



## Netzbasierte Informationssysteme **Betrieb, Nutzung, Caching**

Prof. Dr.-Ing. Robert Tolksdorf  
Freie Universität Berlin  
Institut für Informatik  
Netzbasierte Informationssysteme  
mailto: [tolk@inf.fu-berlin.de](mailto:tolk@inf.fu-berlin.de)  
<http://www.robert-tolksdorf.de>



## Betriebsaspekte sehr großer Dienste

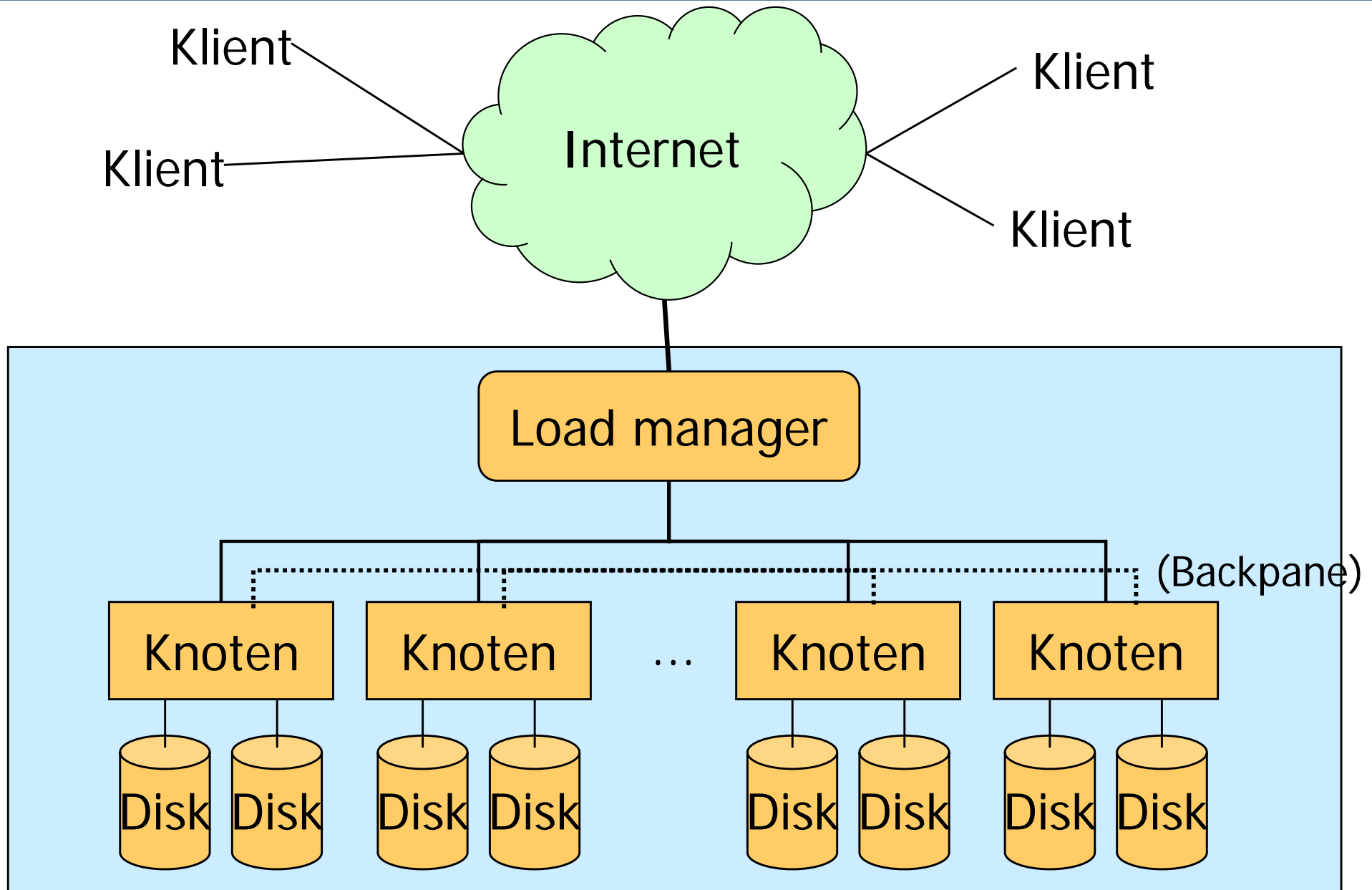
- Reale große Dienste sind um Dimensionen größer als beispielsweise [www.inf.fu-berlin.de](http://www.inf.fu-berlin.de)
  - [www.cnn.com](http://www.cnn.com)
  - [www.msn.com](http://www.msn.com)
  - [www.google.com](http://www.google.com)
  - ...
- [Brewer2001] berichtet über Erfahrungen
  - Eric Brewer (UC Berkeley) ist Mitgründer von Inktomi, Hersteller von (OEM) Suchmaschinentechnologie (2003 von Yahoo! gekauft)



- Anzahl von Suchanfragen bei Suchmaschinen
  - Nur Nutzer in USA im März 2006
  - Hochgerechnet aus Messungen an Proxies
  - [<http://searchenginewatch.com/showPage.html?page=2156461>]

<b>Searches</b>	<b>Per Day (M)</b>	<b>Per Month (M)</b>
Google	91	2,733
Yahoo	60	1,792
MSN	28	845
AOL	16	486
Ask	13	378
Others	6	166

- Vgl: ca. 33000 Zugriffe/Tag auf [www.inf.fu-berlin.de](http://www.inf.fu-berlin.de) (Durchschnitt 1.-9.11.2007)



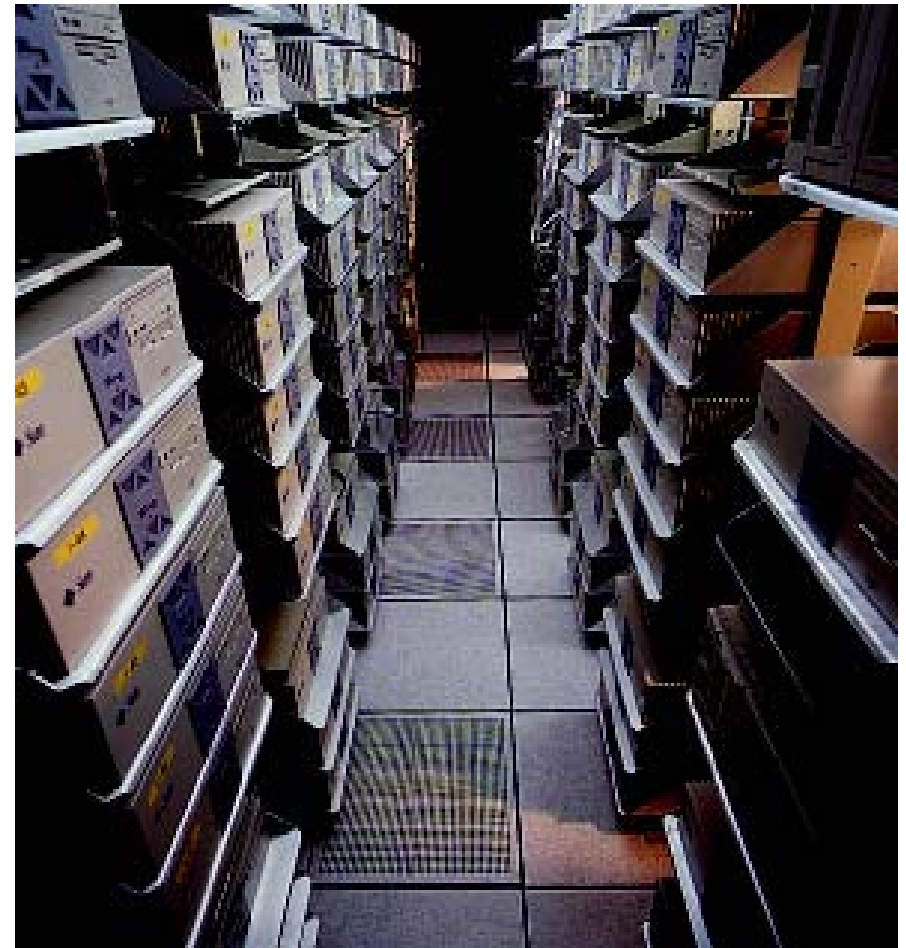
# Annahmen

---

- Annahmen
  - Klienten und Netz außerhalb der Kontrolle des Betreibers
  - Dienste werden vom Klienten aus initiiert (HTTP: GET)
  - Erheblich mehr Lesezugriffe als Schreibzugriffe
  - Cluster verwendet
- Datenhaltung
  - *Replikation*
    - Alle Knoten haben komplette Kopie des gleichen Datenbestands
  - *Partitionierung*
    - Knoten haben jeweils einen Teil des kompletten Datenbestandes
    - Alle Knoten zusammen haben kompletten Bestand
  - *Mischformen*

# Cluster

- *Cluster* (auch: Serverfarm):  
Zusammengeschaltete  
Standard-Hardware
- Google jeweils  
15000 PCs



# Cluster - Vorteile

---

- Absolute Skalierbarkeit
  - Notwendig wegen Nutzerwachstum
- Kosten
  - Cluster sind günstig
  - Netz- und Betriebskosten deutlich höher als Hardware-Kosten
- Isolierte Fehler
  - Fehler treten auf einem Rechner des Clusters auf, nicht auf einem zentralen großen Server
- Skalierung ist inkrementell
  - Knoten leicht austauschbar
  - Knoten haben ca. 3-jährige Lebenszeit
  - Ausgetauscht, wenn sie ihren Raumbedarf im Serverschrank nicht mehr rechtfertigen (Moore's Law: vierfache Leistung je Raumbedarf alle 3 Jahren)

