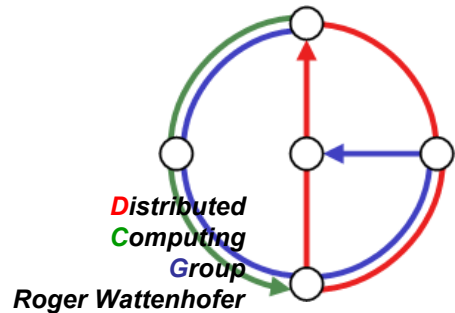


Von Client/Server zu Peer-to-Peer

Ein Paradigmenwechsel

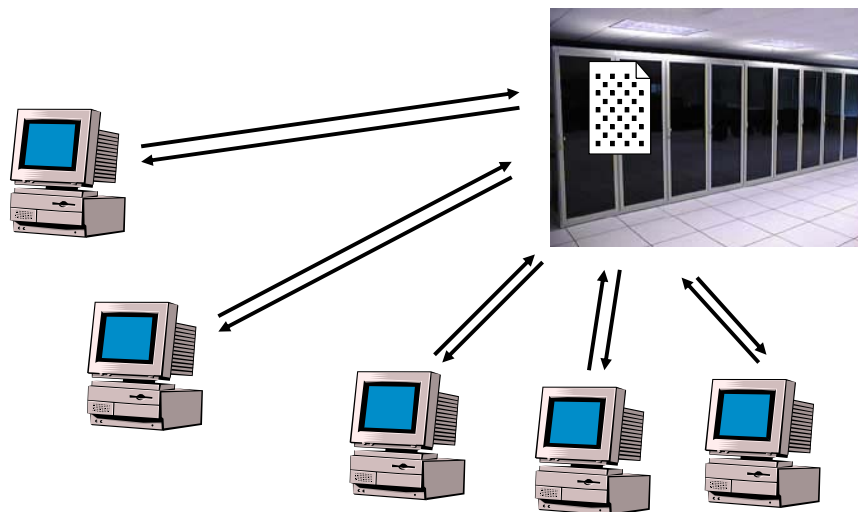


„Paradigmenwechsel“

Eitle Wissenschaftler umschreiben eine Situation, in der ihnen ein Thema oder eine Methode **allmählich langweilig geworden** ist, weswegen sie sich nun anderem zugewandt haben, das **spannender zu sein scheint**, gern als *Paradigmenwechsel*.

Nicht selten kommt einem solchen Wechsel dieselbe Beliebigkeit zu wie einem „**Tapetenwechsel**“. [www.wissen.de]

