ein Schlagwort, dass 2003 in O'Reilly Media aufkam**
 - Populär seit den ersten Web 2.0-Konferenzen 2004
- **Bezeichnet eine als zweite Generation**
 - web-basierter Communities und Web-dienste
 - wie Social-Networking-Webseiten, Wikis, ...
- **Der Begriff suggeriert neue Technologie**
 - tatsächlich sind die meisten Medien schon seit dem Anbeginn des Internets da
 - und wurden erst kürzlich populär

Ende der 12. Vorlesungswoche



Albert-Ludwigs-Universität Freiburg
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

Systeme II
Christian Schindelhauer
schindel@informatik.uni-freiburg.de