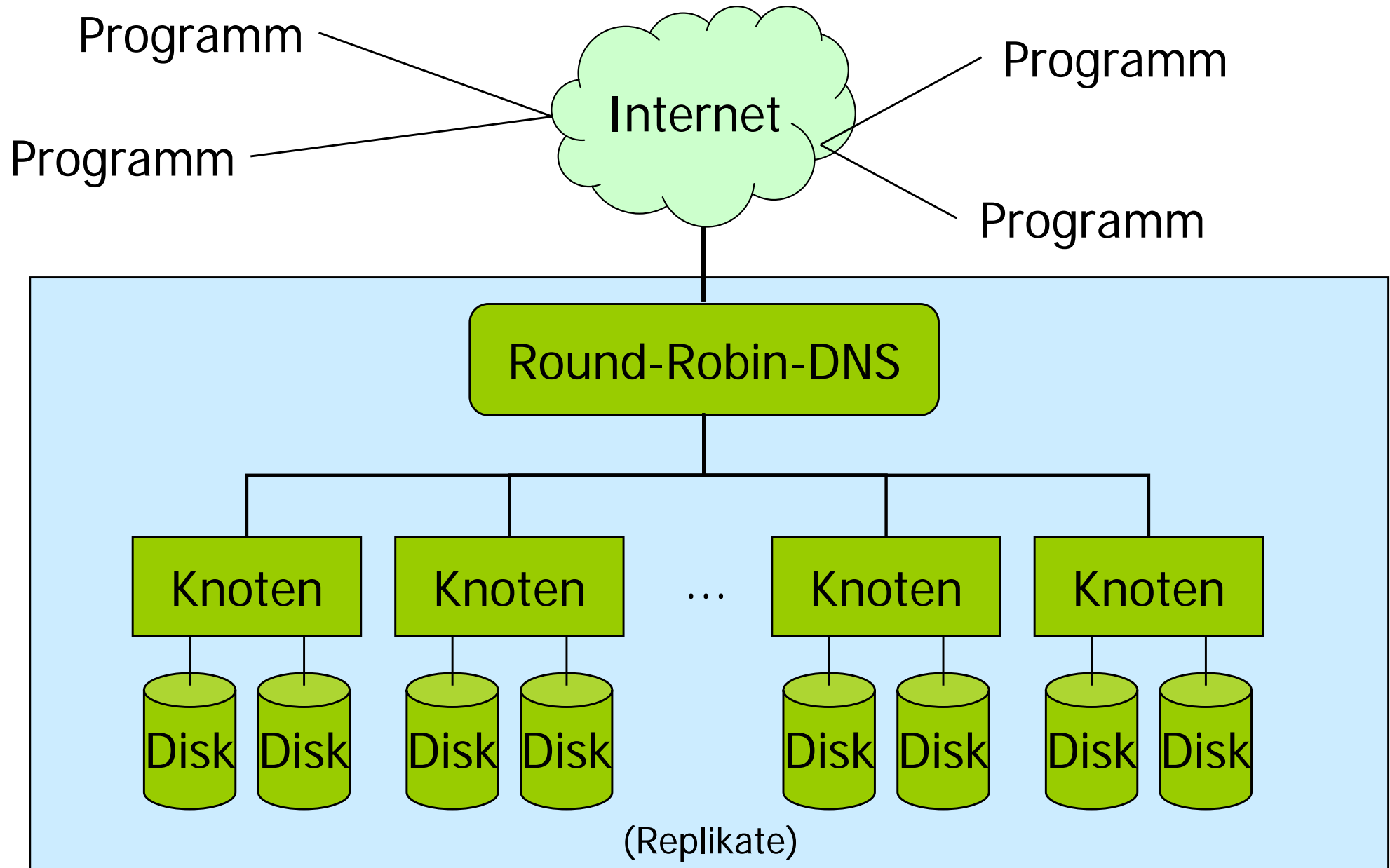


# Load management

---

- Ziel: Dienste gleichmäßig auf verfügbare Server verteilen
- Erste Ansätze: *Round-robin DNS*
  - DNS: Rechnername -> IP-Nummer
  - Idee: DNS Eintrag zyklisch ändern
  - Name wird zyklisch zu jeweils anderen Servern aufgelöst (IP-Nummer<sub>1</sub>, IP-Nummer<sub>2</sub>, ..., IP-Nummer<sub>n</sub>)
- Nachteile
  - Löst auch nach Servern auf, die gerade nicht erreichbar sind
  - Klienten kann auf gleicher Nummer Versuche wiederholen
  - Kurzes Time-to-live von DNS-Einträgen -> Viele DNS-Lookups
  - Nicht zuverlässiges Klientenverhalten

# Einfache Web-Farm mit Round-Robin-DNS

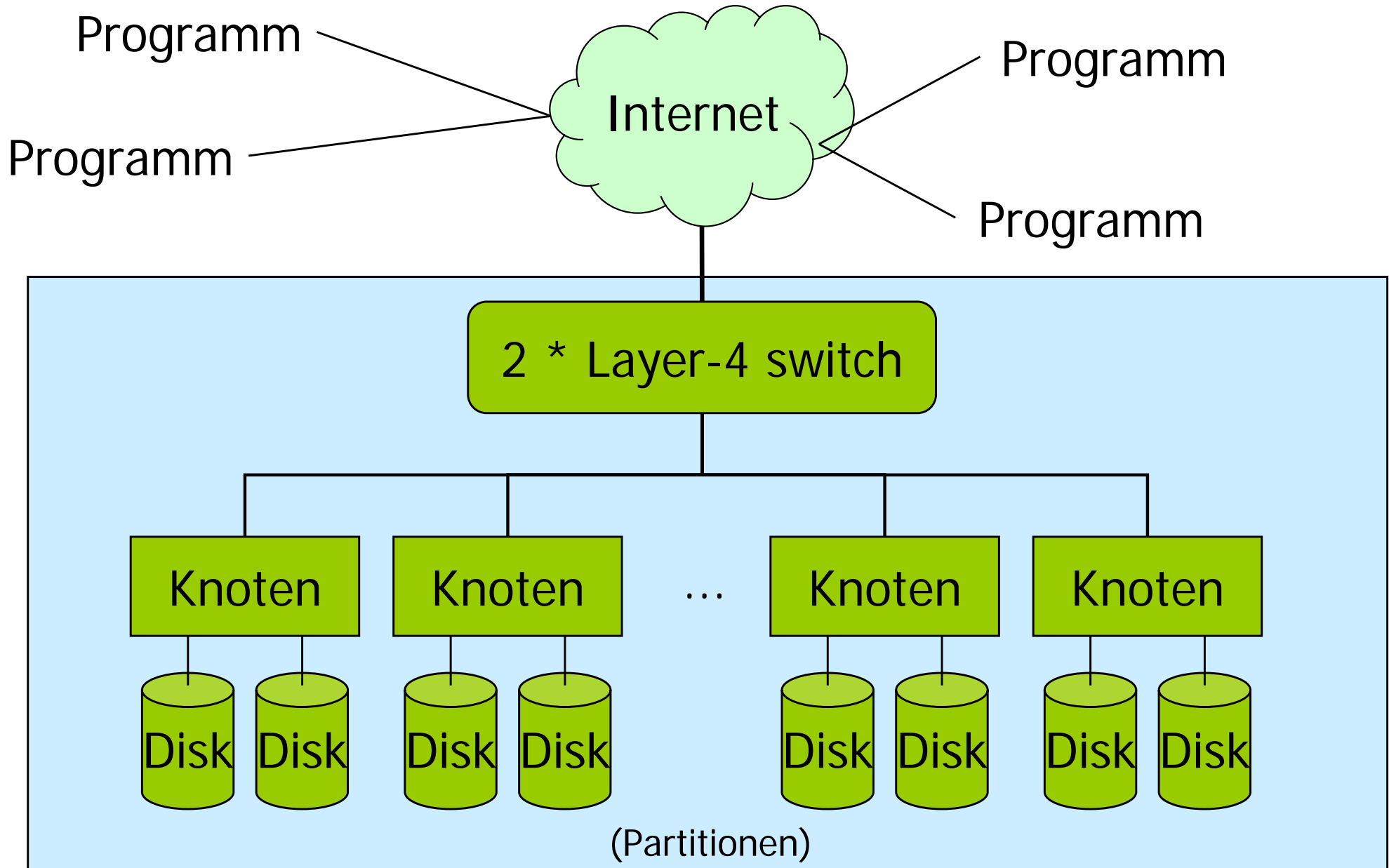


# Load management

---

- Layer-4-switches
  - Hardware die TCP "versteht"
  - Switch leitet Pakete aufgrund von TCP-Dienste-Feld und Portnummern an unterschiedliche Server weiter
- Layer-7-switches
  - Hardware die HTTP "versteht"
  - Können URLs mit Netzbandbreite parsieren und leiten Pakete entsprechend weiter
- Meistens als Paar vorhanden
- >20Gbits/s Durchsatz
- Automatisches Monitoring von Knoten

# Einfache Web-Farm mit Layer-4 Switch



# Verfügbarkeit

---

- Zentrale Anforderung an große Dienste:

## Verfügbarkeit (Availability)

- Gemessen in "Neunern":
  - Vier Neuner: 0,9999 Verfügbarkeit (<60 Sek. Ausfall/Woche)
  - Fünf Neuner: 0,99999
- Ähnlich geleitete Systeme:
  - Telefonsystem
  - Zugverkehr
  - Wasserversorgung

## Weitere Maße

- MTBF: Mean-time-between-failures, mittlerer zeitlicher Abstand zwischen Fehlern
- MTTR: Mean-time-to-repair, mittlere Dauer der Fehlerbehebung

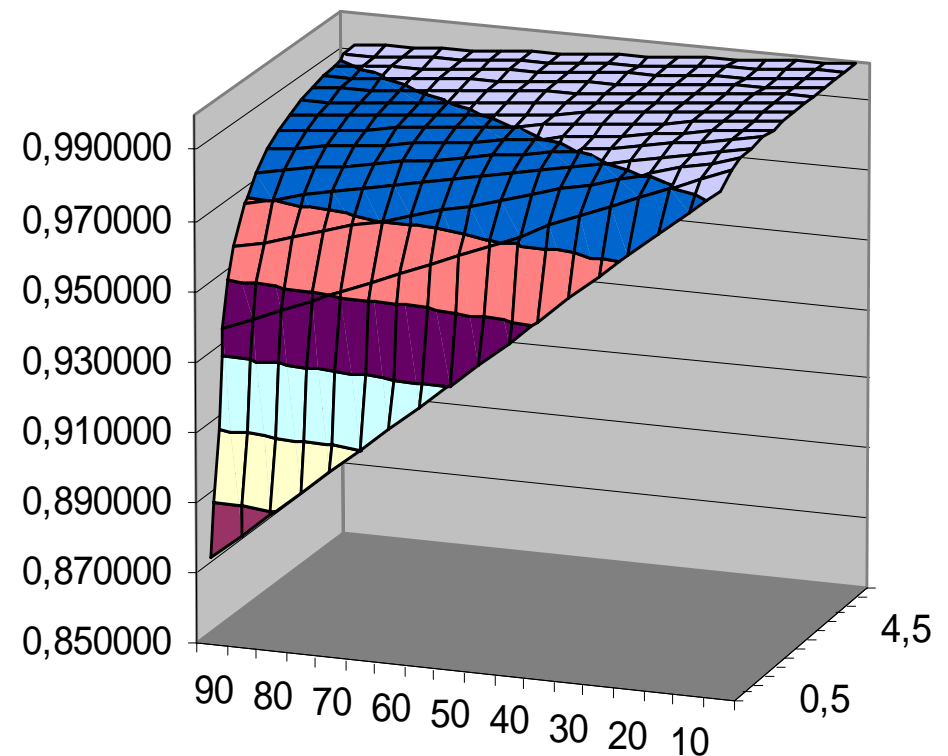
- $$\text{uptime} = \frac{(\text{MTBF} - \text{MTTR})}{\text{MTBF}}$$

- MTBF = 6 Tage (518400s), MTTR = 20 Min (1200s)

$$\text{uptime} = \frac{(518400\text{s} - 1200\text{s})}{518400\text{s}} = 0,997685$$

## Weitere Maße

- Uptime erhöhen durch:
  - Erhöhung der MTBF
  - MTTR senken
- MTBF = 7 Tage  
 MTTR = 20 Min  
 uptime=0,998016
- MTBF = 6 Tage  
 MTTR = 10 Min  
 uptime= 0,998843



- MTBF schwerer zu ermitteln, MTTR klarerer Prozess
- MTTR stabiler auch bei neuen Komponenten
- Fokus auf MTTR, Best-Effort bei MTBF

# Yield und Harvest

- Weiteres Maß: Yield – wie viele Anfrageergebnisse?

- $$\text{yield} = \frac{\text{bearbeitete Anfragen}}{\text{gestellte Anfragen}}$$

- entspricht Nutzererfahrung
- gewichtet Uptime-Sekunden

- Weiteres Maß: Harvest – welcher Teil der Datenbank ist nutzbar?

