Forschungszentrum L3S

What does it take to build Google2Google

or

How do you find what you need if it is distributed all over the world and more complicated than a web page?

Wolfgang Nejdl

L3S and University of Hannover

Germany

Overview

Motivation and Background
- L3S Projects & the Semantic Web

Peer-to-Peer Data Management
- Edutella Framework and Infrastructure
- Peer-to-Peer Topologies
- Peer-to-Peer Data Retrieval
- Google2Google

Summary and Conclusions

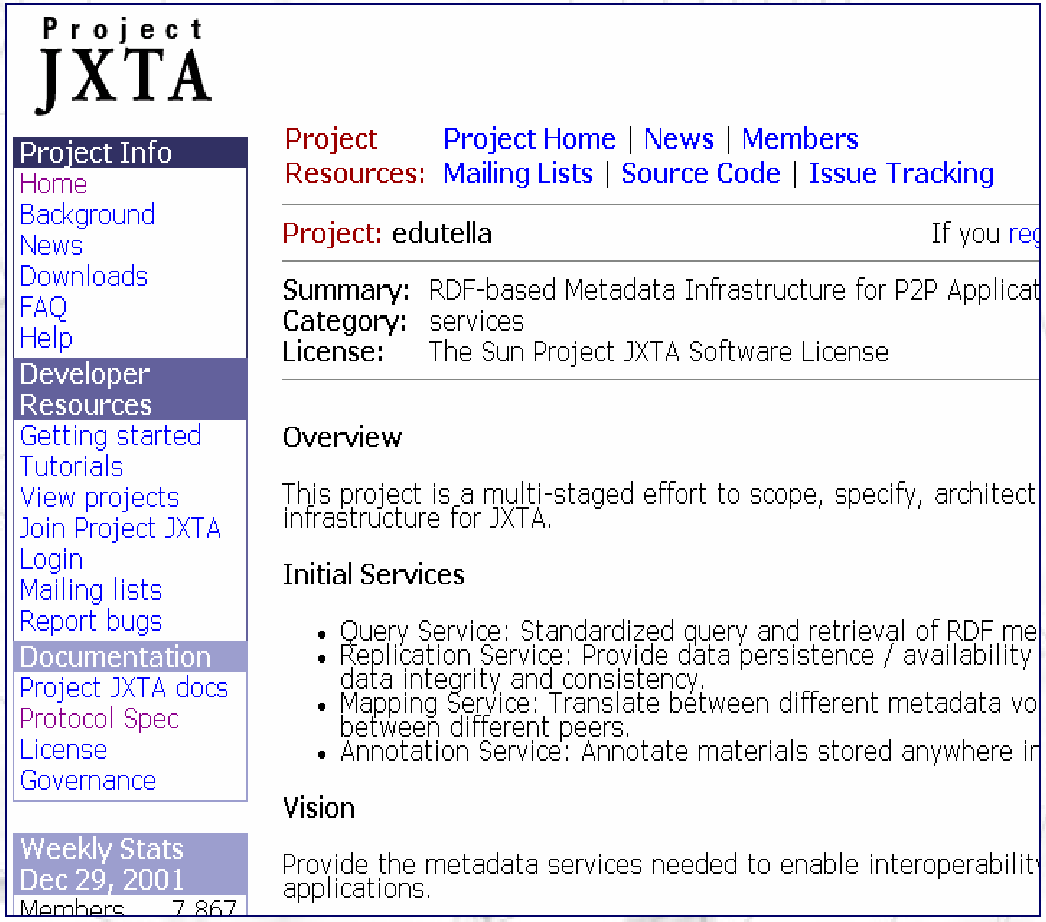
Databases & Semantic Web: Edutella

Specify and implement a RDF-based meta-data infra-structure for P2P networks

Developed as part of the open source peer-to-peer project JXTA, at edutella.jxta.org

60+ contributors from various institutions

Building block for EU/IST FP5 ELENA Smart Learning Space (personalized, distributed queries)



3 New EU/IST FP6 Networks of Excellence @ L3S

Starting in 2004: 3 New EU/IST FP6 Networks of Excellence in the areas of

Technology Enhanced Learning

- NoE PROLEARN (coordinated by L3S)