Client/Server

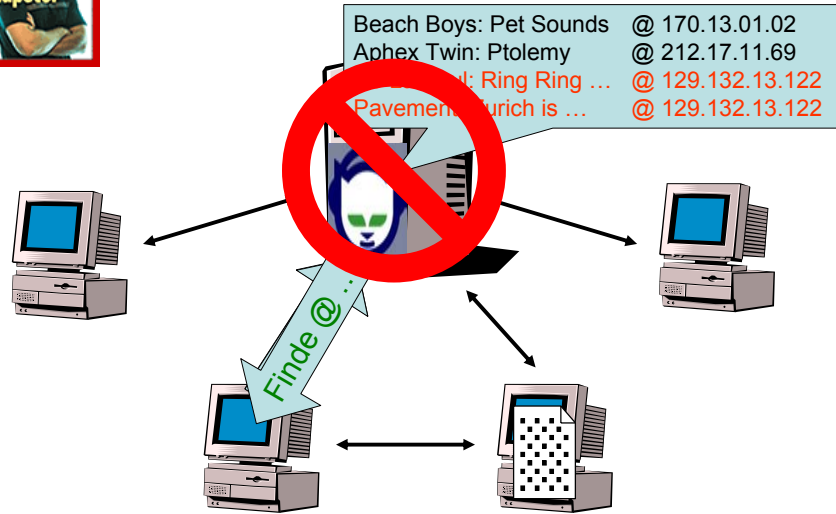


Probleme Client/Server

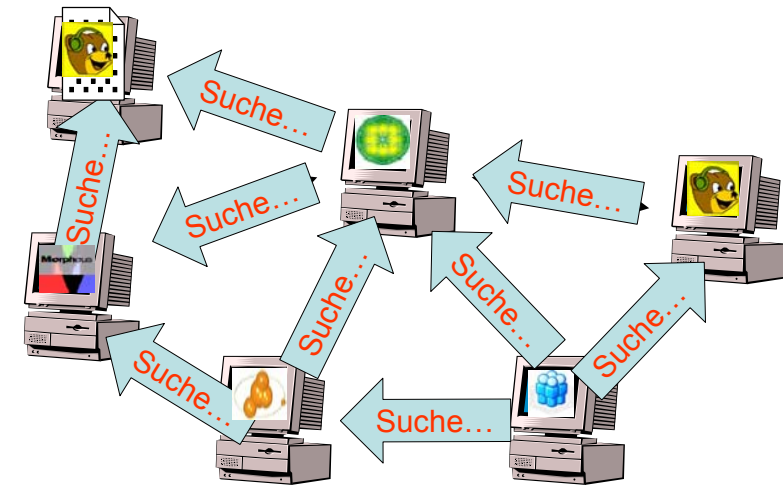
- **Skalierbarkeit**
 - Kann Server 100, 1'000, 10'000 Clients fertig werden? Was ist mit den Kosten?
- **Sicherheitsaspekte / Denial-of-Service**
 - Hacker können Server in die Knie zwingen
- **Replikation**
 - Daten zur Sicherheit replizieren
 - Daten nahe bei Clients replizieren



Fallstudie: Napster



Fallstudie: Gnutella



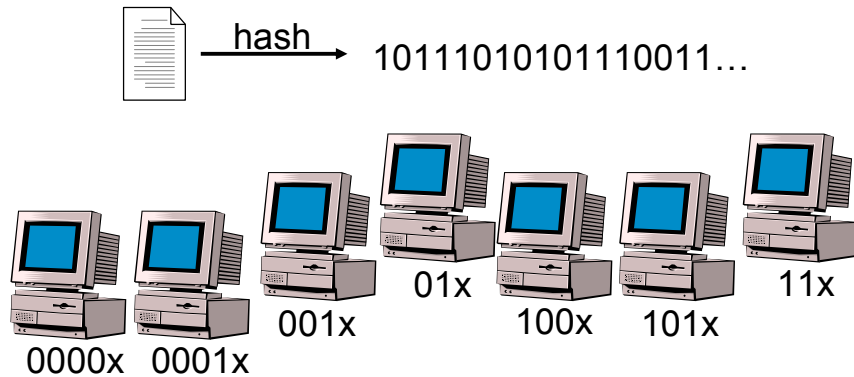
Vor- und Nachteile Gnutella

- vollkommen **dezentral**
- vollkommen **ineffizient**
 - “Fluten” = zielloses Suchen
- Gnutella findet gesuchtes Objekt oft nicht
 - TTL in Suchnachricht
 - Gnutella **“nicht korrekt”**

Was bedeutet „Peer-to-Peer“?

- **Dezentrales („symmetrisches“)** System
 - skalierbar und fehlertolerant
- Auch bekannt als
 - Distributed Computing
 - Overlay Networking
 - Fault Tolerance?
 - Middleware?
 - Grid Computing?
 - Ad-Hoc Networking?!?

Ein besseres Gnutella...