- $$\text{harvest} = \frac{\text{zugreifbare Daten}}{\text{gesamte Daten}}$$

- Erweiterbar zum Anteil der nutzbaren Dienste



# Yield und Harvest

---

- Ziel: 100% Yield und 100% Harvest
- Anpassbar:
  - Fehler bei Replikaten
    - Sinkendes Yield
    - Gleichbleibendes Harvest
  - Fehler bei Partitionen
    - sinkendes Harvest
    - Gleichbleibendes Yield

# DQ (Data/Query) Prinzip

---

- Daten pro Anfrage (D)  $\times$  Anfragen pro Sekunde (Q)  $\rightarrow$  C
- Reflektiert physikalische Beschränkung des Gesamtsystems
  - I/O Bandbreite
  - Plattengeschwindigkeit
  - ...
- Ist bandbreitenorientiert
- Höheres DQ bewirken
  - Mehr Knoten (linear, also Veränderungen leicht testbar)
  - Software-Optimierungen
- Niedrigeres DQ bewirken
  - Fehler
- Relatives DQ entscheidend
  - Auswirkung von Fehlern ermittelbar
  - Zielvorgaben für Optimierungen/Ausbau ermittelbar

# Replikation vs. Partitionierung

---

- Fehleranalyse: 1 Ausfall bei 2 Knoten
  - Replikation
    - Harvest bleibt bei 100%
    - Yield fällt auf 50%
    - D bleibt gleich, Q halbiert sich
  - Partitionierung
    - Harvest fällt auf 50%
    - Yield bleibt bei 100%
    - D halbiert sich, Q bleibt gleich
- Replikation und Partitionierung sind nach DQ Prinzip gleich

# Replikation vs. Partitionierung

---

- Replikation benötigt zusätzlich DQ Kapazität zum Zugriff auf Replikat
- Partitionierung aber auch kein Vorteil: Keine Ersparnis durch Einsparung von Datenplatz, weil DQ eigentlicher Kostenfaktor ist
- Nach anfänglicher Partitionierung Übergang zu mehr Replikation
  - Harvest steuerbar
  - Datenverlustwahrscheinlichkeit steuerbar
  - Skalierung leichter handhabbar

# Kontrollierter Leistungsrückgang

---

- Graceful degradation:  
Bei hoher Last Verfügbarkeit erhalten
- Daten:
  - Hohe Last zu Normallast zwischen 1.6:1 und 6:1
    - Vorhaltung von Leistung zu teuer
    - z.B. SMS an Silvester
  - Extremlast durch einzelne Ereignisse
    - Vorgehaltene Leistung immer noch zu klein
    - z.B. Kartenreservierung Star Wars
  - Zusammenhängende Fehler
    - z.B. Stromausfall



CNN.com - Netscape

# CNN.com

BREAKING NEWS



(AP PHOTO)

## AMERICA UNDER ATTACK

At 8:45 a.m. EDT, the first of two airliners crashed into the World Trade Center, opening a horrifying and apparently coordinated terrorist attack on the United States, which saw the collapse of the two 110-story towers into surrounding Manhattan streets and a later attack on the Pentagon.

[DEVELOPING STORY »](#)

**COMPLETE COVERAGE:**

- [Chronology of terror](#)
- President Bush: U.S. will 'hunt down and punish those responsible'
- World Trade Centers [collapse](#) after planes hit, 10,000 emergency workers head to scene
- Plane hits [Pentagon](#), part of the Pentagon collapses
- American, United both confirm losing two planes each
- Federal buildings, United Nations evacuated
- [FAA grounds](#) all U.S. flights, sends trans-Atlantic flights to Canada
- Sen. John McCain calls attacks 'act of war'
- Israel evacuates embassies
- Non-essential NATO employees asked to leave Brussels HQ
- Taliban issues [statement](#) to tell U.S. 'Afghanistan feels your pain'
- Kennedy Space Center, LAX, [Disney](#) Florida parks closed
- [World shock over U.S. attacks](#)



CNN.com - Netscape

Join or Sign In

# CNN.com

[CNNmoney](#)  
click here!

CNN Europe | CNN Asia | Languages | On CNN TV | Transcripts | Headlines

SEARCH CNN.COM:  GO

Updated: 07:42 a.m. EST (12:42 GMT) January 2, 2003



(AP PHOTO)

## Edwards in 2004?

Democratic Sen. John Edwards said today that he's taking the first step toward a 2004 bid for the White House by forming an exploratory committee. The North Carolina lawmaker joins Vermont Gov. Howard Dean and Massachusetts Sen. John Kerry as likely Democratic candidates.

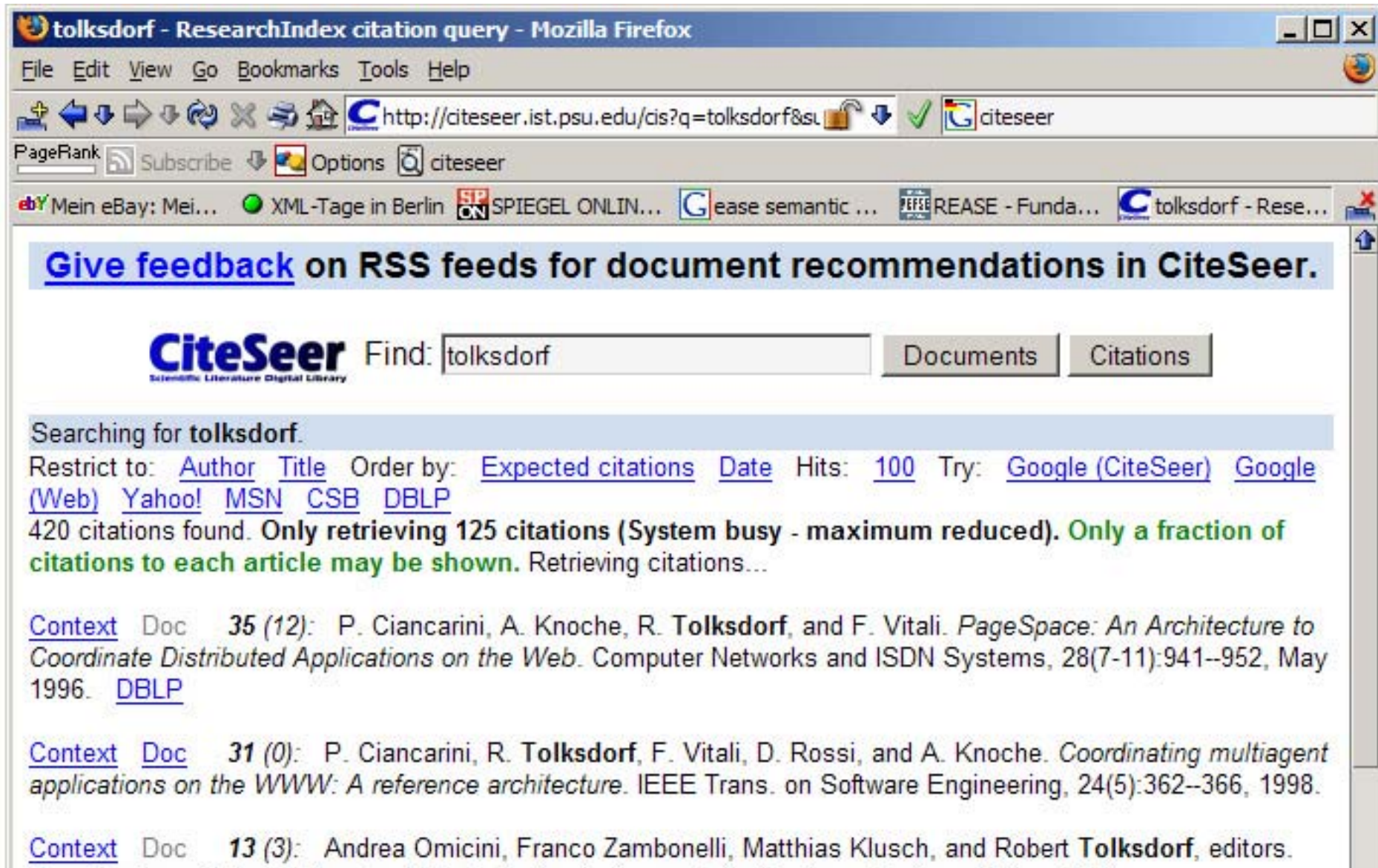
[FULL STORY](#)

- Video: [Profile of Sen. Edwards](#)
- Interactive: [Democratic contenders](#)

WAKE UP WITH CNN's AM E-MAIL [SIGN UP](#)



# citeseer mit beschränkter Antwortmenge



tolksdorf - ResearchIndex citation query - Mozilla Firefox

File Edit View Go Bookmarks Tools Help

http://citeseer.ist.psu.edu/cis?q=tolksdorf&sl

PageRank Subscribe Options citeseer

Mein eBay: Mei... XML-Tage in Berlin SPIEGEL ONLIN... ease semantic ... REASE - Funda... tolksdorf - Rese...

[Give feedback](#) on RSS feeds for document recommendations in CiteSeer.

**CiteSeer** Find:  Documents Citations

Searching for **tolksdorf**.

Restrict to: [Author](#) [Title](#) Order by: [Expected citations](#) [Date](#) Hits: [100](#) Try: [Google \(CiteSeer\)](#) [Google \(Web\)](#) [Yahoo!](#) [MSN](#) [CSB](#) [DBLP](#)

420 citations found. **Only retrieving 125 citations (System busy - maximum reduced). Only a fraction of citations to each article may be shown.** Retrieving citations...

[Context](#) Doc **35** (12): P. Ciancarini, A. Knoche, R. **Tolksdorf**, and F. Vitali. *PageSpace: An Architecture to Coordinate Distributed Applications on the Web*. Computer Networks and ISDN Systems, 28(7-11):941--952, May 1996. [DBLP](#)

[Context](#) [Doc](#) **31** (0): P. Ciancarini, R. **Tolksdorf**, F. Vitali, D. Rossi, and A. Knoche. *Coordinating multiagent applications on the WWW: A reference architecture*. IEEE Trans. on Software Engineering, 24(5):362--366, 1998.

[Context](#) Doc **13** (3): Andrea Omicini, Franco Zambonelli, Matthias Klusch, and Robert **Tolksdorf**, editors.

- D (Daten pro Anfrage) verkleinern um Q (Anfragen pro Sekunde) zu erhöhen
  - Datenbank verkleinern
  - Dadurch Yield erhalten
  - siehe CNN am 11.9.01
- Q limitieren um D zu erhalten
  - Durch Zugangsbeschränkungen Q senken
    - Anfragekosten schätzen, aufwändige verbieten
    - Bestimmte Anfragen priorisieren
    - Anfragekosten durch mindere Datenqualität senken
  - Dadurch Harvest erhöhen



# Upgrades / Wartung

---

- Grosse Dienst müssen fortlaufend gewartet werden
  - Software-Erweiterungen/-Korrekturen
  - Hardware-Erweiterungen/-Ersatz
- Wartung ist kontrollierter Ausfall
- Verlust an DQ Kapazität

# Upgrades / Wartung

---

- Schneller Neustart
  - Maschinen werden in neue Konfiguration gebootet
  - Yield geht verloren
  - Optimiert durch geeigneten Zeitpunkt (off-peak)
- Rolling Upgrade
  - Jeweils ein Knoten upgraden
  - Replikation: Yield sinkt minimal
  - Partitionierung: Harvest sinkt
  - Zwei Versionen müssen verträglich koexistieren
- „Big Flip“
  - Halber Cluster wird neu gestartet, danach andere Hälfte
  - Durch Layer-4 Switch transparent nach aussen
  - 50% DQ Verlust

- [www.google.com](http://www.google.com) wird per DNS auf jeweils nächstes Rechenzentrum geleitet
- In einem Google Rechenzentrum wird die Last der Rechner von Hardware überwacht und balanciert
- Replizierte Index- und Dokumentenserver werden befragt
- Replikation und Datenunabhängigkeit ist Basis für Nebenläufigkeit
- Google-Cluster
  - Fehlertolerante Hardware (Netzteile, Platten...)
  - Verfügbarkeit und Fehlertoleranz durch Replikation
  - Leistungs/Preis Verhältnis bestimmt CPU Auswahl

- Sehr große Dienste sind durch sehr große Cluster realisiert
- Lastverteilung durch spezielle Switches
- Verfügbarkeit zentral
- Bewegte Daten pro Sekunde als Obergrenze von Systemen (DQ Prinzip)
- DQ Prinzip: Replikation besser als Partitionierung
- DQ Prinzip zur Modellierung von Ausnahmemanagement
- Upgrades unterschiedlich handhabbar