Knowledge Technologies and Semantic Web

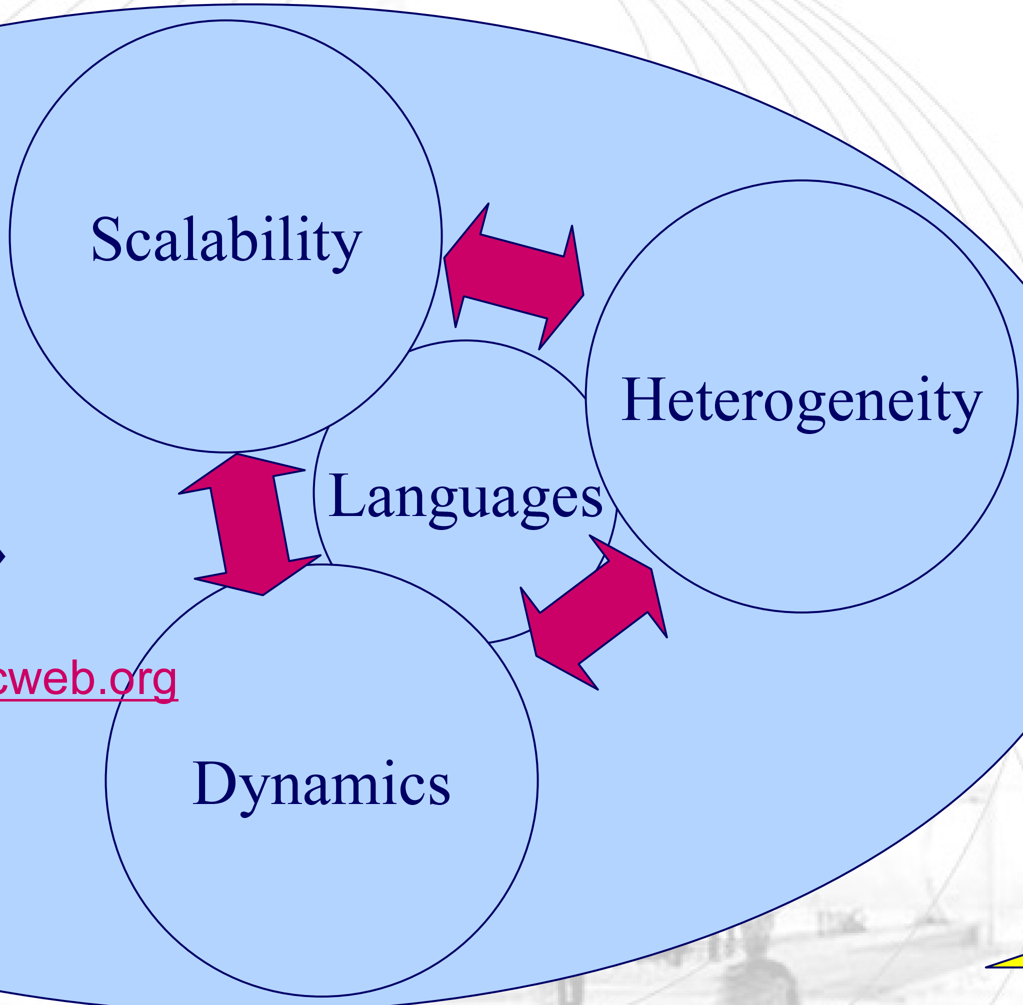
- NoE KnowledgeWeb (L3S as Core Partner)
- NoE REWERSE (L3S as Core Partner)

Duration of Networks: 4 years

Number of Partners: 20 (30 for REWERSE)

Budget per Network: 6-7 Mill. Euro

Scalability, Heterogeneity and Dynamics: KnowledgeWeb



«It's distributed information, stupid»

