

P2P DATENBANKEN

Anwendungen 1 – WS 2009/2010

Julissa Cusi Juarez

**Department Informatik
HAW Hamburg**

02.12.2009

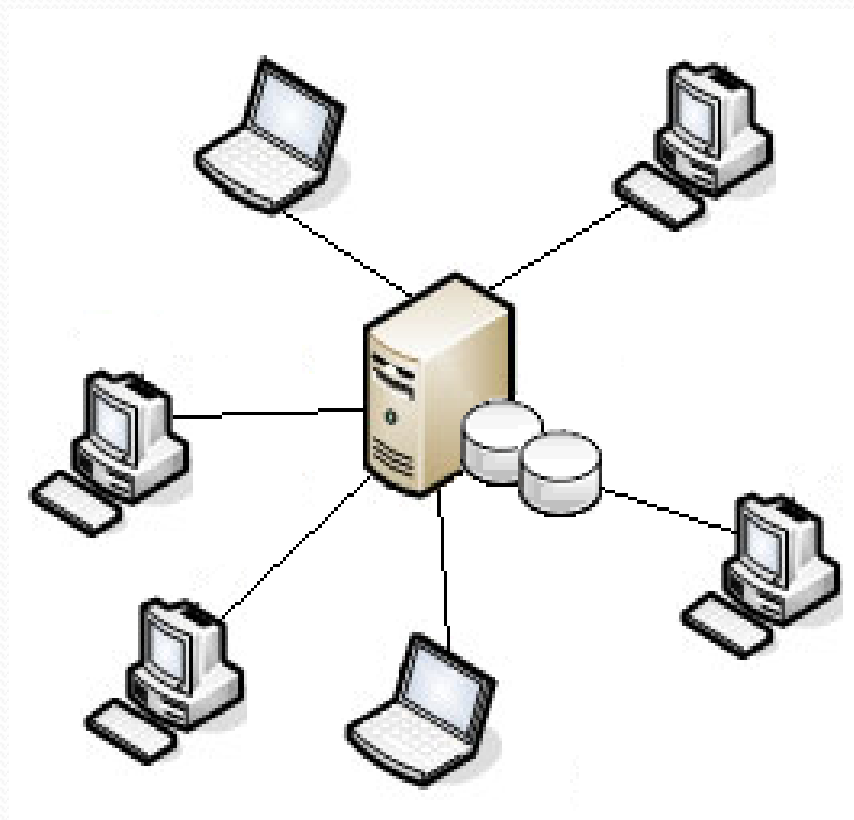
Agenda

1. Motivation
2. Klassische Datenbanksysteme
3. Verteilte Datenbanksysteme
4. Peer-to-Peer (P2P)
5. P2P Datenbanksysteme
6. Query Processing in P2P Datenbanksystemen
 - 6.1. Beispiel: Skyframe
7. Zusammenfassung
8. Ausblick
 - a. Literatur

1. Motivation

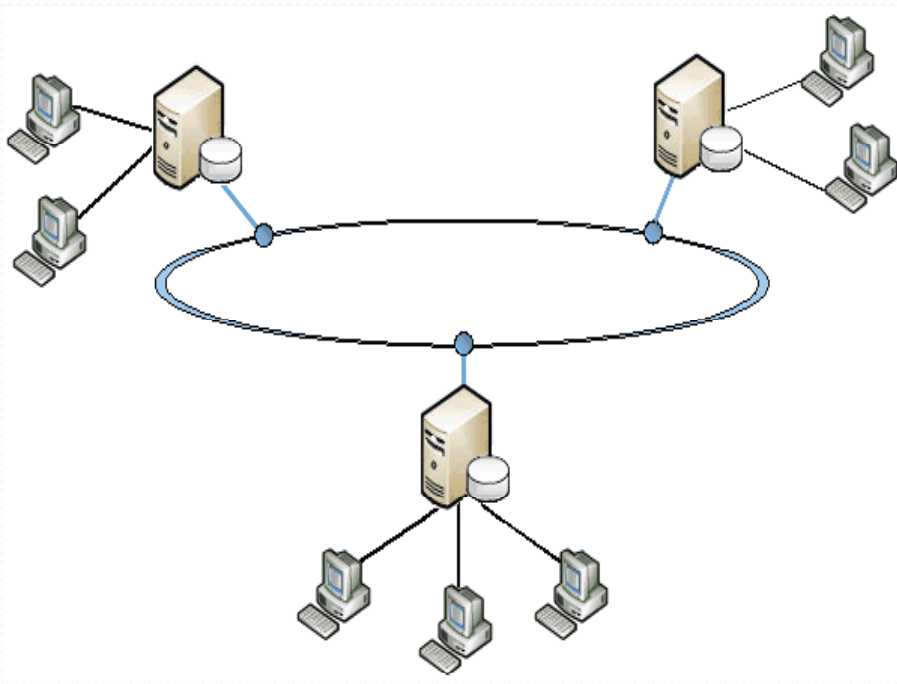
- Entwicklung der Endbenutzer-Kapazitäten
 - Hohe Rechenleistung
 - Hohe Speicherkapazität
- Entwicklung der Kommunikations-Netzwerke
 - Breitbandverbindung verfügbar
 - Flatrate
- Content Sharing Websysteme
 - Hoher Bedarf, Information auszutauschen
 - Hohe Anzahl von Benutzern weltweit