# Literatur

---

- Eric A. Brewer. *Lessons from Giant-Scale Services*. IEEE Internet Computing. July/August 2001. 46-55.
- Matrix.Net. *Internet Withstands Attack on America*. Presseerklärung.  
<http://www.infohq.com/Computer/computer-news-Sep01-8-16.htm#matrixnet>
- Barroso, L. A., Dean, J., and Hölzle, U. 2003. Web Search for a Planet: The Google Cluster Architecture. *IEEE Micro* 23, 02 (Mar. 2003), 22-28.  
<http://labs.google.com/papers/googlecluster-ieee.pdf>

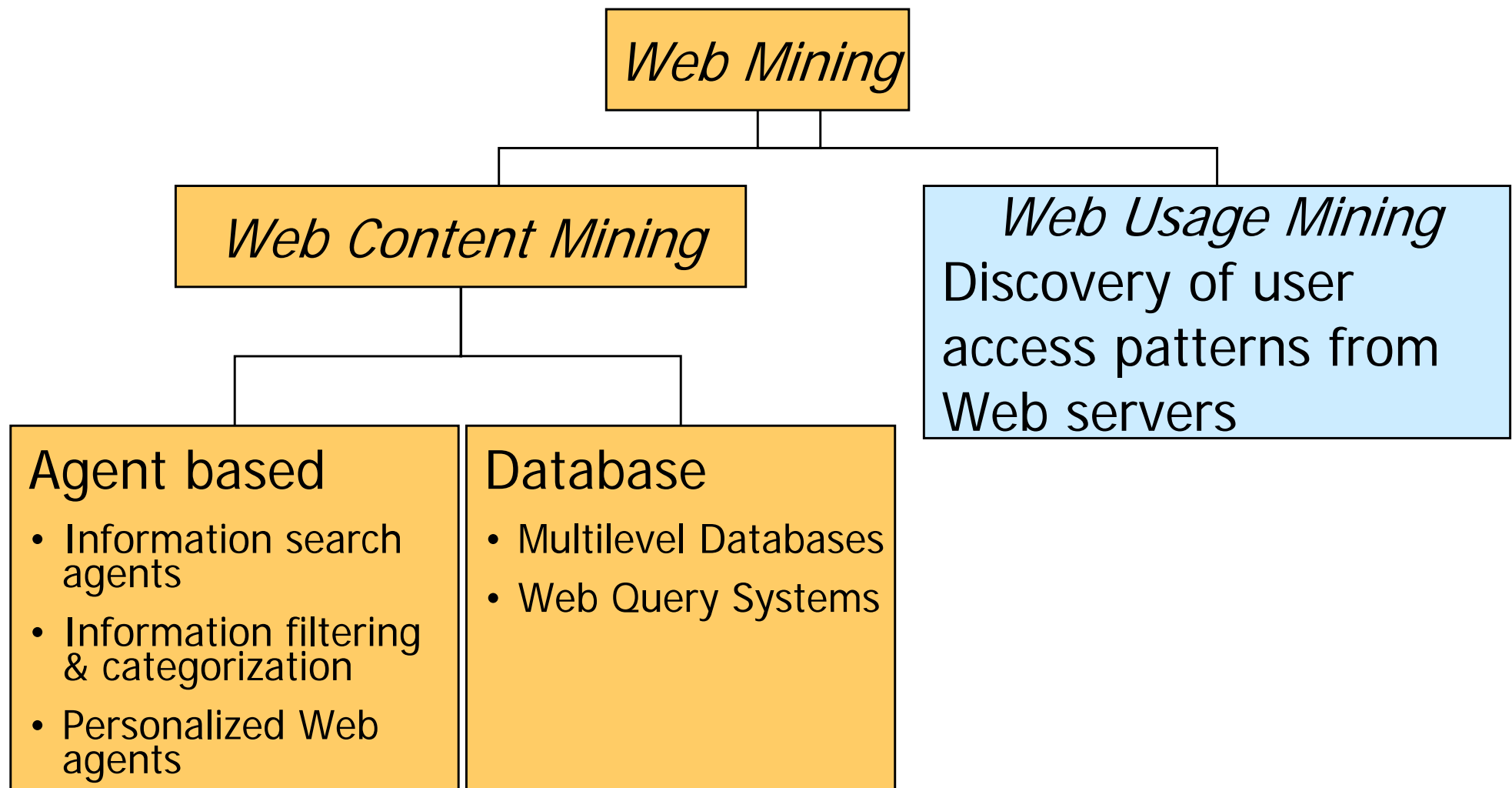


## Nutzung und Nutzer von Web-Sites

- Nutzer von Web-Sites sind für den Server anonym
  - Keine Identifikation des tatsächlichen Rechners: Proxies, Caches, private Netze, dynamische IP-Nummern
  - Keine Identifikation des Nutzerprozesses: Mehrbenutzerrechner, Proxies, Caches
  - Keine Identifikation des Nutzers: Account-Informationen lokal
- Informationen über Nutzer sind aber nützlich
  - Personalisierung
  - Optimierung des Angebots
  - Grundlage des Geschäftsmodells

# Web Usage Mining

- *Web Mining*: The discovery and analysis of useful information from the Web





# Logfiles auf Web-Servern

- Logfiles werden zeilenweise geschrieben
- Mögliches Format: Common Logfile Format
  - remotehost: IP-Nummer oder Name des Client-Rechners
  - rfc1413: Nutzer-ID auf Quellrechner (ident Dienst)
  - authuser: Nutzer-ID für Web-Session
  - [date]: Datum des Eintrags
  - "request": HTTP-Request Zeile
  - status: HTTP Antwortcode
  - bytes: Größe der Antwort
- - - - [19/Dec/2002: 10: 07: 30 +0100] ↵  
"GET /~tol k/cool 2. gi f HTTP/1. 1" 200 4942
- - - - [19/Dec/2002: 10: 08: 06 +0100] ↵  
"GET /~tol k/%22http: /www. dcs. ed. ac. uk/home/cdw/ ↵  
phdproj ect/SECD/Appl et/I i spki t. html %22 HTTP/1. 1" 404 -

# Logfiles auf Web-Servern

---

- Extended Common Logfile Format
  - CLF Felder
  - "referer": Seite von der Link verfolgt wurde
  - "user agent": Client-Software
- - - - [19/Dec/2002: 10: 07: 30 +0100] ↵  
"GET /~tol k/cool 2. gi f HTTP/1. 1" 200 4942 ↵  
"http: //grunge. cs. tu-berl i n. de/~tol k/vml anguages. html " ↵  
"Mozi l l a/4. 0%20(compati bl e; %20MSI E%206. 0; %20Wi ndows%20NT%  
205. 1) "

# Logfiles auf Web-Servern

---

- Probleme:
  - remotehost:  
Nummer des Rechners, der einen Socket zum Server aufbaut ist noch nicht Rechner an dem der Nutzer ist
  - rfc1413:  
Läuft ident-Dienst überhaupt? Was soll man mit Ergebnis anfangen?
  - [date]:  
Nicht eindeutig bei vielen Zugriffen in kurzen Abständen
  - "request":  
GET mit IfModifiedSince-Header, Caches
  - "referer":  
Nicht bei Direkteingabe, Bookmarks
  - "user\_agent":  
Keine zuverlässige Angabe, was ist mit Crawlern?

# Messgrößen

---

- Auf Basis von Logfiles lassen sich verschiedene Aussagen über die Nutzung einer Site treffen
- Insbesondere sind diese Aussagen Basis für die Preisfindung der Werbewirtschaft
- Diese Aussagen sind von unterschiedlicher Güte

# Messgrößen

---

- Hits
  - Anzahl der Abrufe von Informationen
  - Summe der Anzahl der Requests mit 200 und 304 Antwort
  - Nicht sehr aussagekräftig, weil nicht jede Datei eigenständige Informationseinheit
- Pageviews/Page Impressions
  - Anzahl der abgerufenen HTML-Seiten
  - Anzahl der Hits mit HTML Dateien als Antwort
  - Beschränkt auf einen Medientyp

# Messgrößen

- 4 Hits, 1 Pageview:
  - - - [19/Dec/2002: 12: 05: 51 +0100]  
"GET /~tol k/vml [anguages.html](#) HTTP/1.1" 200 81671  
"http://search.msn.com/resul ts.asp?FORM=sCPN&RS=CHECKED&un=doc &v=1&q=j ava%20wi ndow%20commands"  
"Mozi l l a/4. 0%20(compati bl e; %20MSI E%206. 0; %20Wi ndows%20NT%205. 1)"
  - - - [19/Dec/2002: 12: 05: 51 +0100]  
"GET /~tol k/[uncl ear.gi f](#) HTTP/1.1" 200 988 "[http://fl p. cs. tu-berl i n. de/~tol k/vml anguages.html](http://flp.cs.tu-berl i n.de/~tol k/vml anguages.html) "  
"Mozi l l a/4. 0%20(compati bl e; %20MSI E%206. 0; %20Wi ndows%20NT%205. 1)"
  - - - [19/Dec/2002: 12: 05: 51 +0100]  
"GET /~tol k/[new.gi f](#) HTTP/1.1" 200 907  
"[http://fl p. cs. tu-berl i n. de/~tol k/vml anguages.html](http://flp.cs.tu-berl i n.de/~tol k/vml anguages.html) "  
"Mozi l l a/4. 0%20(compati bl e; %20MSI E%206. 0; %20Wi ndows%20NT%205. 1)"
  - - - [19/Dec/2002: 12: 05: 51 +0100]  
"GET /~tol k/[cool 2.gi f](#) HTTP/1.1" 200 4942  
"[http://fl p. cs. tu-berl i n. de/~tol k/vml anguages.html](http://flp.cs.tu-berl i n.de/~tol k/vml anguages.html) "  
"Mozi l l a/4. 0%20(compati bl e; %20MSI E%206. 0; %20Wi ndows%20NT%205. 1)"
- Pageviews und Framesets
  - Erster Abruf des Framesets ist 1 Pageview
  - Jedes Neuladen eines Inhaltsframes ist 1 Pageview
  - Zum Messen immer nur einen Frame neuladen (DMMV)

# Messgrößen

---

- Visits / Sessions
  - Zusammenhängende Abrufe in einem Zeitraum
  - Navigationspfade aus Logfile
  - Nicht zuverlässig identifizierbar
  - Problem: Wann ist Visit beendet?
- Heuristiken
  - Zeitorientiert:
    - Gesamtdauer einer Visit ist nach oben begrenzt
    - Verweildauer auf einer Seite ist nach oben begrenzt
  - Navigationsorientiert
    - Topologische Begrenzung: Sitzungsende, wenn Seite nicht von vorherigen Seiten aus erreicht werden konnte
    - Begrenzung durch Referrer: Sitzungsende, wenn Seite nicht durch Navigation von vorheriger Seite erreicht wurde

# Messgrößen

- Unique Visitors
  - Abrufe von gleicher IP Adressen als 1 Besucher gezählt
  - Objektiv nicht aussagefähig (Proxies, Dynamische IP Adressen, etc.)
- AdImpressions / Clickthroughs
  - Klick auf Werbebanner
  - Messbar beim Werbekunden
  - Quelle durch Referer ermittelbar
  - Abrechnung
    - Preis nach Attraktivität des Werbeträgers: Pageviews und Visits als Maß
    - Preis nach Effizienz des Werbemittels: Clickthroughs als Maß



# Messgrößen

---

- Viewtime
  - Dauer des Verweilens auf einem Angebot
  - Kaum aus Logfile messbar
  - Klientenseitige Unterstützung notwendig (z.B. Scripting)
  - Sitzt der Nutzer vor dem Rechner?
- Durch zusätzliche direkte Befragung ermittelbar:
  - Qualified visits: Bestätigte Besuche
  - Regionale Herkunft
  - Alter, Geschlecht etc.
  - Interessen
  - Akzeptanz