knowledgeweb.semanticweb.org

Overview

Motivation and Background

- L3S Projects & the Semantic Web

Peer-to-Peer Data Management

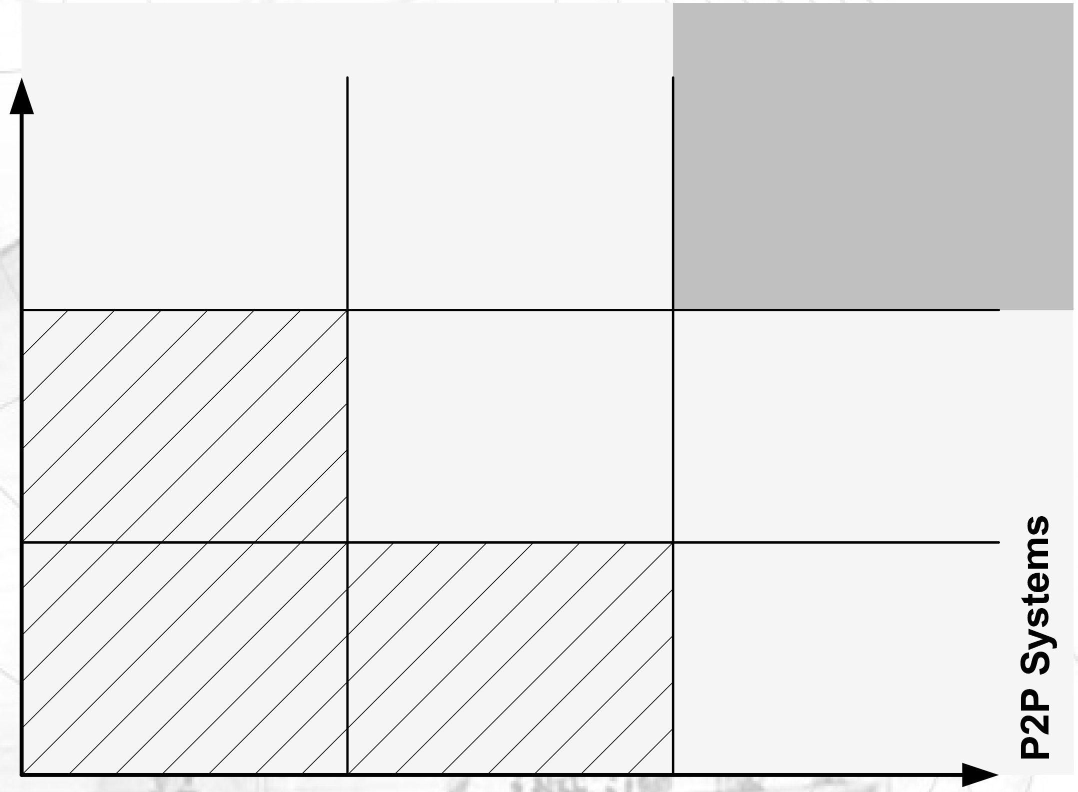
- Edutella Framework and Infrastructure
- Peer-to-Peer Topologies
- Peer-to-Peer Data Retrieval
- Google2Google

Summary and Conclusions

Edutella and Schema-Based Peer-to-Peer Networks

User-definable schemas
Structured schemas
Query language

Decentralized control
Node autonomy
Transient peers
Self organization



RDF / RDF Schema for Describing Distributed Resources

Basic Formalisms for the Semantic Web

- URIs to identify resources
- Combine resources and annotate resources with attributes, using <Subject, Property, Value> Tuples
- Graph as basic model, easy to translate to logic facts
- RDFS allows us to define the RDF vocabulary used (classes and attributes), and thus to represent simple semantic models
- Possible extensions towards more expressive schema languages, e.g. description logic (DAML+OIL / OWL)

Using RDF / RDFS in the P2P context

- Distributed annotations for distributed resources
- Flexible schema definitions, which can be uniquely identified and combined, as well as extended by additional properties

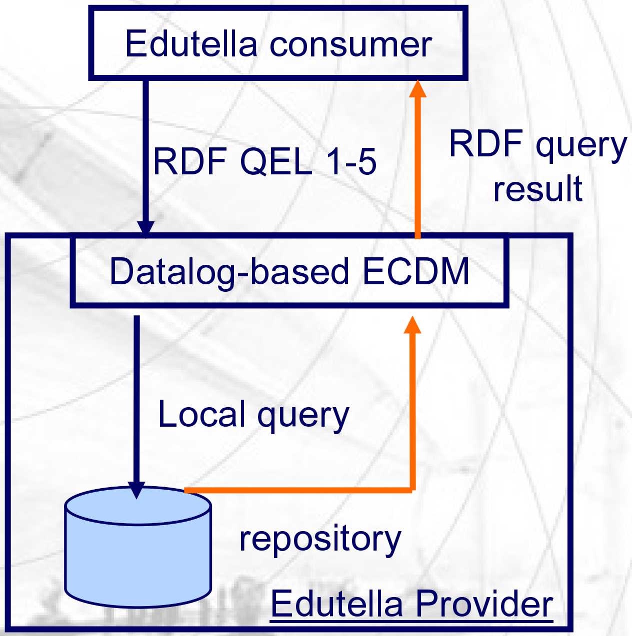
Edutella Architecture: Wrapping with RDF-QEL

Datalog-based Query Exchange Language (RDF-QEL)