2. Klassische Datenbanksysteme



- Basierend auf Client/Server Model
- Zentralisiertes DBS, paßt gut zu zentralisierten Unternehmen
- Zentrale Administration des Servers
- Steuert relationale Datenbanken
- Benutzt SQL Anfragesprache

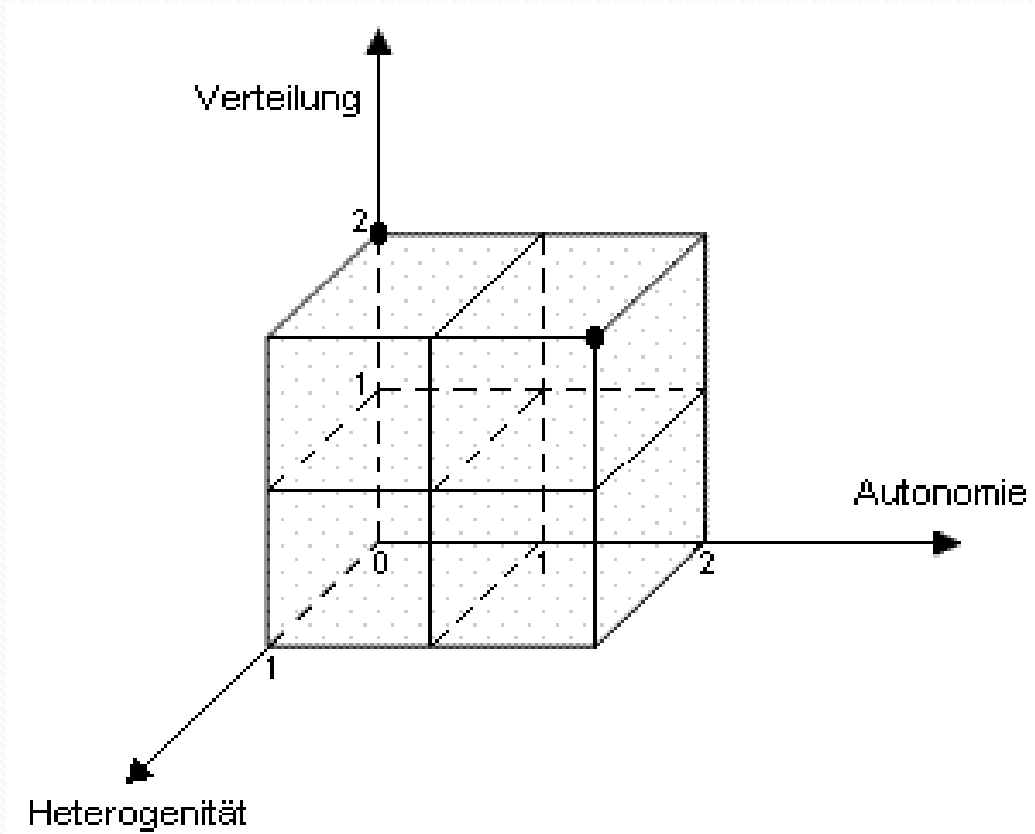
3. Verteilte Datenbanksysteme



- Ein Verteiltes Datenbankmanagementsystem ist eine Software, die eine oder mehrere logisch zusammenhängende Datenbanken über ein Netzwerk steuert.

3. Verteilte Datenbanksysteme

Klassifikation



- Dimension Autonomie
0: Vollständige Integration
1: Semi-autonome System
2: Vollständige Autonomie
- Dimension Verteilung
0: Nicht verteilt (zentralisiert)
1: Client / Server System
2: Verteilung Peer-to-Peer
- Dimension Heterogenität
0: Homogenes System
1: Heterogenes System

3. Verteilte Datenbanksysteme

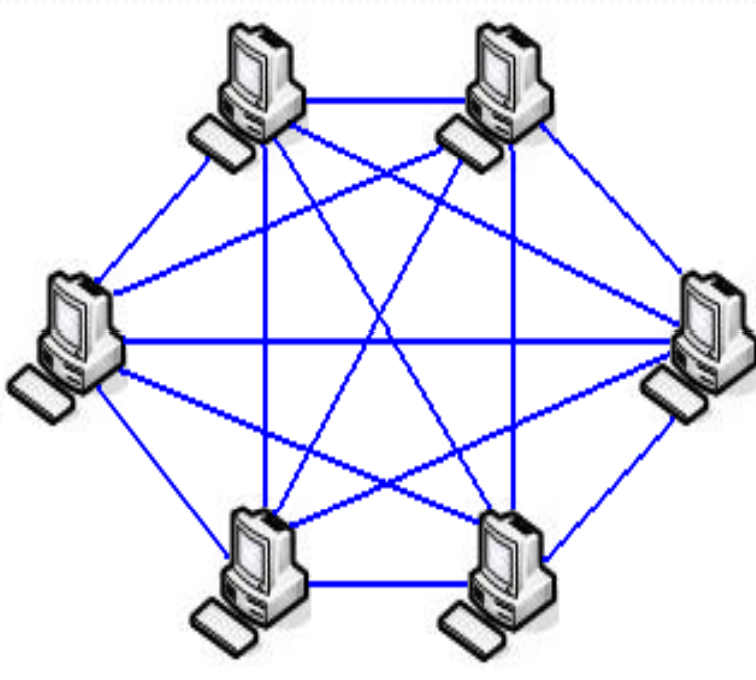
Fragmentierung: Verteilung der Daten auf Knoten