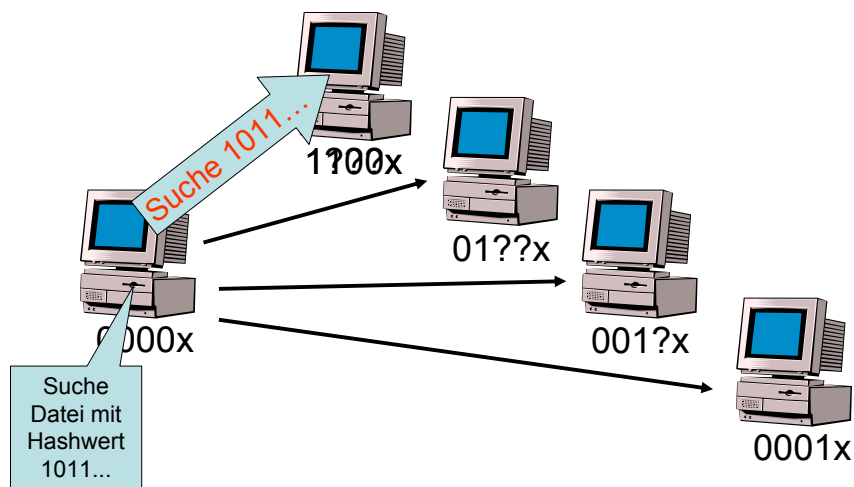


- Statt Dokument direkt zu speichern, kann jeder Peer Anfragen auch weiterleiten

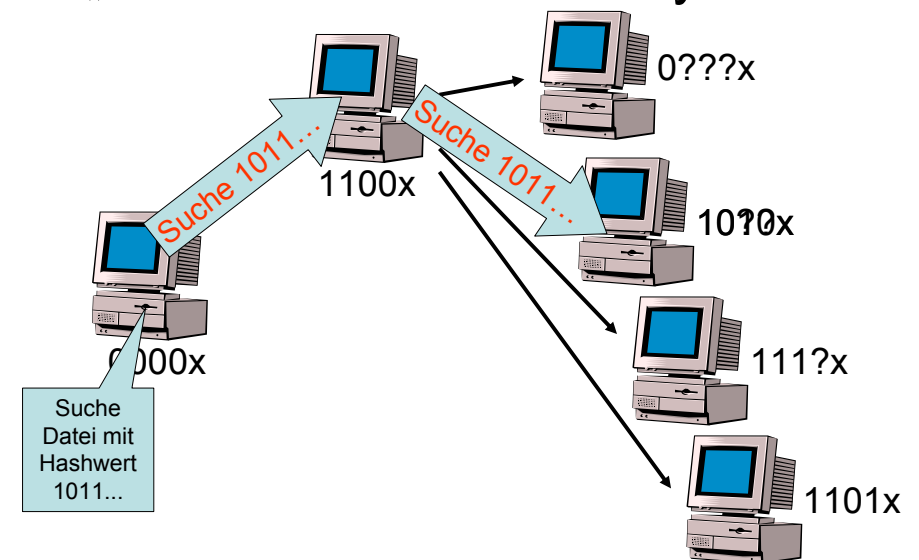
P2P-Systeme sind hochdynamisch

- Peers “kommen und gehen” (join & leave)
 - Es ist absolut unmöglich, dass jeder Peer immer weiss, welcher andere Peer für welchen Hash-Bereich zuständig ist
- Idee: Jeder Peer kennt nur einen kleinen Teil aller Peers im System

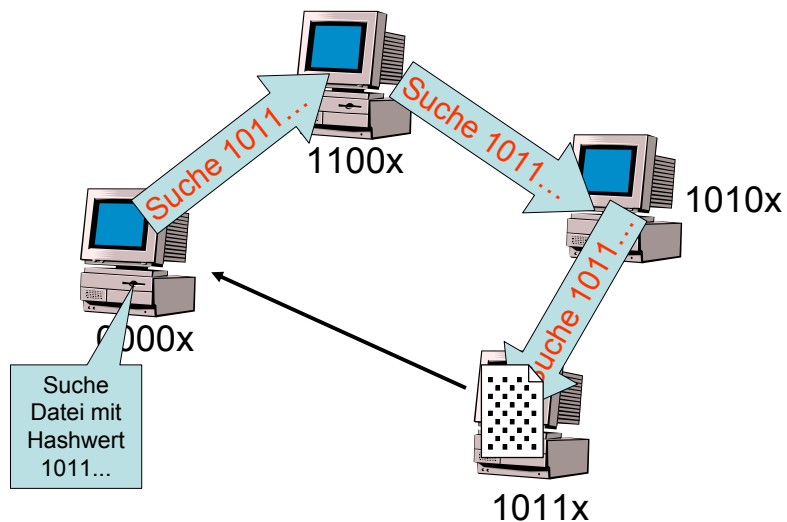
„Gutes“ Peer-to-Peer-System



„Gutes“ Peer-to-Peer-System



„Gutes“ Peer-to-Peer-System



Diskussion

- Suche in $O(\log n)$ Schritten (bei n Peers)
- Neuer Eintritt oder Wegfall von Peer betrifft nur $O(\log n)$ Peers; Änderung mittels lokaler Suche
- Verbesserungen: zum Beispiel Caching
- Verschiedene „gleiche“ Vorschläge
 - Plaxton/Rajaraman/Richa, Chord/CFS, Pastry/Tapestry, Viceroy, ..., Freenet