- RDF QEL1: conjunctive query up to
- RDF QEL5: RDF QEL4 (SQL3) + general recursion

see Nejdl et al: „EDUTELLA: A P2P Networking Infrastructure Based on RDF“, WWW 2002

- Datalog is used as the internal data model (ECDM: Edutella Common Data Model) and provided as a set of Java classes
- RDF is used to represent the queries transmitted between the peers
- Wrappers for other RDF query languages (RQL, TRIPLE, etc.) and XML query languages (like Xpath)



Edutella query data flow

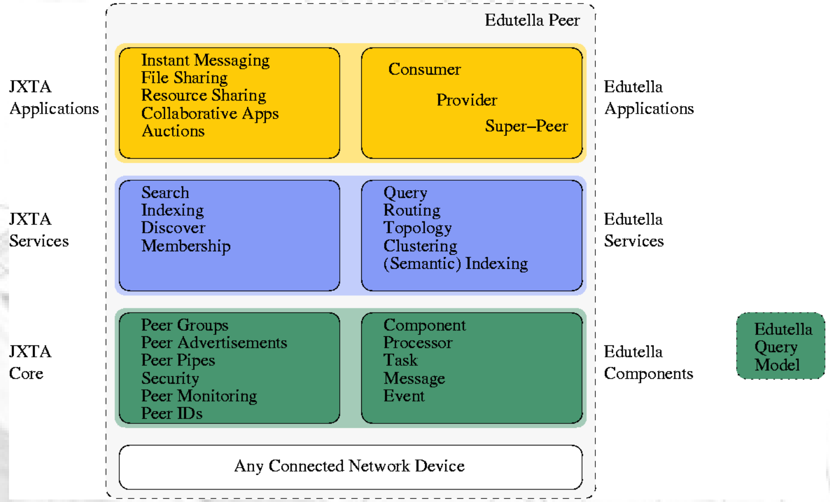
Edutella Architecture

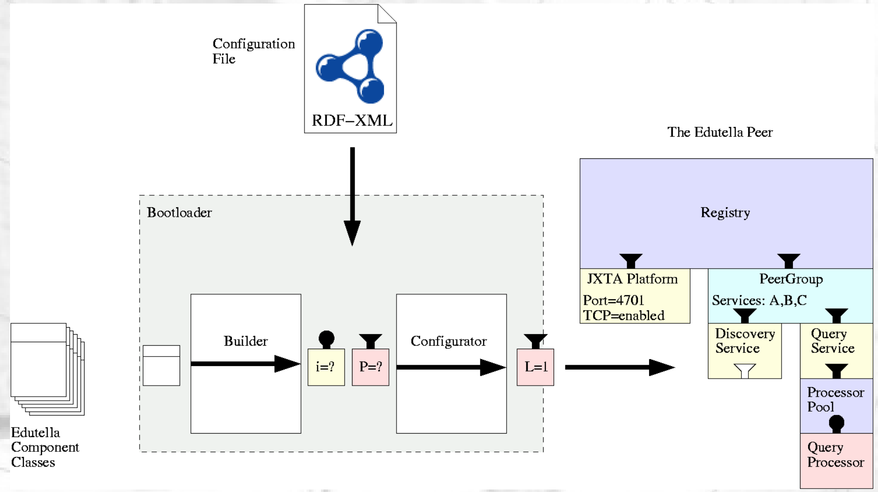
JXTA is build on three layers :
 Core, Services, Applications

Edutella follows this scheme,
 extending the JXTA layers
 with modular Edutella
 components

"Edutella Bootloader" to
 assemble peers

Good prototyping environment
 for experimenting with schema-
 based peer-to-peer algorithms





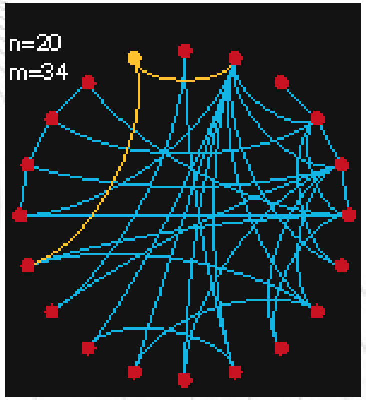
Overview

Motivation and Background
- L3S Projects & the Semantic Web

Peer-to-Peer Data Management
- Edutella Framework and Infrastructure
- Peer-to-Peer Topologies
- Peer-to-Peer Data Retrieval
- Google2Google