- Horizontal (Zeilen)
- Vertikal (Spalten)
- Hybrid

Replikation : Absichtlich redundante Speicherung der Daten auf den Knoten

- Vollständige Replikation
- Partielle Replikation
- Keine Replikation

4. Peer-to-Peer (P2P)



- Overlay Architektur
- Dezentralisiertes und selbstorganisiertes System
- Das System ist dynamisch
- Peers sind gleichberechtigt und autonom
- Kein Peer hat einen Überblick über das Gesamtsystem
- Benutzt verteilte und geteilte Ressourcen

4. Peer-to-Peer (P2P)

Eigenschaften

- **Skalierbarkeit** bezüglich der Knotennummer
- **Dezentralisiert** in der reinsten Form von P2P Netzen, gibt es kein Konzept von „Server“
- **Autonomie** bezüglich Speicherung, Ausführung, Standzeit und Verbindung
- **Direkter Zugriff** auf Daten der verschiedenen Knoten
- **Robustheit und Härting** gegen Peer-Ausfälle
- **Einfache Entwicklung**, weil die Knoten keine besondere Infrastruktur brauchen

4. Peer-to-Peer (P2P)

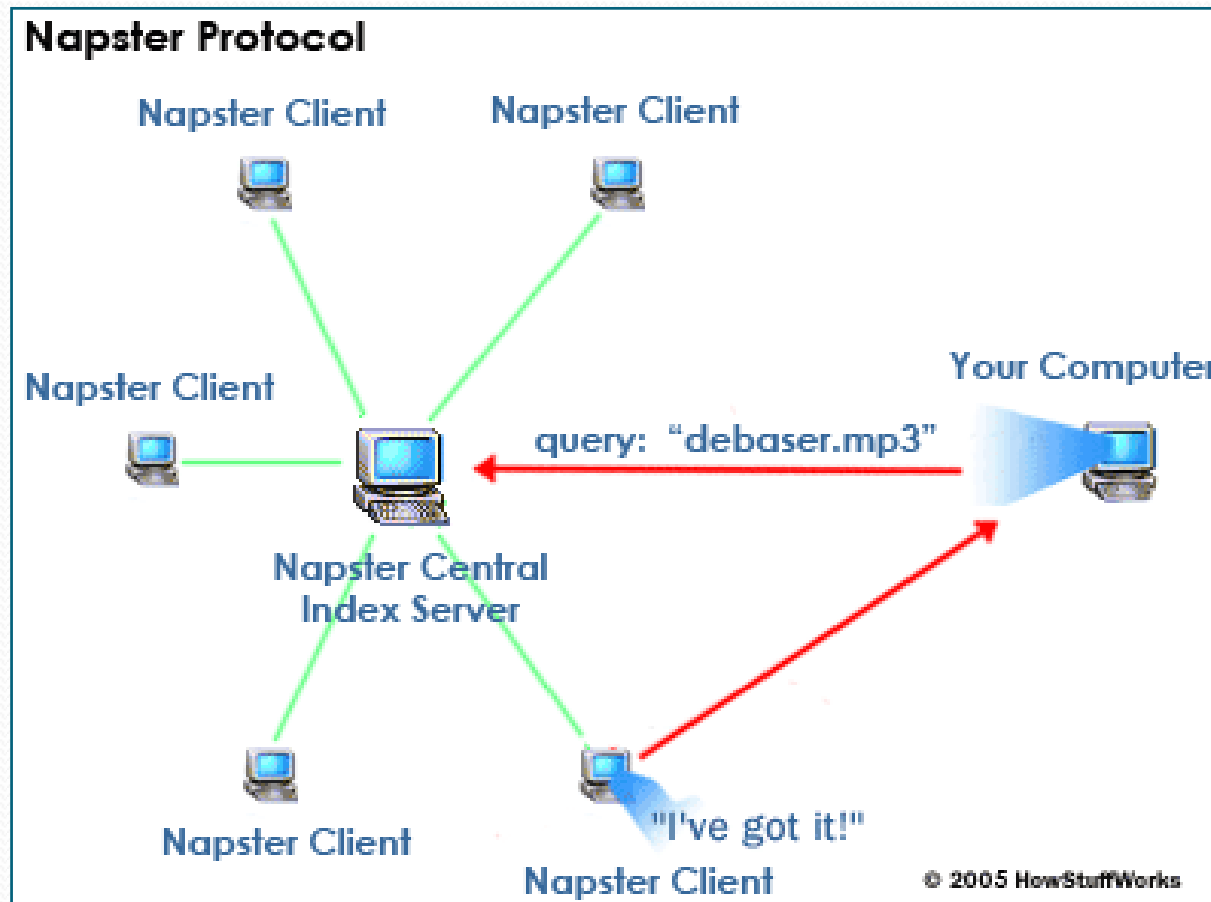
Unstrukturierte P2P Netze

- Overlay Links beliebig etabliert
- Anfragebearbeitung durch „Flutung“
- Manchmal können die Anfragen nicht beantwortet werden
- Flutung erzeugt hohen Kommunikationsoverhead