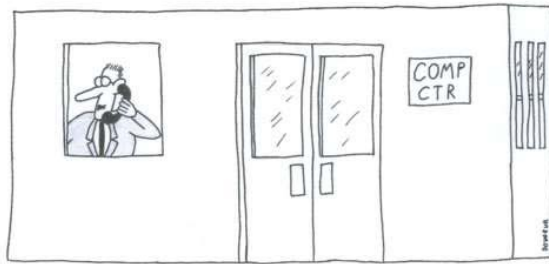
Übersicht

- Motivation
 - Client/Server, Napster, Gnutella
 - „Gutes“ Peer-to-Peer-System
- **Peer-to-Peer Projekte**
 - Farsite, PeerMan, ...
 - Koordinationsstrukturen
 - Echtzeitservice, Verteilter Zähler
 - Praktische Schwierigkeiten
- Ausblick

Farsite Microsoft Research

- Vorstudie
 - Bei vielen Rechnern wird oft nur ein Teil der Festplattenkapazität genutzt
 - Viele Benutzer speichern die gleichen Dateien
- Idee: Nutze die restliche Kapazität, um Sicherheitskopien der Dateien zu erstellen. (Nur innerhalb Firma/Campus)

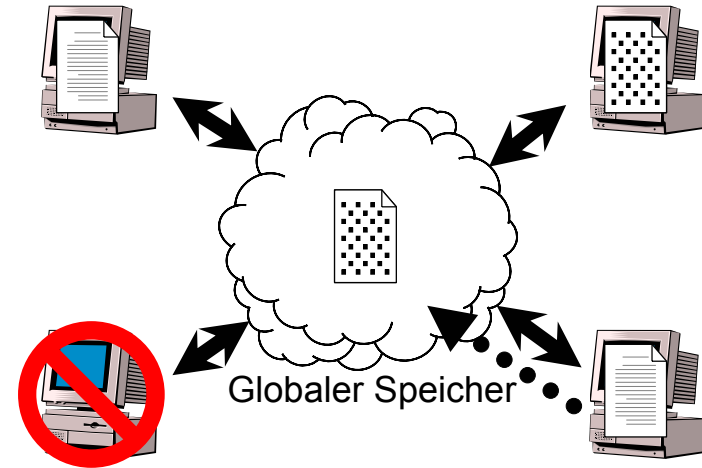
The Far Site



"Don't worry, Mr. Simms. Your data is stored on a three-way redundant server cluster."

by John Douceur of Farsite team, with apologies to Gary Larson

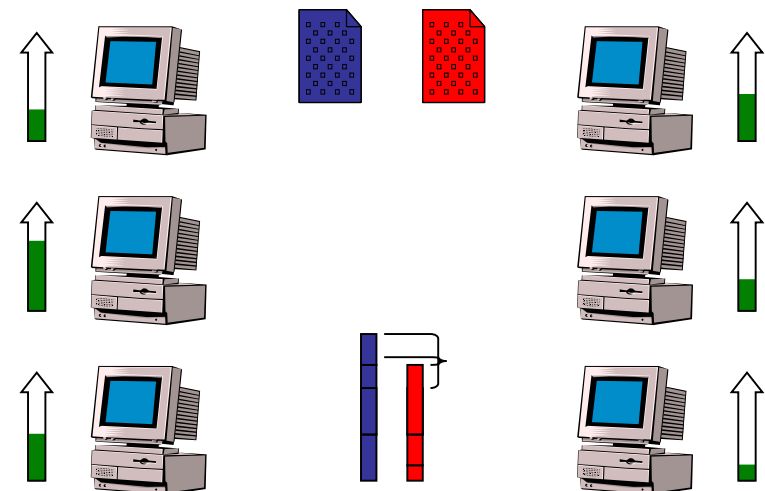
Farsite-System



Farsite: Wichtige Aspekte

- Datenschutz & **Privatsphäre**
 - Verschlüsselung
- Datensicherheit: **Integrität**
 - Kryptographisch sicherer Hashwert
 - Farsite hat byzantinisches Modell
- Datensicherheit: **Langlebigkeit**
 - Replikation (alle Rechner gleich gut)
- **Datenverfügbarkeit** (hier und jetzt)
 - Replikation (Rechner unterschiedlich)