Summary and Conclusions

P2P and Efficient Routing

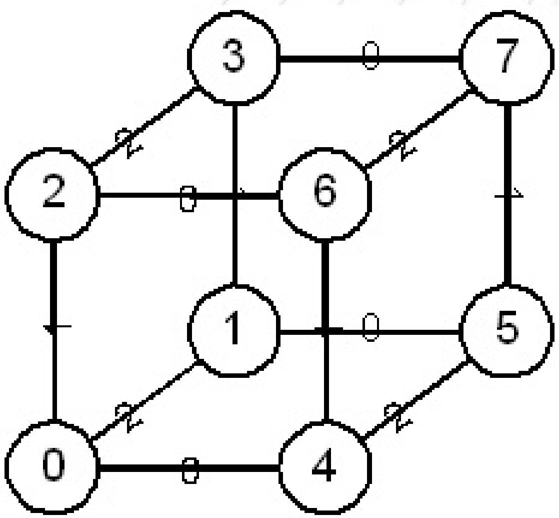


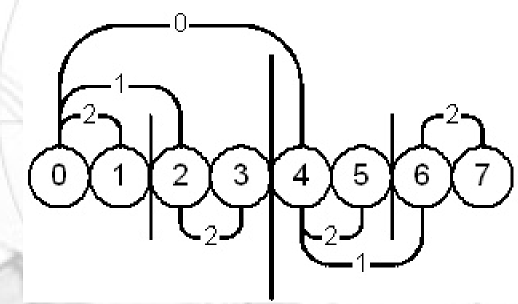
How do peer-to-peer networks scale?

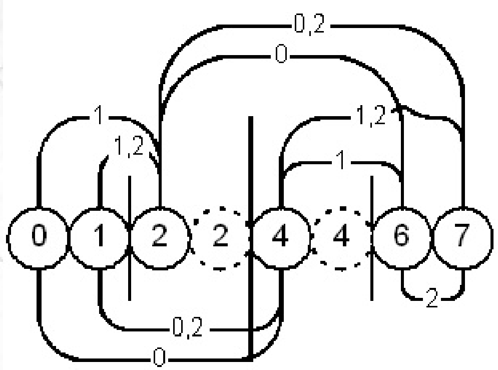
Requirements:

- Symmetric topology (every node is a root)
- Low network diameter (small worlds property, should be O(log n))
- Limited node degrees (number of peer-connections from a node, should be O(log n) at most (or even constant))
- Load balancing of traffic
- Efficient broadcast (receive broadcast messages only once)
- Adaptable to dynamic number of peers

HyperCuP Peer-to-Peer Topology







Details: see e.g. Schlosser, Sintek, Decker, Nejdl: „HyperCuP – Shaping Up Peer-to-Peer Networks“, 2nd Intl. WS on Agents and P2P Computing, 2002

Hypercube Topology

Broadcast Algorithm

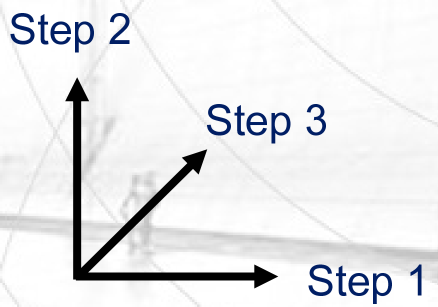
- Annotate messages with the “dimension” of the peer-to-peer connection, and only forward it along “higher” dimensions

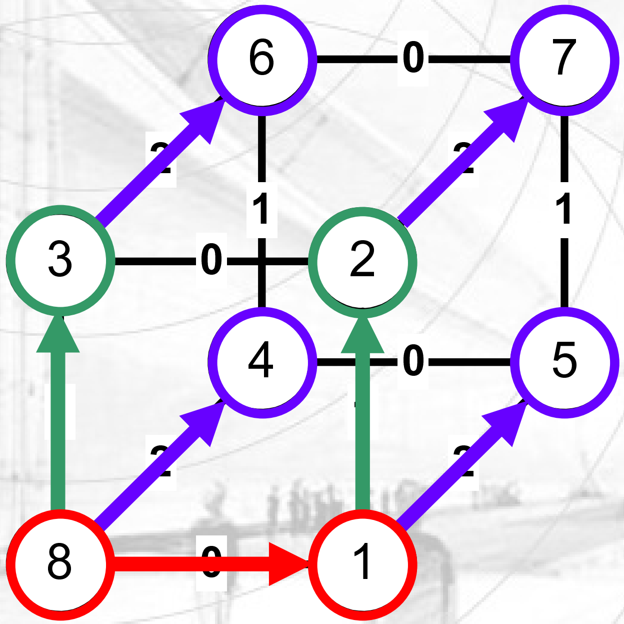
Properties

- Network diameter, characteristic path length and number of nodes are $O(\log_b N)$
- Fault tolerant, vertex-symmetric