Typen	Beispiel
Zentralisierte P2P-Systeme	Napster
Reine P2P-Systeme	Gnutella 0.4, Freenet
Hybride P2P-Systeme	Gnutella 0.6, Gnutella2, JXTA

4. Peer-to-Peer (P2P)

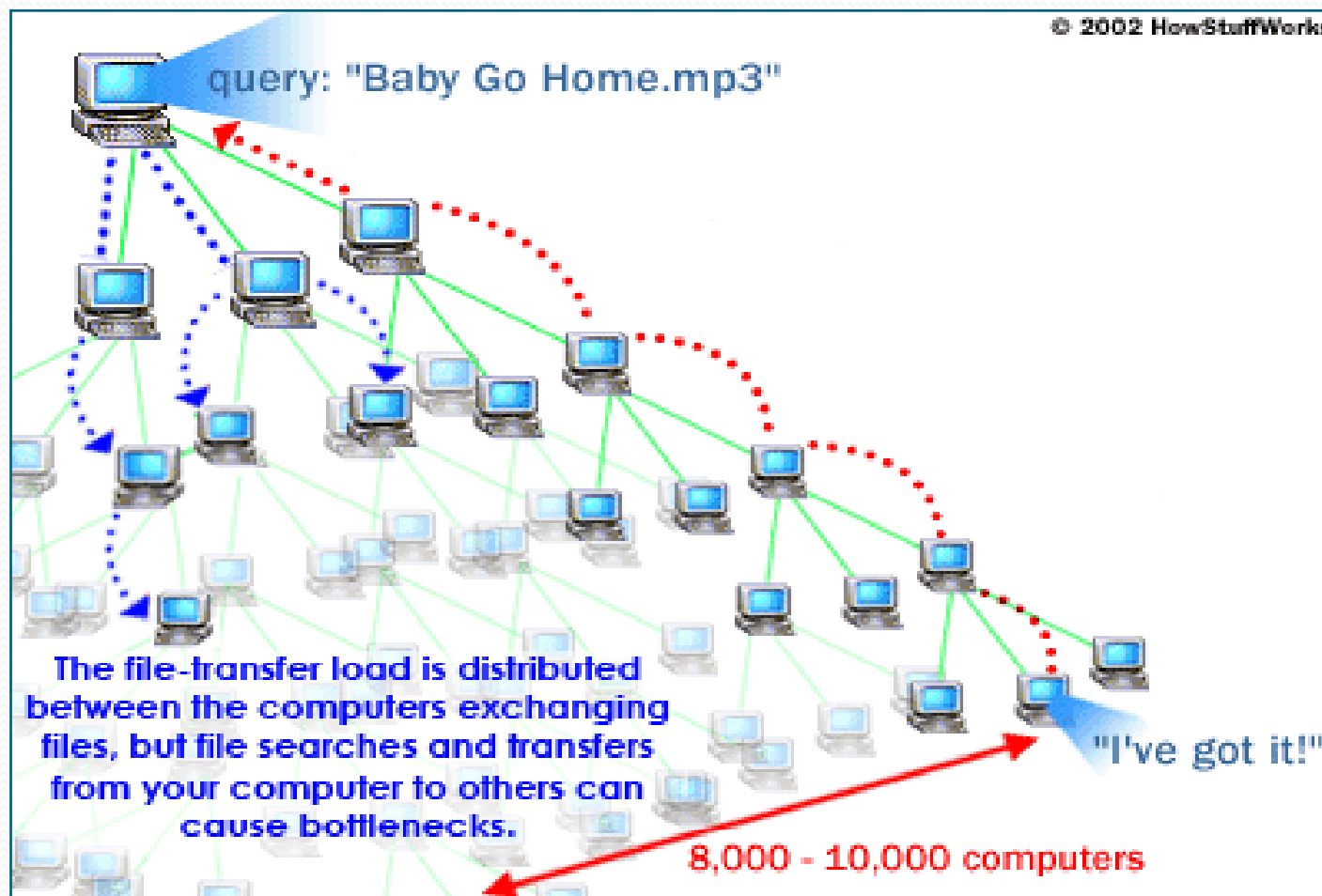
Unstrukturierte Systeme, zentralisiertes P2P: Napster



Napster Protokoll [8]