Verbesserung der Verfügbarkeit



Unsere Forschung

- Peer-to-Peer ist mehr als Dateien tauschen
- Wofür ist Peer-to-Peer-Ansatz noch gut?
 - Weltweites Multicast-Streaming (Fussball-WM)
 - Weltweites Namenssystem (DNS)
 - Fehlertolerante Informationssysteme
 - Oder ganz allgemein: Dezentrale und fehlertolerante Strukturen (Datenstrukturen, Algorithmen, Koordinationsstrukturen)

Projekte

- Einerseits: P2P Theorie & Algorithmik
- Andererseits: Wir bauen & testen Systeme
 - Viele Benutzer & 100% Legal
- Idee: Spiele
 - PeerMan, PeerBlast
 - Jassen: Kooperation mit 



Übersicht

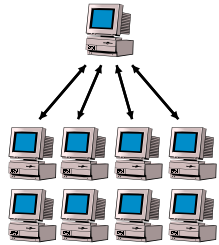
- Motivation
 - Client/Server, Napster, Gnutella
 - “Gutes” Peer-to-Peer-System
- Peer-to-Peer Projekte
 - Farsite, PeerMan, ...
 - **Koordinationsstrukturen**
 - Echtzeitservice, Verteilter Zähler
 - Praktische Schwierigkeiten
- Ausblick

Echtzeit-Service

- Alle Peers müssen periodisch Informationen austauschen (“All-to-All”)
- Beispiel Action-Spiel
 - Alle Spieler müssen allen anderen Spielpartnern 4x pro Sekunde ihre eigene Position mitteilen.

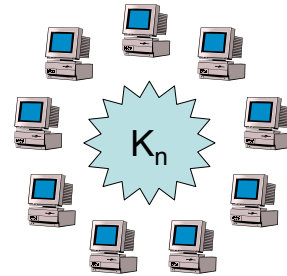
Einfache Lösungen

- Client/Server



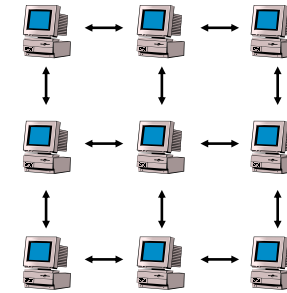
- Nicht fehlertolerant
- Server verarbeitet $O(n)$ Nachrichten

- All-to-All



- Jeder Peer verarbeitet $O(n)$ Nachrichten!

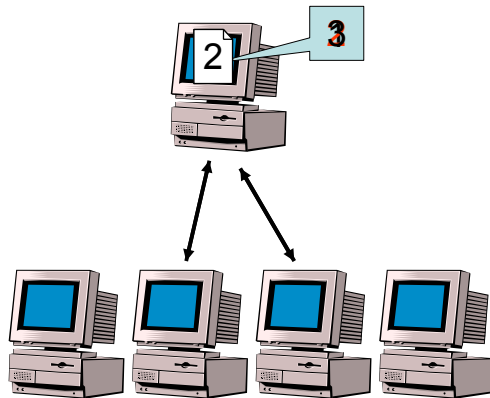
“Peer-to-Peer” Lösung



- Vorteil: Jeder Peer muss nur noch $O(\sqrt{n})$ Nachrichten austauschen
- d -dimensionales Gitter: $O(n^{1/d})$ Nachrichten, aber auch $O(d)$ Latenzzeit