Extensions

- Dynamic hypercube
- Base=N hypercube
- Cayley graphs





P2/DHTs and Static P2P/DHT Topologies

DHTs	static DHT topologies	(avr.) degr.	diameter
HyperCup	hypercubes	$O(\log N)$	$O(\log N)$
Chord	ring graphs	$O(\log N)$	$O(\log N)$
Pastry/Tapestry	Plaxton trees	$O(\log N)$	$O(\log N)$
Viceroy	butterfly	$O(1)$	$O(\log N)$
Cycloid	cube conn. cycles	$O(1)$	$O(d)$, $N = d \cdot 2^d$
Koorde/D2B	de Bruijn	$O(1)$	$O(\log N)$
Distance Halving	k-base de Bruijn	$O(k)$	$O(\log N/\log k)$
CAN	d-dimensional torus	$O(d)$	$O(d \cdot N^{1/d})$

- Non-constant degree DHTs: HyperCup (hypercubes), Chord (ring graphs), Pastry/Tapestry (Plaxton trees)
- Constant degree DHTs: Viceroy (butterfly), Cycloid (cube connected cycles), Koorde/D2B/Distance Halving (de Bruijn), CAN (d-dimensional torus).

Cayley DHTs: A Group-Theoretic Model for Analysing DHTs based on Cayley Graphs

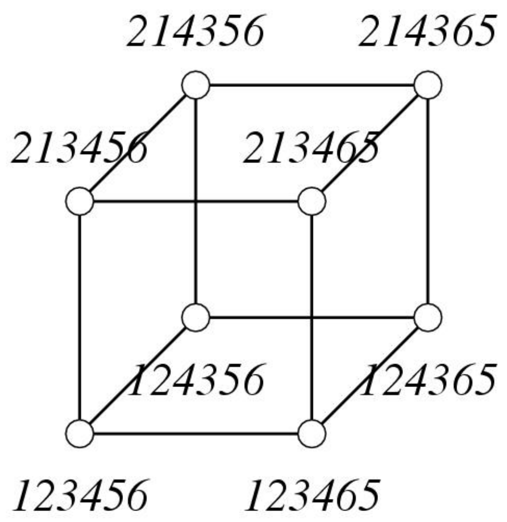
- Cayley graphs: Given a set of generators for a finite group G, we can draw a graph, called Cayley graph, in which the vertices correspond to the element of the group G and the edges correspond to the action of the generators. That is, there is an edge from an element a to an element b if there is an generator g such that ag=b. The set of generators is required to be closed under inverses so that the resulting graph can be viewed as being undirected.
- Features of Cayley graphs:

  Universality: Cayley graphs embody almost all symmetric interconnection networks, and every symmetric interconnection network can be represented as the quotient of two Cayley graphs.

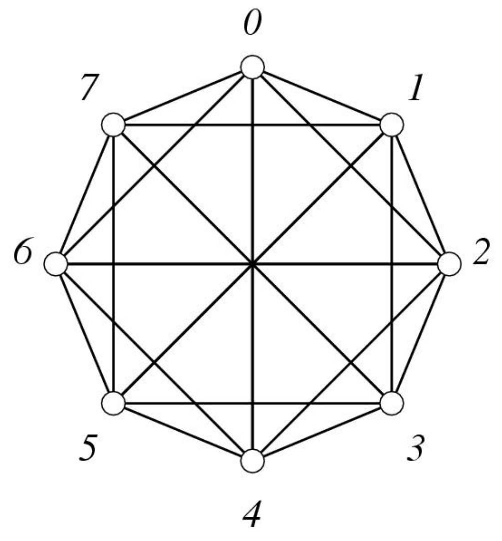
  High performance: Cayley graphs represent a class of interconnection networks with small degree and diameter, good connectivity, and simple routing algorithms.