4. Peer-to-Peer (P2P)

Unstrukturierte Systeme, reines P2P: Gnutella 0.4

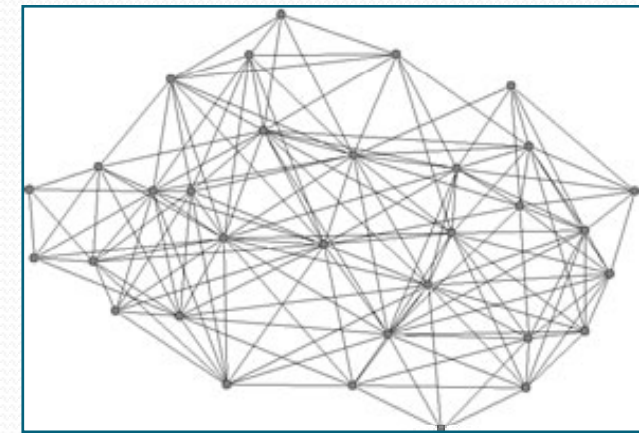


Gnutella Protokoll [8]

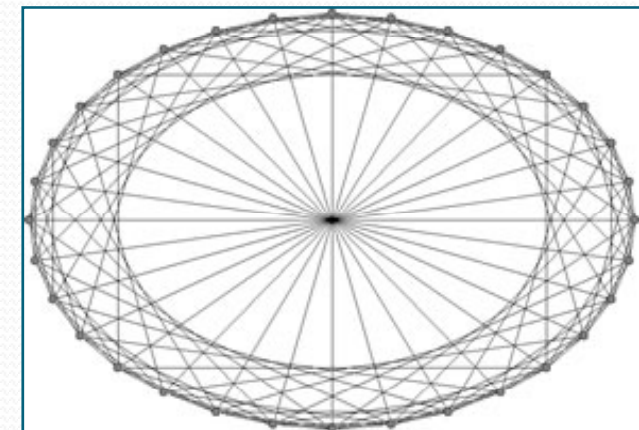
4. Peer-to-Peer (P2P)

Strukturierte P2P Netze

- Benutzen Distributed Hash Table (DHT)
- Jeder Peer ist verantwortlich für einen bestimmten Teil des Netzwerkgehaltes
- Das Protokoll wird benutzt, um die geeigneten Peers zu finden
- Höhere Erhaltungskosten
- Beispielsysteme: CAN, Chord