# Wer misst?

- Serverbetreiber nach eigenem Verfahren und eigener Auswertung
- Serverbetreiber oder Externer nach standardisiertem Verfahren und Auswertung
  - „Informationsgemeinschaft zur Feststellung der Verbreitung von Werbeträgern e.V.“ (IVW) (<http://www.ivwonline.de/>)
  - Messung z.B. durch transparente Grafiken („IVW-Pixel“) auf Seiten
    - ``
    - ``
  - Lösen Messung aus
  - IVW Zahlen sind Grundlage für Preisgestaltung

# Aus [www.spiegel.de/index.html](http://www.spiegel.de/index.html)

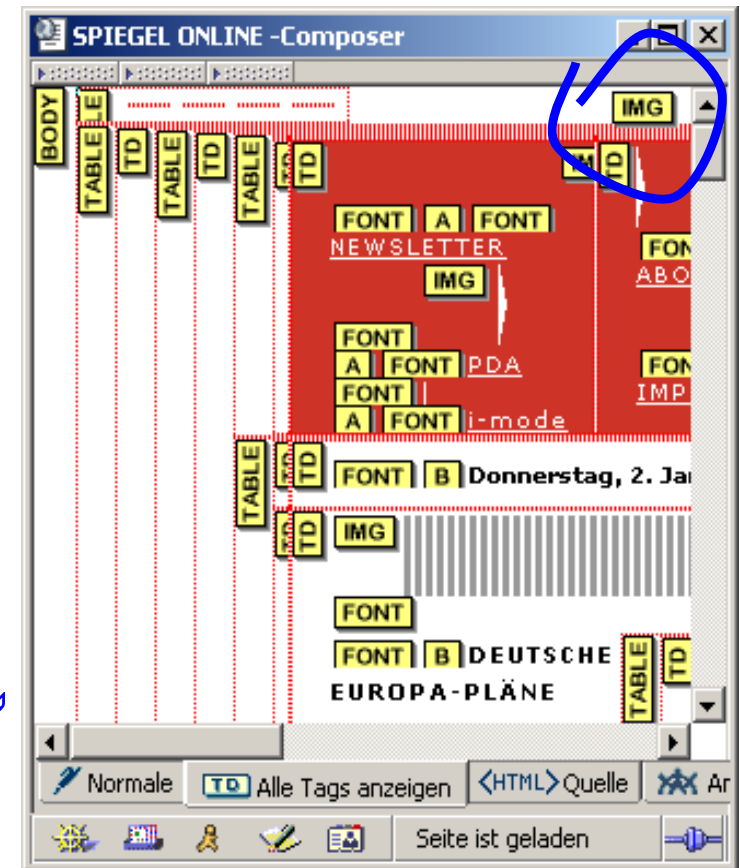
```

<body bgcolor="#ffffff" text="#000000"
  link="#b20a15" vlink="#b20a15" alink="#ff0000"
  marginheight="0" marginwidth="4" leftmargin="4"
  topmargin="0" rightmargin="4" bottommargin="0">
<!-- I W V E R S I O N = " 1 . 2 " -->
<script language="JavaScript">
<!--
  var IWV="http://spiegel.iwbox.de/cgi-bin/iw/CP/
    spiegel;/home/c-18/be-PB64-aG9tZXBhZ2UvY2VudGVy";
  document.write('<IMG SRC="' + IWV + '?r=' +
    escape(document.referrer) + '" WIDTH="1"
    HEIGHT="1" BORDER="0" ALIGN="RIGHT">');
// -->
</script>

<noscript>
  <IMG SRC="http://spiegel.iwbox.de/cgi-bin/iw/CP/spiegel;
    /home/c-18/be-PB64-aG9tZXBhZ2UvY2VudGVy"
    WIDTH="1" HEIGHT="1" BORDER="0" ALIGN="RIGHT">
</noscript>
<!-- /IWV -->
<!-- IWV VERSION="prev" -->

<!-- /IWV -->

```



# IVW Messungen

IVW Online Nutzungsdaten <span>09-2007</span>		Info		Basisdaten		Kategorien					
Gemeldete Angebote:	506	Info	Local - Liste	Visits	PageImpressions	Redaktioneller Content	User generierter ...	E - Commerce	Kommunikation	Suchmaschinen, Verzeichnisse ...	Spiele
Gemeldete PageImpressions:	18.361.197.827										
Gemeldete Visits:	1.847.648.735										
Angebote Netzwerke Vermarktungsgemeinschaften											
Springer-Tagblatt.de				126.262	1.312.733	1.249.362	7.336	12.147	1.107	40.941	4
Spektrum der Wissenschaft/Wissenschaft online				377.983	1.364.186	1.025.841	—	259.485	15.135	7	—
SPIEGEL ONLINE				68.659.980	405.467.226	395.959.105	5.922.523	2.643.487	73.551	—	—
Spieletipps.de				4.574.826	39.568.845	38.561.141	62.843	—	237.346	—	—
Spin.de				7.731.337	308.934.589	117.483	166.982.170	—	136.671.311	—	4.754.858
Sport Auto				109.071	737.631	715.600	20.671	1.330	—	—	—
Sport1				24.544.218	174.416.190	161.198.918	—	—	1.481.473	—	10.853.759
Hinweis: Aus technischen Gründen konnten für diesen Monat keine vollständigen Nutzungsdaten ermittelt werden											
STADTPLANDIENST				1.307.977	16.759.249	12.309	—	18.801	—	16.727.932	—
STAR FM 87.9 MAXIMUM ROCK!				124.213	448.072	441.577	—	—	—	5.226	—
Stellen Online				57.584	248.899	13.253	—	197.455	—	38.167	—
stellenanzeigen.de				995.518	6.112.430	2.018.916	—	3.917.382	176.048	—	—
stern.de				12.759.503	130.459.351	108.625.499	20.269.217	32.818	1.137.166	190.344	—
studieren.de				126.602	1.933.270	1.933.254	—	—	—	—	—
StudiVZ				111.637.977	3.666.027.724	17.293.349	3.648.217.686	429.792	—	—	—
Stuttgarter Zeitung online & Stuttgarter Nachrich ...				1.139.171	8.337.212	3.882.245	2.425.551	1.842.704	10.523	46.172	40.035
Stylepark				82.520	818.153	29.782	—	—	—	788.370	—

- Datenaufbereitung
  - Extraktion relevanter Zugriffe, also z.B. nicht Hits auf eingebettete Daten etc. (jpg, map, robots.txt)
  - Hinzufügen verloren gegangener Zugriffe
    - Zusammenführen mit Cookie-Informationen
    - Zusammenführen mit Registrierungsinformationen
    - Heuristiken zur Cache Nutzung
- Sitzungsermittlung
  - Ziel: Sequenzen von zusammengehörigen (gleicher Nutzer, gleiche Nutzung) Zugriffen als Sitzung (Session, Visit) identifizieren
  - Problem vergleichbar mit dem Problem der Identifizierung eines Nutzers

- Path analysis
  - Ermittlung von Pfaden in Graphen (oder Graphen aus Pfaden), die Web-Site repräsentieren
    - Link-Struktur einer Site
    - Ähnlichkeitsstruktur von Seiten einer Site
    - Linkverfolgungsstruktur einer Site
  - Weitere Zusammenhänge ermitteln:
    - 70% der Nutzer, die /inst/ag-nbi/lehre/03/S\_SW/ zugegriffen haben kamen über den Pfad /inst, /inst/ag-nbi (20% über /lehre/, ...)
    - 5% der Nutzer haben ihren Besuch bei /inst/ag-nbi begonnen
    - 70% der Nutzer haben ihre Sitzung nach einem Pfad der Länge 5 beendet
  - Nutzung dieser Zusammenhänge für die Struktur der Site

- Association rule
  - Ermittlung von Korrelationen zwischen Zugriffen einer Sitzung
    - 30% der Nutzer die /inst/ag-nbi besucht haben, haben auch /inst/ag-tech besucht
    - 2% der Nutzer von /inst/ag-nbi/lehre/0203/V\_NBI/ haben sich danach in die Mailingliste eingetragen auf [http://lists.spline.inf.fu-berlin.de/mailman/listinfo/nbi\\_v\\_nbi](http://lists.spline.inf.fu-berlin.de/mailman/listinfo/nbi_v_nbi)
- Sequential pattern
  - Ermittlung von Zusammenhängen zwischen Sitzungen
    - 20% der Nutzer, die sich über [http://lists.spline.inf.fu-berlin.de/mailman/listinfo/nbi\\_v\\_nbi](http://lists.spline.inf.fu-berlin.de/mailman/listinfo/nbi_v_nbi) in eintragen, haben das innerhalb von 10 Tagen über [http://lists.spline.inf.fu-berlin.de/mailman/listinfo/nbi\\_s\\_xml](http://lists.spline.inf.fu-berlin.de/mailman/listinfo/nbi_s_xml) auch für die andere Mailingliste eingetragen

# Analysetechniken

- Classification rules
  - Ermittlung von Profilen von Nutzergruppen
    - 80% derjenigen, die sich unter [http://lists.spline.inf.fu-berlin.de/mailman/listinfo/nbi\\_v\\_nbi](http://lists.spline.inf.fu-berlin.de/mailman/listinfo/nbi_v_nbi) eingetragen haben, studieren Diplom-Informatik
    - Bachelor-Studierende besuchen eher Seiten unter /inst/ag-nbi als unter /inst/ag-bio
- Clustering
  - Gruppierung ähnlicher Nutzer und Daten
    - Interesse an hochpreisiger Consumer-Electronic
  - Nutzung für Marketing und Site-Personalisierung (z.B. amazon)



# Zusammenfassung

---

- Ermittlung von Informationen über Nutzer und Nutzung notwendig
- Logfiles als Datenbasis bei Servern, verschiedene Format
- Verschiedene Messgrößen verbreitet
- Ermittlung teilweise sehr schwer
- Web Usage Mining zur Ermittlung komplexerer Zusammenhänge

# Literatur

---

- *Common Logfile Format*.  
<http://www.w3.org/Daemon/User/Config/Logging.html#common-logfile-format>
- Mike StJohns. *Identification Protocol*. Request for Comments 1413. February 1993  
<http://www.ietf.org/rfc/rfc1413.txt?number=1413>
- DMMV. *Messgrößen*.  
[http://www.dmmv.de/de/7\\_pub/homepagedmmv/themen/emarketing/media/zielemedia.cfm](http://www.dmmv.de/de/7_pub/homepagedmmv/themen/emarketing/media/zielemedia.cfm)
- Bettina Berendt, Bamshad Mobasher, Myra Spiliopoulou, and Jim Wiltshire. *Measuring the accuracy of sessionizers for web usage analysis*. In Workshop on Web Mining at the First SIAM International Conference on Data Mining, pages 7-14, April 2001.  
<http://maya.cs.depaul.edu/~mobasher/papers/wm-siam01.pdf>
- R. Cooley, B. Mobasher, J Srivastava. *Web Mining: Information and Pattern Discovery on the World Wide Web*. Proceedings of the 9th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'97), November 1997.  
<http://maya.cs.depaul.edu/~mobasher/papers/webminer-tai97.ps>



## Caching im Web



# Überblick

---

- Caches
- Caching Architekturen
- Cache Füllung
- Cache Ersetzung
- Cache Kohärenz

- Zwei Messungen 10.1.05 beim ISP sprint in USA  
[<http://ipmon.sprint.com/packstat/packet.php?050110>]