Non-constant degree Cayley DHTs

- HyperCup: a Cayley DHT with the degree of *O(log N)*
- Chord: a Cayley DHT with the degree of *O(log N)*



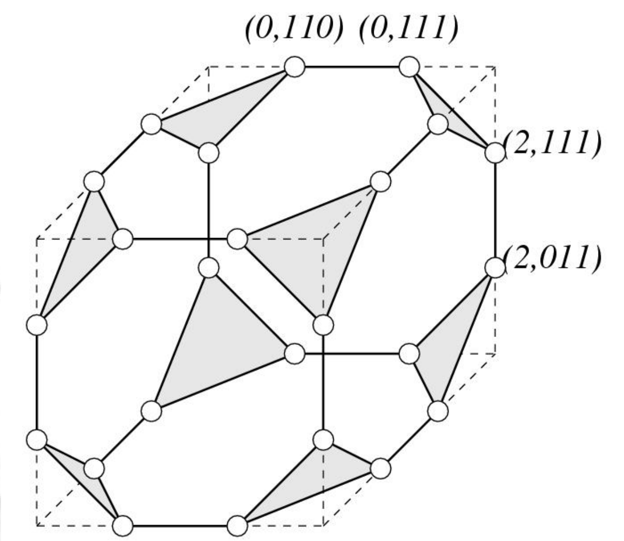
3-dimensional binary hypercube
(generators: 213456,124356,123465)



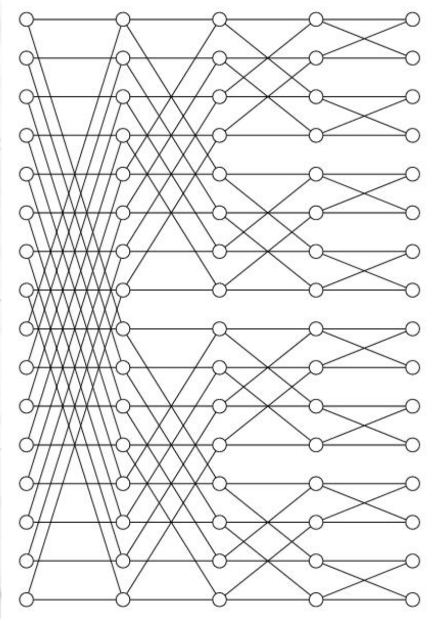
A Chord ring graph with 3-bit key space
(generators: 213456,124356,123465, 214356, 214365)

Constant degree Cayley DHTs

- Cycloid: a Cayley DHT with the degree of *O(1)*
- Viceroy: a Cayley DHT with the degree of *O(1)*
- CAN: a Cayley DHT with the (adjustable) degree of *O(d)*



The cube connected cycle of a 3-dimensional hypercube



The wrapped butterfly B4

Cayley Graph Properties and DHTs

- Every Cayley graph is vertex transitive.

  Significance for Analysing DHTs: explicitly enable an algebraic design approach for the routing algorithm of Cayley DHTs in that the routing between two arbitrary vertices can be reduced to the routing from an arbitrary vertex to a special vertex.

  provide a unified method to evaluate the communication load on each nodes of DHTs. All Cayley DHTs can achieve communication load balancing on each nodes, whereas non-Cayley DHTs cannot.

Cayley Graph Properties and DHTs

- Some Cayley graphs are hierarchical thus can be decomposed into smaller sub-graphs using elementary group theory (recursive decomposition).

  Significance for Analysing DHTs: provide a method to evaluate two important features of DHTs: fault tolerance (network resistance) and proximity (network latency). All hierarchical Cayley DHTs (i.e. HyperCup, Chord) are optimally fault tolerant, and might support easy solutions to ensure proximity of DHTs, whereas non-Cayley DHTs and non-hierarchical Cayley DHTs (i.e. Viceroy, Cycloid, CAN) might not.
- Most of Cayley graphs are optimally fault tolerant except for a very particular family.

  Significance for Analysing DHTs: Almost all Cayley DHTs (i.e. HyperCup, Chord, Cycloid, Viceroy) are “naturally” optimally fault tolerant from pure static DHT topology perspective. It is thus much easier for such DHTs to technically ensure optimal fault tolerance in the dynamic DHT algorithm design for sparsely populated DHT identifier space, or frequent leaving/failing nodes. For “non-naturally” optimally fault tolerant DHTs, it is usually not the case.

Benefits

- Cayley DHTs have some clear advantages over non-Cayley DHTs.