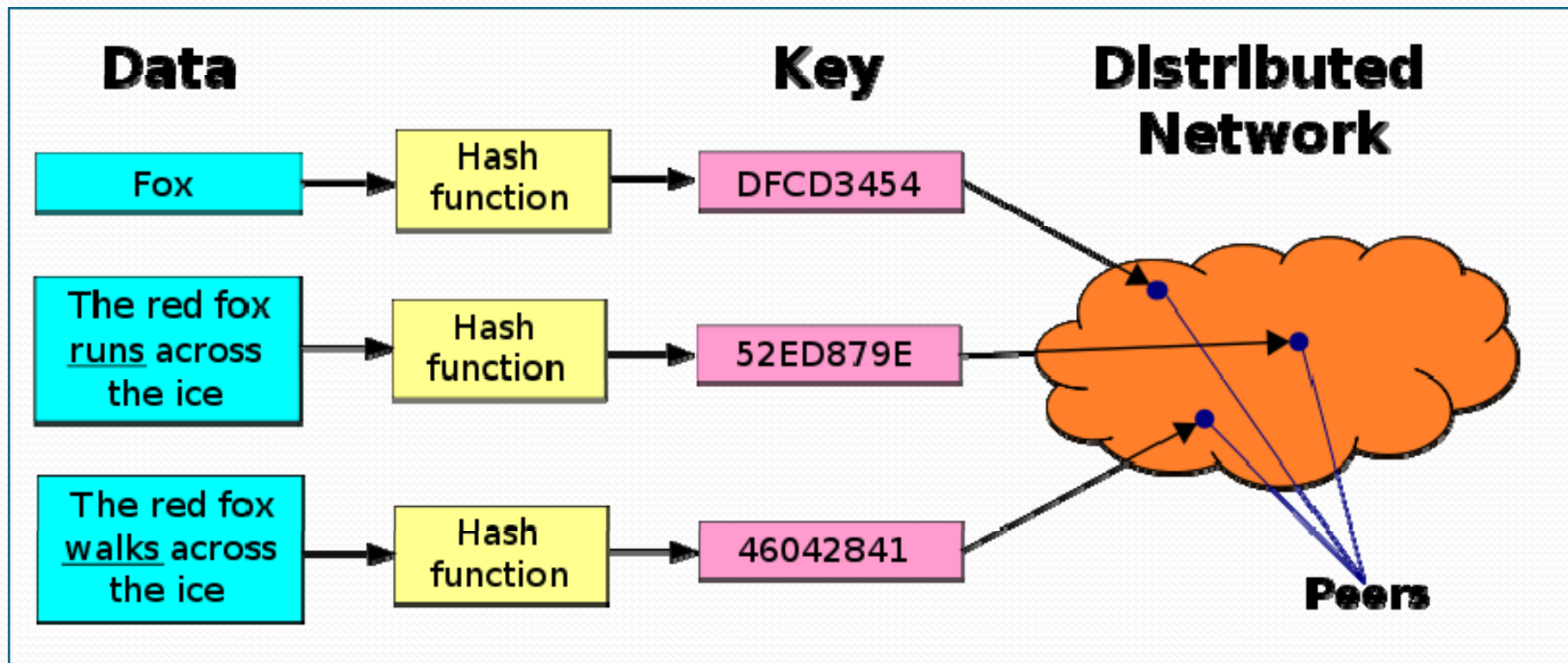
CAN



Chord

4. Peer-to-Peer (P2P)

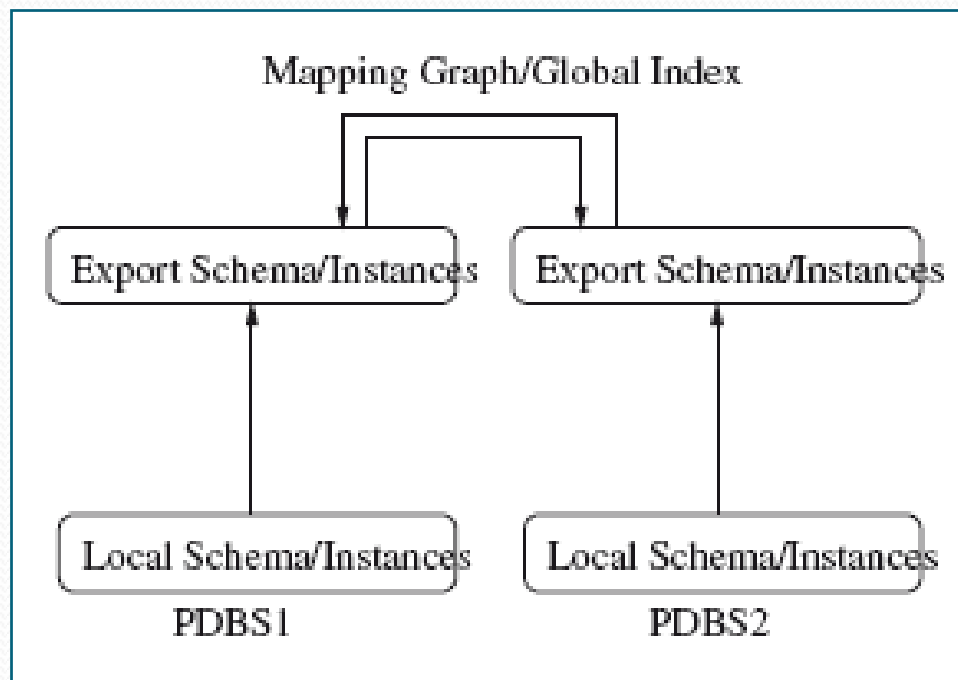
Strukturierte Systeme : Distributed Hash Tables (DHT)



Distributed Hash Tables [7]

5. P2P Datenbanksysteme

- Eine Kollektion vom lokalen autonomen Datenbehältern (Peers)
- Man kombiniert die dezentralisierten Eigenschaften und die Autonomie der P2P Systeme mit der Semantik der VDBS
- Man vermeidet globale Datenbankschemata



Integrationarchitektur der Daten [1]

5. P2P Datenbanksysteme

Vorteile

- P2P-Vorteile bleiben erhalten, z.B. hohe Skalierbarkeit
- Keine besondere Infrastruktur in den Knoten erforderlich

Nachteile

- Keine Garantie alle Antworten zu erhalten
- SQL typischerweise nicht unterstützt

6. Query Processing in P2P DBS

- Anfragen werden durch das Overlaynetz verbreitet
- Man benutzt spezifische Indexes
- Anfragesprache oft begrenzt und von der Art des Netzwerkes abhängig
- Es gibt keine Garantie, alle Antworten zu erhalten
- Benutzt Replikation- und Laufzeittoleranztechniken

6. Query Processing in P2P DBS

Abhängigkeiten vom verfügbaren Index

Typen	Beschreibung	
Kein Index	Routing durch Flutung (wie in Gnutella 0.4)	
Zentralisierter Index	Ein Peer hat die Information über den Inhalt von allen Knoten. Bottleneck (wie in Napster)	
Verteilter Index (hängt von Overlaytopologie ab)	Unstrukturierten Systeme	Routing Indexes, ein Peer hat Information über den Inhalt von erreichbaren Knoten
	Strukturierten Systeme	Jeder Peer hat Indexinformation aller Daten über die Hash Funktion