**Application Breakdown**

Category	Packets (%)	Bytes (%)	Flows (%)
Web	51.20	58.08	34.70
File Sharing	5.89	6.19	7.50
FTP	0.65	1.62	0.41
Email	4.21	4.03	2.93
Streaming	1.82	0.69	4.03
DNS	1.36	0.55	4.96
Games	0.00	0.00	0.00
Other TCP	16.24	10.46	19.99
Other UDP	7.02	7.29	14.06
Not TCP/UDP	11.60	11.09	11.42

**Application Breakdown**

Category	Packets (%)	Bytes (%)	Flows (%)
Web	42.19	60.86	26.48
File Sharing	8.02	6.17	11.76
FTP	0.65	0.35	0.52
Email	3.46	4.09	1.92
Streaming	5.76	3.56	5.19
DNS	0.96	0.27	3.55
Games	0.00	0.01	0.00
Other TCP	21.51	11.66	34.90
Other UDP	6.44	4.97	6.79
Not TCP/UDP	11.00	8.06	8.89

# Caching

---

- Ursprünglich aus Rechnerarchitektur:
  - CPU schneller als Hauptspeicher
  - → Daten in schnellem Zwischenspeicher, dem *Cache* halten
- Ziel im Web: Netzwerklatenz kaschieren
  - Klient schneller als Netz (+ Server)
- Grundbegriffe:
  - Gesuchte Daten zwischengespeichert vorgefunden: *Hit / Treffer*
  - Gesuchte Daten nicht gefunden: *Miss / Fault / Fehler*
  - Bei Fehler nachgeladenes Originaldatum in Cache gespeichert
  - → Annahme: Mehrere Zugriffe zeitlich gruppiert
  - Oft: Block um gesuchte Daten in Cache geholt
  - → Annahme: Zugriffe örtlich gruppiert

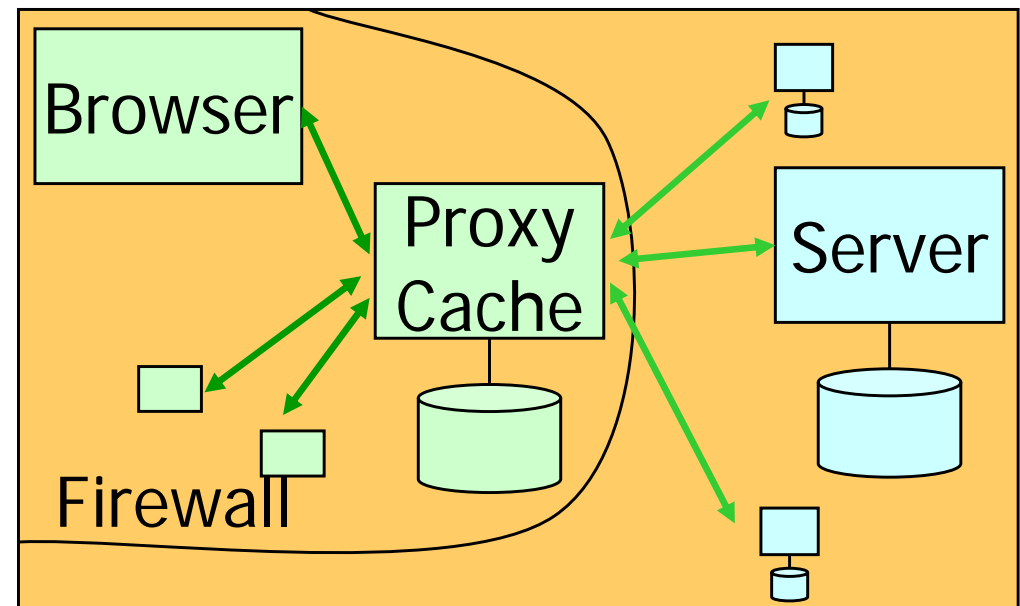
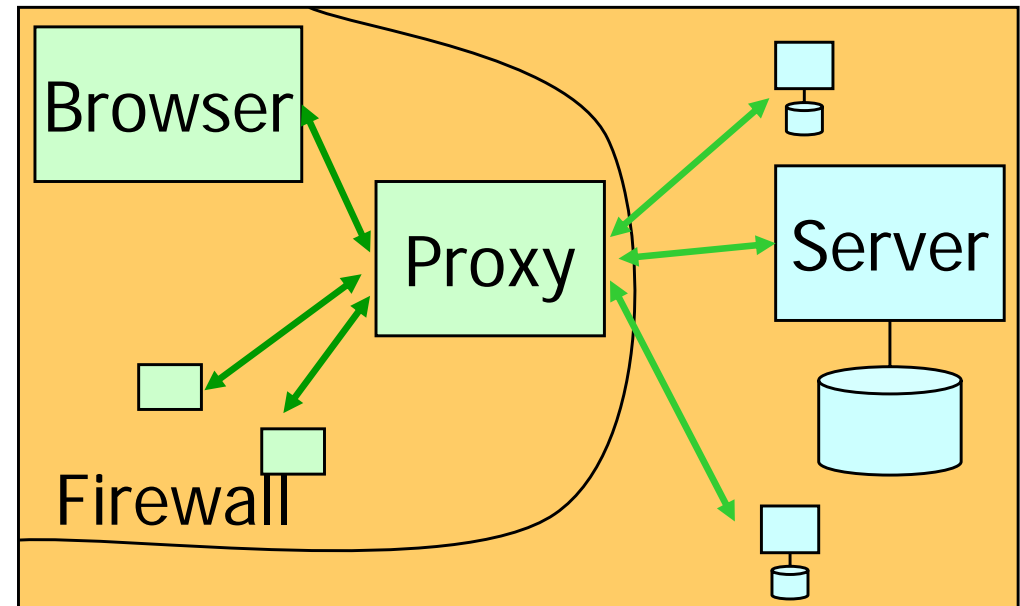
# Gütemaße

---

- Hit Rate:  
Anteil der aus dem Cache geholten Objekte an Gesamtzahl der angefragten Objekte
- Weighted Hit Rate:  
Anteil der Gesamtgröße in Bytes der aus dem Cache geholten Objekte an Gesamtgröße der angefragten Objekte
- Time:  
Mittlere Antwortzeit beim Nutzer

# Proxies und Caching

- *Proxy/Stellvertreter* anstelle vom Klienten
  - Leitet HTTP Anfrage von Klienten an Server / andere Proxies weiter
  - Tritt für den Server als Klient auf
  - Leitet Antwort an Klienten weiter
- *Proxy Cache*
  - Agiert auch als Cache
  - ➔ Annahme: Zugriffe organisatorisch gruppiert





# Vorteile

- Vorteile von (Proxy-)Caching
  - Netzlast kann effektiv gesenkt werden
  - Netzlatenz für Nutzer sinkt
    - Übertragungszeit sinkt, wenn Objekte im (netztopologisch) nahen Cache gefunden werden
    - Nicht gefundene Objekte werden schneller geholt wegen
      - geringere Netzlast auf dem Weg zum Server
      - geringere Last beim Server [<http://www.heise.de/newsticker/meldung/80812>]
    - "~~eight~~ second rule": Web Seite muss innerhalb von ~~8~~ 4 Sekunden angezeigt werden oder Nutzer verlieren Interesse an Site (bzw. Kauf...)
  - Serverlast sinkt wegen geringerer Zugriffszahl
  - Verfügbarkeit von Objekten steigt
  - Proxies lassen Nutzungsanalysen zu

- Nachteile von (Proxy-)Caching
  - Cache-Inhalt muss nicht konsistent mit Originaldaten sein
  - Bei einem Cache-Fehler steigt die Netzlatenz  
(→ Hit-Rate maximieren, Miss-Kosten minimieren)
  - Proxy wird zum Engpass für Klienten
  - Proxy wird zum Single-point-of-failure
  - Proxy-Cache senkt Hit-Raten (und anderen Maße) beim Server  
(→ Server können versuchen, Caching zu verhindern)
  - Unnötig bei Seiten die nur einmalig geladen werden

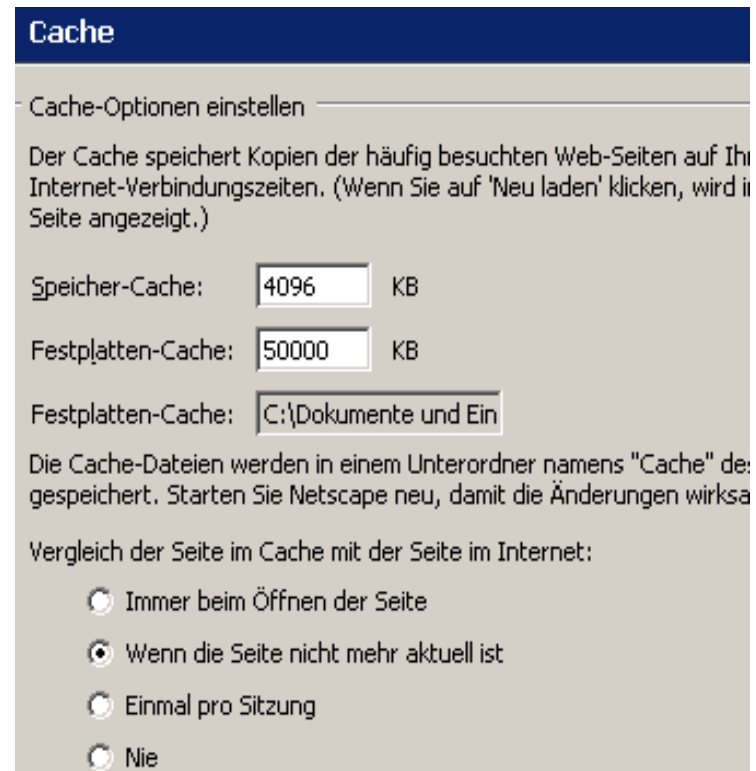
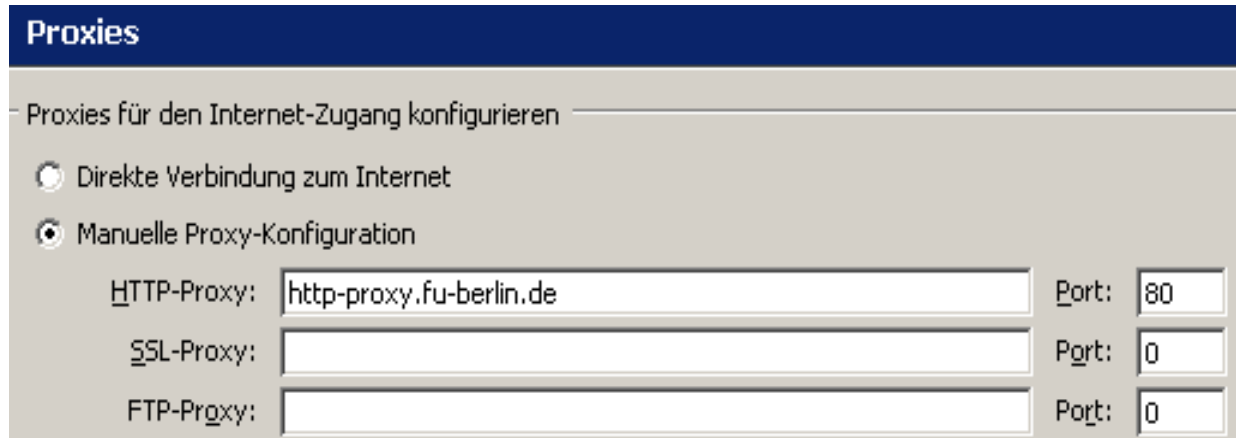
# Notwendigkeit