Verteilter Zähler

- Problem & Client/Server-Lösung

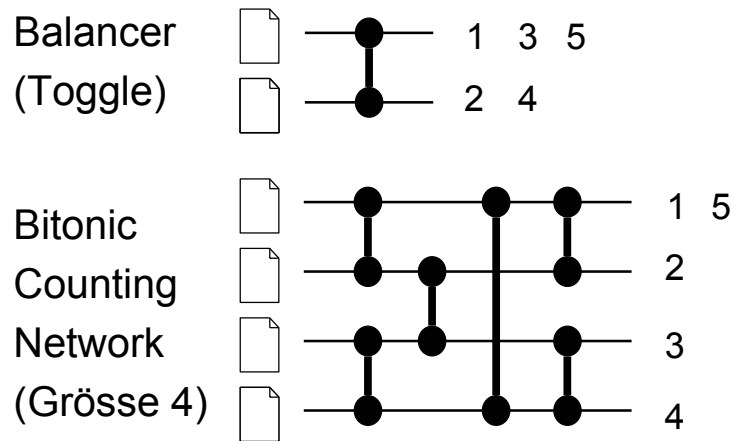


- Anfrage heisst “fetch & inc”
- ist atomar

Verteiltes Zählen

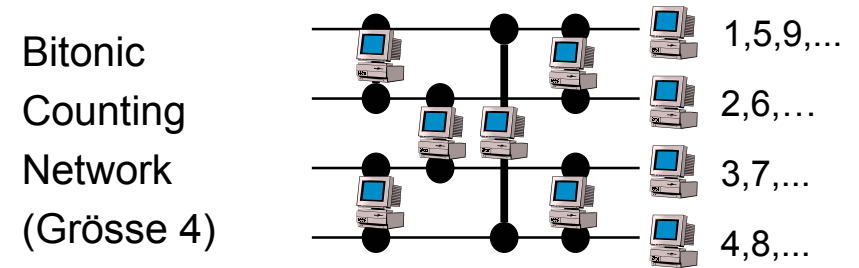
- Nach k Aufrufen von „fetch & inc”
 - sind genau die Zahlen $1, \dots, k$ verteilt
 - ist der “globale” Zählerwert $k+1$
- Eine P2P-Lösung scheint illusorisch
 - Es gibt aber sehr elegante Lösungen...

Counting Networks



Counting Networks

- Implementation: Peers simulieren Balancer und „Minizähler“.



Übersicht

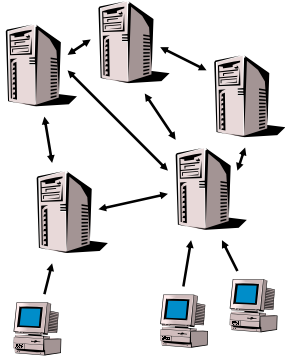
- Motivation
 - Client/Server, Napster, Gnutella
 - “Gutes” Peer-to-Peer-System
- Peer-to-Peer Projekte
 - Farsite, PeerMan, ...
 - Koordinationsstrukturen
 - Echtzeitservice, Verteilter Zähler
 - **Praktische Schwierigkeiten**
- Ausblick

Praktische Schwierigkeiten

- **Firewalls & Network Address Translation**
 - Rechner hinter Firewall oder NAT kann nicht von andern Rechnern angesprochen werden, kann also kein Server oder Peer sein
- **Sicherheit & Krypto**
 - Sicherheit schwieriger als bei Client/Server

Client/Peer

- Statt Server gibt es “Peer-Farm”




- Organisiert als Peer-to-Peer-System
- Client kontaktiert beliebigen (lokalen) Rechner der Peer-Farm
- Plug-and-Play

Übersicht

- Motivation
 - Client/Server, Napster, Gnutella
 - “Gutes” Peer-to-Peer-System
- Peer-to-Peer Projekte
 - Farsite, PeerMan, ...
 - Koordinationsstrukturen
 - Echtzeitservice, Verteilter Zähler
 - Praktische Schwierigkeiten
- **Ausblick**

Andere Projekte

- Mobile Computing
 - Ad-Hoc Networking  NCCR/MICS
- Peer-to-Peer-Theorie
 - Was kann man nicht dezentral machen?
 - Was kann man wie effizient dezentral machen?
- Neue algorithmische Paradigmen
 - Dynamic, Local, Online, Selfish, Malicious, ...