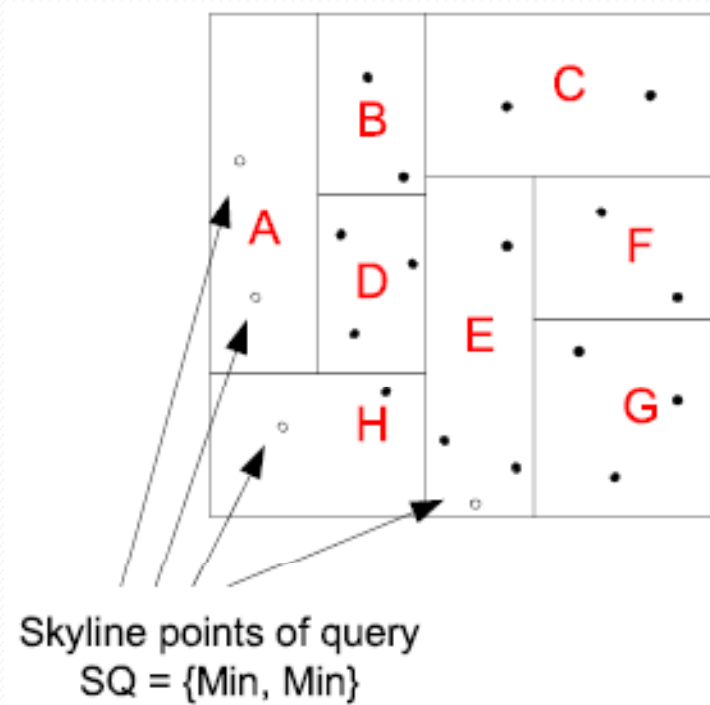
6. Query Processing in P2P DBS

Mögliche Datenmodelle und Anfragesprachen

Datenmodell	Anfragesprache
XML	XPath, XQuery
RDF	SPARQL, Logic Language

6.1. Beispiel: Skyframe

Skyline Query

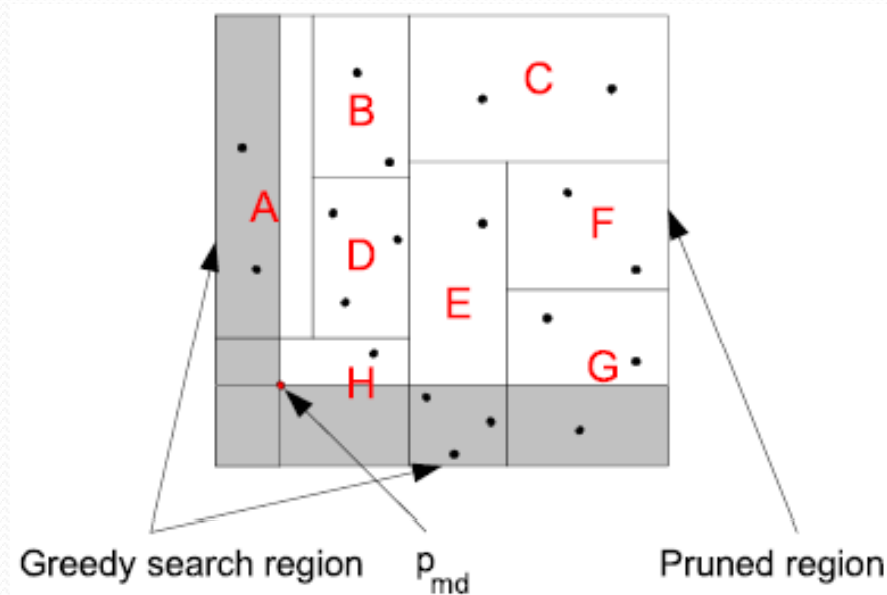


Beispiel von Skyline Query im zweidimensionalen Raum [6]

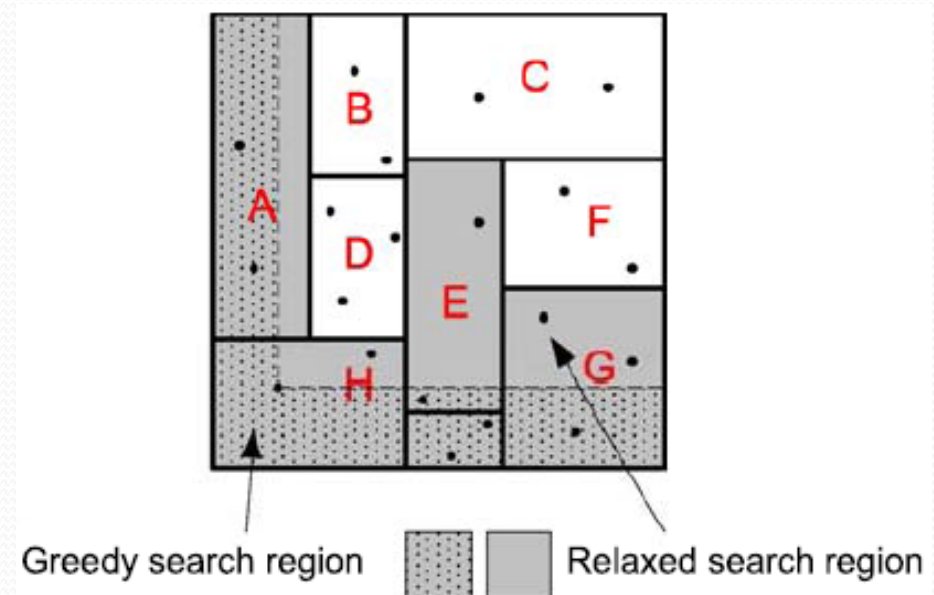
6.1. Beispiele: Skyframe

Architektur

1. Query Processing Manager: Greedy Skyline Search (GSS) und Relaxed Skyline Search (RSS)



Greedy Skyline Search [6]