---

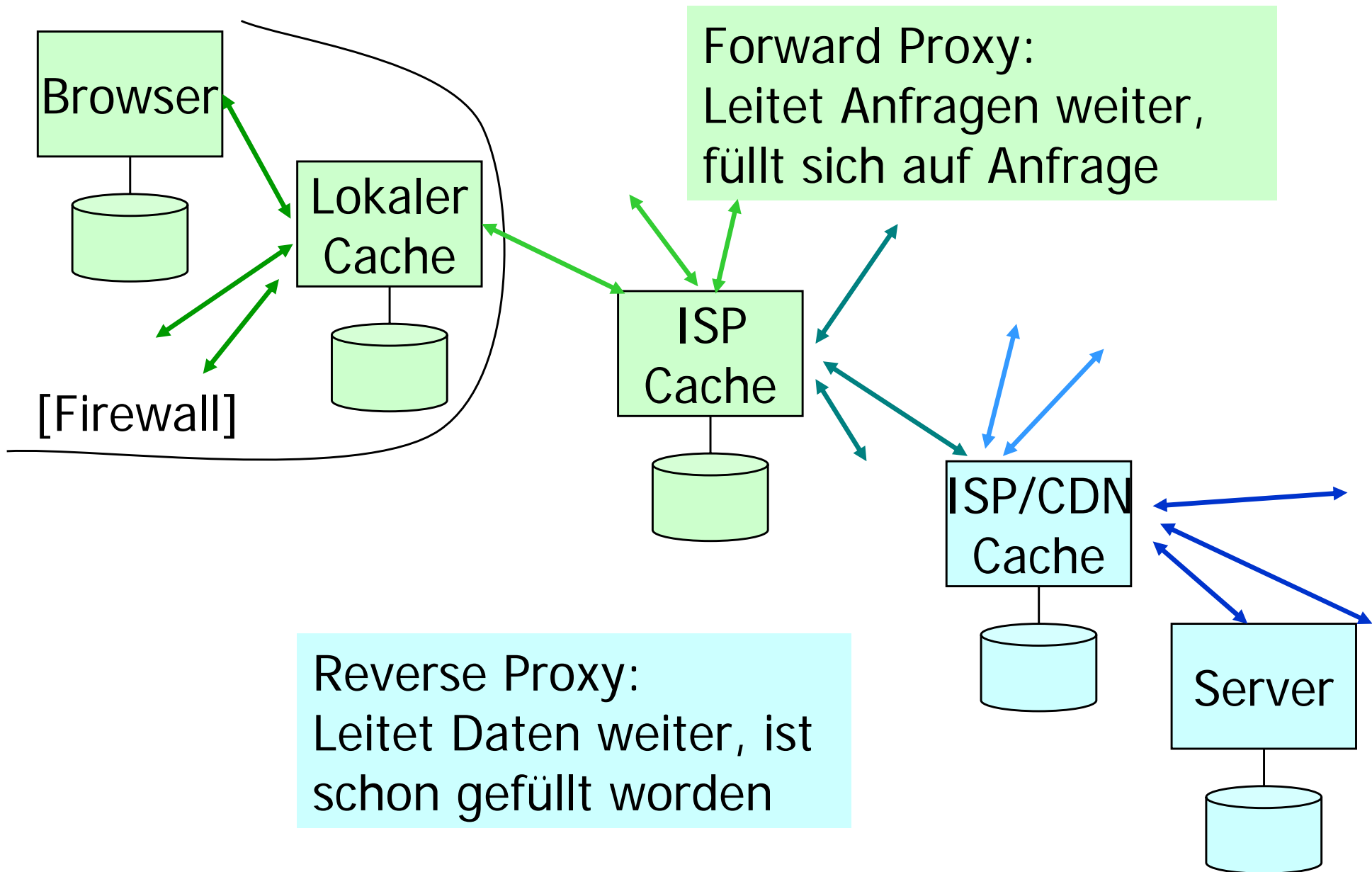
- Netznutzung verursacht immer Kosten
- Netzlatenzen werden immer schwanken
- Entfernungen im Netz werden durch mehr Geräte größer
- Bandbreite der Inhalte wird immer steigen
- Populäre Server sind immer überlastet
- Netzkosten sind größer als Rechenkosten

# Browser Konfiguration

- Browser kann zu externem Proxy-Cache gerichtet werden
- Browser hat aber auch selber schon einen lokalen Cache (Speicher / Disk)



# Caches im Web allgemein



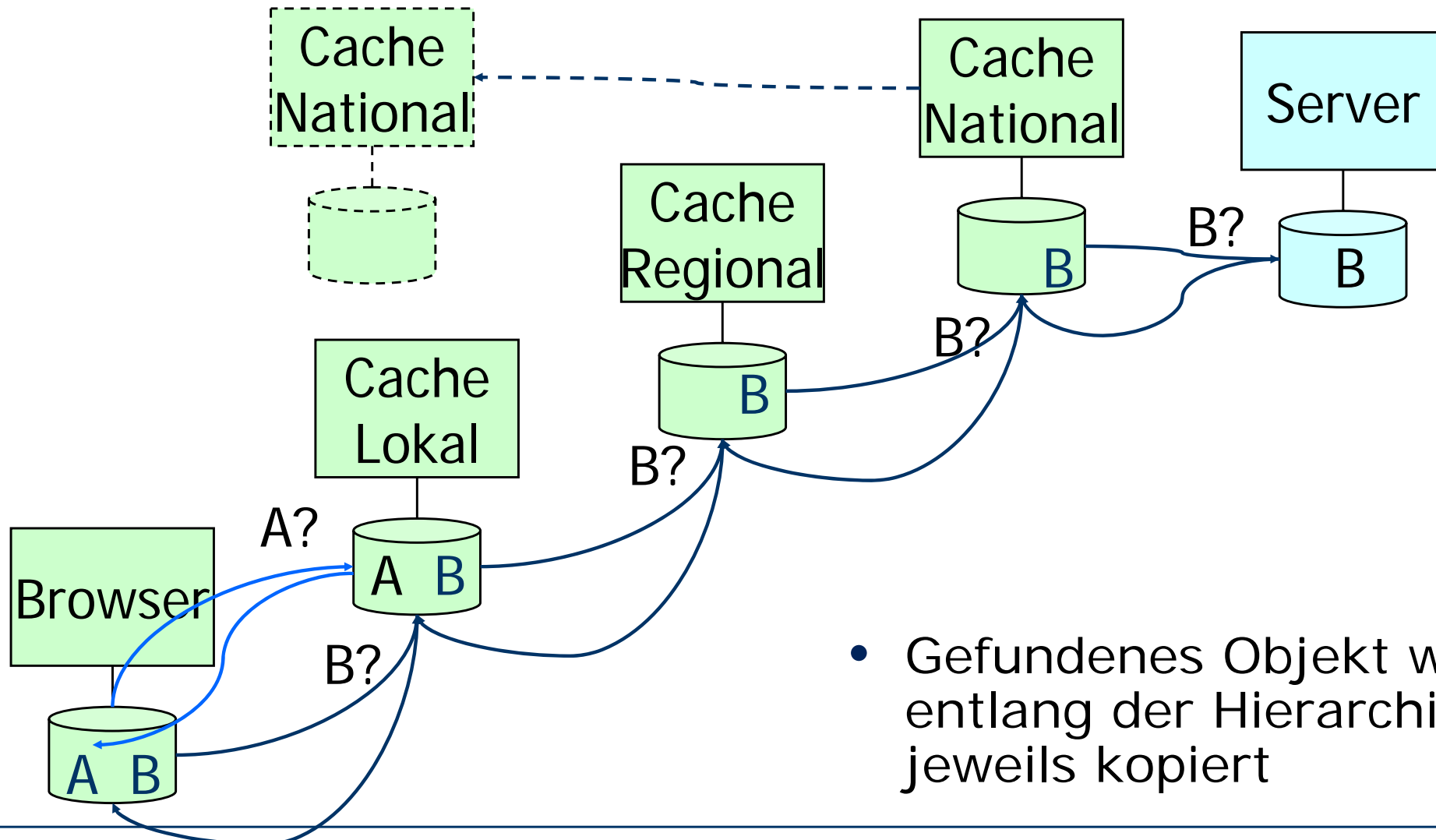
- Forward Proxy Caches
  - Stellvertreter für Klienten
  - Vermeidung der Netznutzung
  - Browser-Cache: Lokal, nutzerspezifisch
  - Proxy-Cache: Organisationsweit, gruppenspezifisch
  - ISP-Cache: Teilnetzweit
- Reverse Proxy Caches („HTTP Accelerators“)
  - Stellvertreter für Server
  - Vermeidung der Servernutzung
  - Content Delivery Network-Cache: Anbieterweit, sitespezifisch
  - Server Cache: Serverweit, sitespezifisch

# Design-Ziele

---

- *Senkung der Netzlatenz* für Nutzer
- *Robustheit* gegenüber Fehlern
  - Toleranz zu einzelnen Ausfällen
  - Kontrollierter Leistungsrückgang
  - Einfaches Wiederherstellung nach Ausfällen
- *Transparenz* des Cachings für den Nutzer
- *Skalierbarkeit* mit Web-Wachstum
- *Effizienz* der Ressourcennutzung
  - Minimale zusätzliche Netzlast
  - Erhaltung optimaler Ausnutzung von Ressourcen
- *Adaptivität* zum Nutzerverhalten und Netzzustand
- *Stabilitätserhaltung* der Gesamtnetzes
- *Lastverteilung* entlang den beteiligten Komponenten
- *Einfachheit* als Voraussetzung weiter Verbreitung

- Hierarchisches Caching: Anfrage wird bei einem Cache-Miss über mehrere Hierarchiestufen weitergereicht

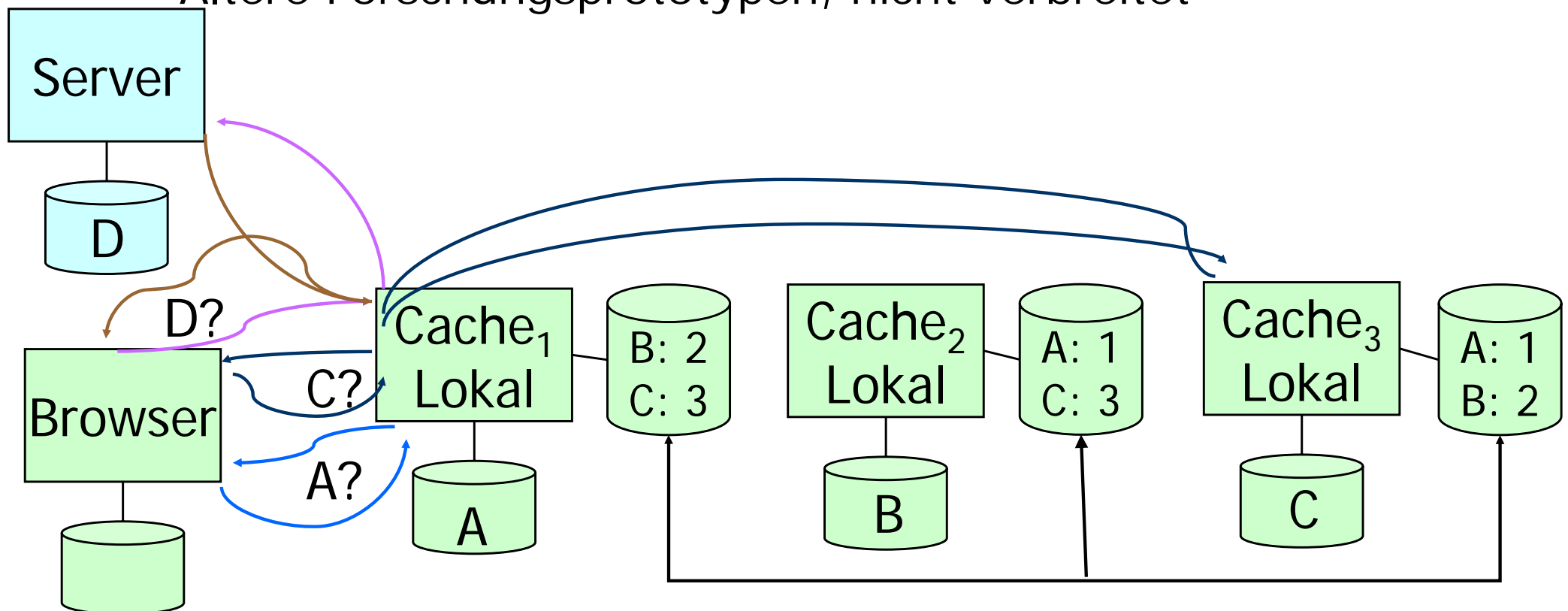


- Gefundenes Objekt wird entlang der Hierarchie jeweils kopiert



- Vorteile
  - In der Hierarchie niedrig stehende Caches profitieren von besserer Bandbreite der hoch stehende Caches
  - Daten werden in Richtung der Nachfrage repliziert
  - Senkt Latenz für kleine Dokumente relativ mehr
- Nachteile
  - Platzierung der Caches an zentralen Netzknoten ist kritisch
  - Je Cache-Level erneut zusätzliche Latenz möglich
  - Je höher ein Cache in der Hierarchie steht umso eher wird er zum Engpass
  - Daten werden an sehr vielen Stellen entlang der Cache-Hierarchie in Kopien gehalten