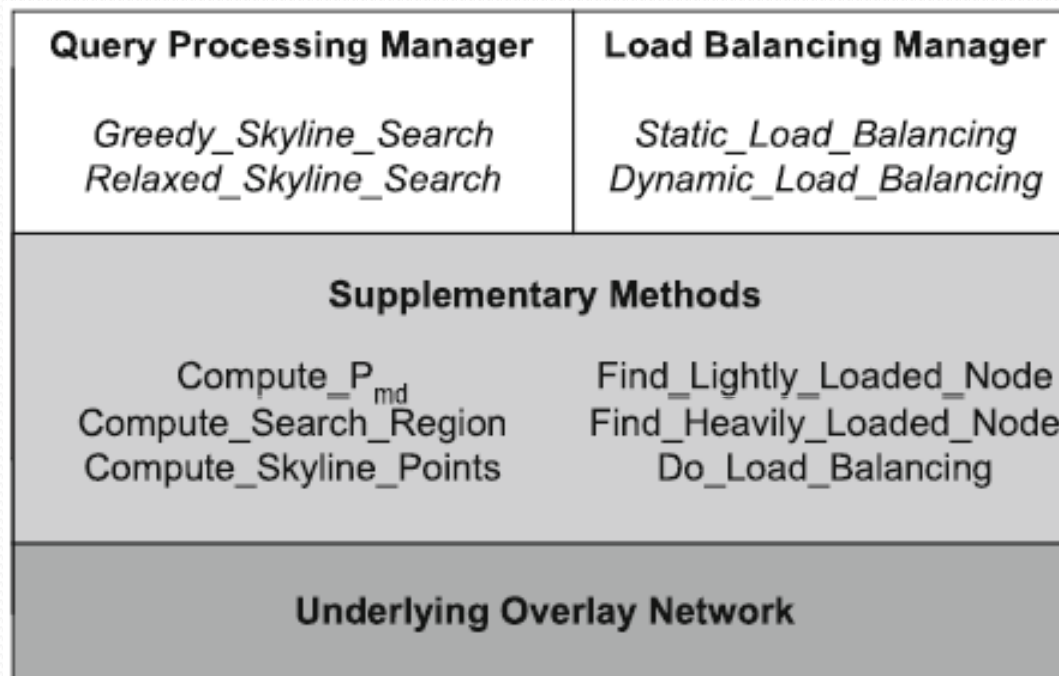
Relaxed Search Space [6]

6.1. Beispiele: Skyframe

Architektur

2. Query load-balancing Manager
 - Query load balanced Partition
 - Dynamic query load balancing

3. Skyframe



Skyframe [6]

7. Zusammenfassung

- Ein P2P DBS kombiniert die dezentralisierten Eigenschaften und die Autonomie der P2P Systeme mit der Semantik der VDBS.
- Es gibt verschiedene Query Processing Methoden je nach Overlayarchitektur.
- Es gibt verschiedene Datenmodelle und Anfragesprachen, je nach P2P Netz.
- Skyframe ist ein beispielhaftes Framework, das den Zeitverlauf optimiert und niedrige Netzwerkkommunikationskosten erreicht.

8. Ausblick

- Entwicklung von einem Query-Processing-Verfahren für ein strukturiertes P2P-System
- Anwendung des Verfahrens in typischen Szenarien
- Vergleich des entwickelten Verfahrens mit existierenden Verfahren für das betrachtete Szenario

A. Literatur

- [1] A. Bonifati, P. K. Chrysanthis, A. M.Ouksel, K. Sattler. *Distributed Databases and Peer-to-Peer Databases: Past and Present*. In ACM SIGMOD Record, Vol.37 No.1, 2008.
- [2] A. Eyal, A.Gal. *Self Organizing Semantic Topologies in P2P Data Integration Systems*. In IEEE International Conference on Data Engineering, 2009.
- [3] X. Shen, Z. Li. *Implementing Database Management System in P2P Networks*. In International Seminar on Future Information Technology and Management Engineering , 2008.
- [4] M. Karnstedt, K. Sattler, M. Haß, M. Hauswirth, B. Sapkota, R. Sachmidt. *Estimating the Number of Answers with Guarantees for Structured Queries in P2P Databases*. In ACM Conference on Information and Knowledge Management CIKM'08, 2008.

A. Literatur

- [5] G. Koloniari, E. Pitoura. *Peer-to-Peer Management of XML Data: Issues and Research Challenges*. In ACM SIGMOD Record, Vol.34 No.2, 2005.
- [6] S. Wang, Q. H. Vu, B. C. Ooi, A. K. H. Tung, L. Xu. *Skyframe: a framework for skyline query processing in peer-to-peer systems*. In VLDB Journal, Vol.18, No.1, 2009.
- [7] http://en.wikipedia.org/wiki/Distributed_hash_table
- [8] http://www.umkc.edu/is/security/p2p_explanation.asp

Vielen Dank für Ihre
Aufmerksamkeit !