- Caches kooperieren
  - wissen, welcher Cache welche Objekt hält (Tabelle, Hashing,...)
  - fragen nach (→ Harvest ICP)
  - Ältere Forschungsprototypen, nicht verbreitet



# Cache Füllung

---

- Caches kommen auf Hit-Raten von 40-50%  
→ Wahrscheinliche Zugriffe vorwegnehmen
- Cache wird gefüllt durch
  - Kopieren nachgefragter Objekte
  - Aktives holen benachbarter Objekte
    - „Nachbarschaft“ ergibt sich aus Einbettung von Objekten auf Web-Seiten und Links
    - Aktives Füllen eines Web-Caches: *Prefetching*
- Weitere Beobachtung:  
Die meisten Objekte im Cache werden nur einmalig genutzt

# Prefetching

---

- Auswirkung
  - zwischen Browser und Server
    - 45% weniger Latenz beim Browser
    - 200% Netzlast
  - zwischen Proxy und Server
    - 60% weniger Latenz beim Browser
  - zwischen Klient und Proxy
    - 23% Weniger Latenz
    - Großer Browser-Cache notwendig
- Push-Ansätze
  - Server/Proxy verteilt Dokumente zu Klienten
  - nicht im HTTP Modell
- Prefetching nicht verbreitet / nicht als sinnvoll angesehen

- Wenn Cache voll ist, muss Platz durch Löschung von Objekten geschaffen werden
- Drei Ansätze
  - Traditionelle Ansätze, basiert auf Objektnutzung
    - Least Recent Used LRU:  
Objekt mit ältester Nachfrage wird gelöscht
      - Populärste Strategie
      - Großes neues Objekt verdrängt viele kleine alte
    - Least Frequently Used LFU:  
Objekt mit wenigsten Nachfragen wird gelöscht
      - Größe von Objekten nicht berücksichtigt
      - Da die meisten Objekte gleich oft genutzt werden (1) ist Auswahl eigentlich zufällig
    - Pitkow/Recker:  
Wie LRU, aber: falls alle Objekte am selben Tag nachgefragt wurden, wird größtes gelöscht

- Basiert auf Ordnung von Objekteigenschaften
  - Size:  
Jeweils Größtes Objekt wird gelöscht bis Platz ist
  - LRU-MIN:  
Wendet LRU auf Objekte größer als  $s$  an, falls nur kleinere Objekte, LRU auf Objekte größer als  $s/2$  anwenden usw.
  - LRU-Threshold:  
Wie LRU, aber Objekte über einer Größe  $s$  werden nicht zwischengespeichert
  - Hyper-G:  
Wie LFU, aber gleicher Rang über LRU und Size aufgelöst
  - Lowest-Latency-First (Latency Access Time, LAT):  
Objekt mit geringster Ladedauer wird gelöscht
  - Hybrid:  
Gewichtetes Maß aus Zugriffszeit, Zugriffsfrequenz und Größe

- Simulation mit Web-Traces der Univ. Berkeley
  - 1.11.96-18.11.96, 10000 Personen, 2500000 Zugriffe
- Algorithmus „Perfect“: Cache mit unbegrenzter Größe

Algorithms	Hr	Whr	Time
Perfect	30.2	28.7	8.97
LRU	24.6	21.4	10.54
LFU	25.3	21.8	10.62
SIZE	26.0	18.4	9.43
LAT	24.7	20.4	9.51
LRU-MIN	25.9	20.2	10.20
Hybrid	25.4	22.0	9.31

- SIZE erhöht die Anzahl der Objekte im Cache
- LRU/LFU profitieren von zeitlicher Lokalität der Zugriffe
- LAT profitiert von hohen Latenzen der gehaltenen Objekte
- LRU-MIN und Hybrid haben erhöhten Aufwand

- Basiert auf Nutzungskosten
  - Greedy Dual-Size:  
Kosten werden mit Objekt verbunden, Objekt mit geringstem Kosten/Größe Verhältnis wird gelöscht
  - Lowest Relative Value:  
Nutzen wird mit Objekt verbunden, Objekt mit geringstem Nutzen wird gelöscht
  - Least Normalized Cost Replacement:  
Funktion  $\text{Zugriffshäufigkeit} \times \text{Übertragungskosten} \times \text{Größe}$
  - ...
- Performanz der Ersetzung ist stark von Nutzungscharakteristik abhängig
- Kein Verfahren ist für alle Nutzungscharakteristika überlegen



# Cache Kohärenz

- Nutzer können veraltete Seiten vom einem Cache erhalten
- Ähnlich Cache-Kohärenz in Verteilten Systemen, aber
  - andere Zugriffsmuster
  - andere Dimensionen
  - Web-Objekte werden nur an einem Ort geändert
- HTTP Unterstützung
  - Header Expires: *Datum* liefert Ungültigkeitsdatum
  - GET mit If-Modified-Since: *Datum* Header liefert Seite nur bei Änderungen nach einem Datum
  - Header Pragma: no-cache verhindert Caching
  - Header Last-Modified: *Datum* liefert Änderungsdatum
  - Header Date: *Datum* enthält Datum des letzten Tests auf Aktualität
  - Header ETag: *Signatur* liefert eine Quersumme des Objekts

# Cache Kohärenz Mechanismen

---

- Starke Cache-Konsistenz: Immer aktuelle Objekte halten
  - Klient validiert
    - Annahme: Ressourcen im Cache sind veraltet
    - Vorgehen: Bei jeder Nutzung validieren
    - Implementierung: GET mit If-Modified-Since: Header
      - 200 – Keine Änderung
      - 304 – Not modified Antwort bei keiner Änderung (RFC: „should“)
  - Server invalidiert
    - Annahme: Ressourcen im Cache sind aktuell
    - Vorgehen: Server sendet Mitteilung bei Änderung
    - Implementierung: Listen über Cache-Klienten führen
      - Wie skalieren?
      - Wie Listen aktuell halten?

# Cache Kohärenz Mechanismen

- Schwache Cache-Konsistenz: Irgendwann aktuelle Objekte
  - Klient invalidiert: Adaptive Time-To-Live (TTL)
    - Ausgangspunkt: Lebensdauer von Objekten ist bimodal
      - Entweder sehr kurze Lebensdauer
      - oder sehr lange Lebensdauer
    - Vorgehen: TTL eines Objekts = Anteil seines Alters (Aktuelle Zeit – Last-Modified)
    - Implementierung: Harvest: Anteil = 50% (CERN httpd 10%)
    - Vorteil: Hält Anteil alter Dokumente unter 5%
    - Nachteile
      - Nur Heuristik: Nutzer muss eventuell unnötig warten
      - Nur Heuristik: Keine Aussage über tatsächliche Gültigkeit
      - Nutzer können Heuristik nicht beeinflussen
      - Was passiert bei abgebrochenen Ladevorgängen?

- Piggyback Invalidation (Piggyback = „Huckepack“)
  - Ausgangspunkt: Kommunikation mit Server nutzen um Gültigkeit zu erfragen
  - Vorgehen:
    - Piggyback Cache Validation (PCV):  
Mit einer Anfrage schickt Proxy ein Liste zu validierender Objekte
    - Piggyback Server Invalidation (PSI):  
Mit einer Antwort schickt Server eine Liste geänderter Objekte
    - Hybrid: PCV+PSI
      - Wenn letzter Kontakt lange her: PCV (Overhead bei langer PSI-Liste größer)
      - Wenn letzter Kontakt kurz her: PSI (Liste kurz)

# Cache-Fähigkeit von Objekten

---

- Statischer Inhalt von Seiten sehr gut zwischenspeicherbar
  - „sehr statische“ Inhalte (Logos) mit sehr spätem Expires Header
- Dynamische Inhalte schlecht zwischenspeicherbar
  - „sehr dynamisch“ Inhalte (Börsendaten) nicht cachen
  - „wenig dynamische“ Inhalte (Nutzeranschrift) kurz cachen
  - Serverseitiger Cache als Alternative

# Zusammenfassung

---

- Caches zur Latenzverkürzung beim Nutzer
- Mehrstufiges Caching im Web
- Cache Füllung durch Prefetching
- Cache Ersetzung nach verschiedenen Methoden
- Cache Kohärenz mit verschiedenen Methoden

- Jia Wang. A Survey of Web Caching Schemes for the Internet. *ACM Computer Communication Review*, 25(9), pp. 36-46, October 1999. <http://www.cs.cornell.edu/Info/People/jiawang/web-survey.ps>
- Zona Research, Inc. The Economic Impacts of Unacceptable Web-Site Download Speeds. April 1999  
[http://www.keynote.com/downloads/whitepapers/economic\\_impact\\_of\\_downloadspeed.pdf](http://www.keynote.com/downloads/whitepapers/economic_impact_of_downloadspeed.pdf)
- Brian D. Davison. A Web Caching Primer. *IEEE Internet Computing*, 5(4), pp. 38-45, July/August 2001. <http://www.cs.rutgers.edu/~davison/pubs/2001/internetcomputing/pubprimer.ps.gz>
- Michael Baentsch, Lothar Baum, Georg Molter, Steffen Rothkugel and Peter Sturm. Enhancing the Web's Infrastructure: From Caching to Replication. *IEEE Internet Computing*, 1(2), pp. 18-27, 1997. <http://citeseer.nj.nec.com/baentsch97enhancing.html>
- K. Chinen and S. Yamaguchi. An interactive prefetching proxy server for improvement of WWW latency. *Proceedings of INET'97*, June 1997. Siehe <http://shika.aist-nara.ac.jp/products/wcol/wcol.html>
- R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, T. Berners-Lee. Hypertext Transfer Protocol - HTTP/1.1. RFC 2616. June 1999. <http://www.ietf.org/rfc/rfc2616.txt>
- I. Cooper, J. Dilley Akamai. Known HTTP Proxy/Caching. RFC 3143. June 2001.  
<http://www.ietf.org/rfc/rfc3143.txt>
- Christoph Lindemann, Oliver P. Waldhorst. Analysis of Web Caching in the Gigabit Research Network G-WiN. Abschlußbericht zum Projekt Analyse der Wirksamkeit von Web Caching im G-WiN. University of Dortmund. April 2001. <http://webdoc.sub.gwdg.de/ebook/ah/dfn/Cache-Analysis.pdf>
- Jean-Marc Menaud, Valérie Issarny, Michel Banatre. Improving Effectiveness of Web Caching. In *Recent Advances in Distributed Systems*. S. Krakowiak and S. Shrivastava editors. Springer Verlag, LNCS 1752. 2000. <http://www-rocq.inria.fr/arles/doc/ps00/Caching.pdf>