  Static DHT topologies of Cayley DHTs can naturally ensure most of desirable DHT features thus require less overhead and complexity on dynamic DHT algorithms in comparison to non-Cayley DHTs.
- It is safe and advisable to start any DHT design from choosing Cayley graphs as static DHT topologies

  Cayley DHTs can cover both constant degree and non-constant degree DHT design. Most of “good” non-Cayley graphs (i.e. de Bruijn) applied in DHTs are closely related to Cayley graphs, and can also be “Cayley-lized” to transit into “better” Cayley graphs (i.e. Hyper de Bruijn)

see Qu, C., W. Nejdl, M. Kriesell: Cayley DHTs: A Group-Theoretic Framework for Analysing DHTs Based on Cayley Graphs, TR-Apr-04, L3S, and ISPA’04

Overview

Motivation and Background
- L3S Projects & the Semantic Web

Peer-to-Peer Data Management
- Edutella Framework and Infrastructure
- Peer-to-Peer Topologies
- Peer-to-Peer Data Retrieval
- Google2Google

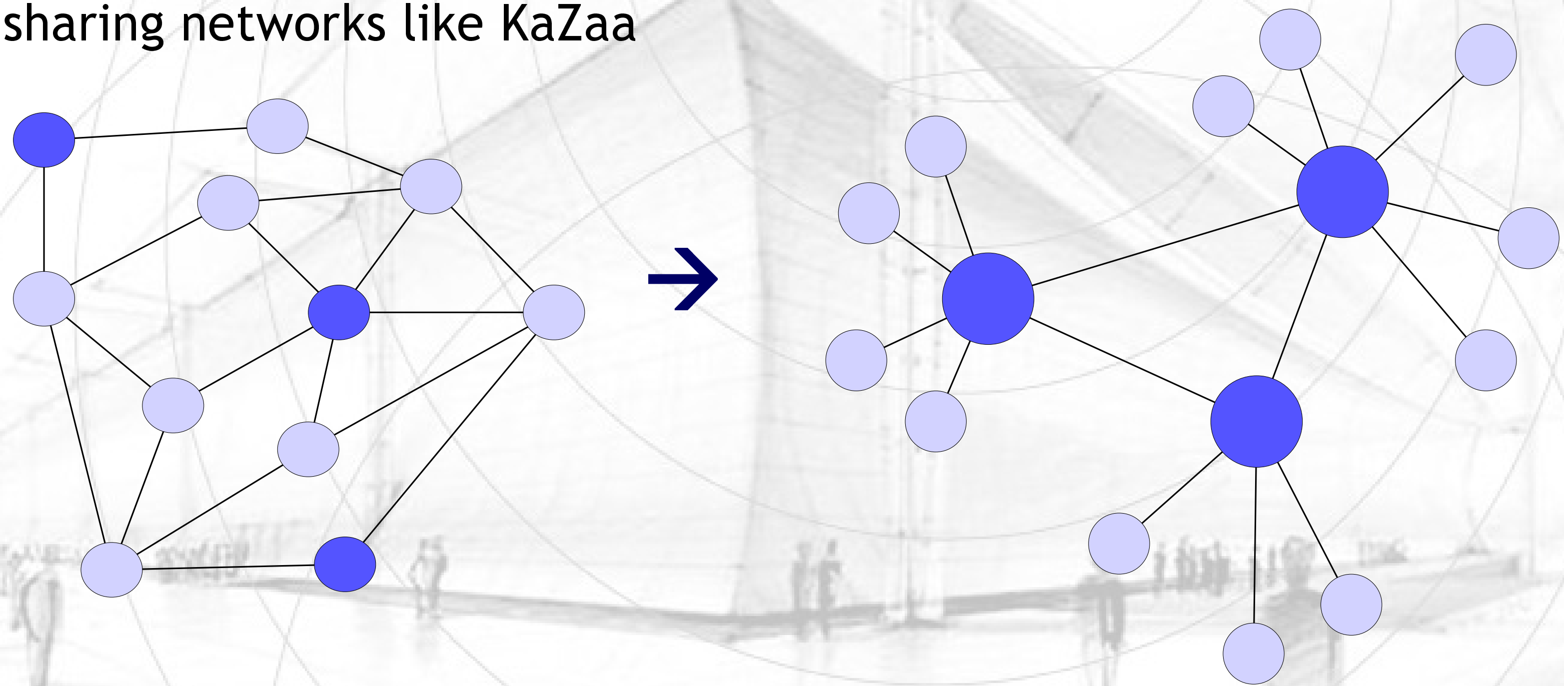
Summary and Conclusions

From Peer-to-Peer to Super-Peer Networks

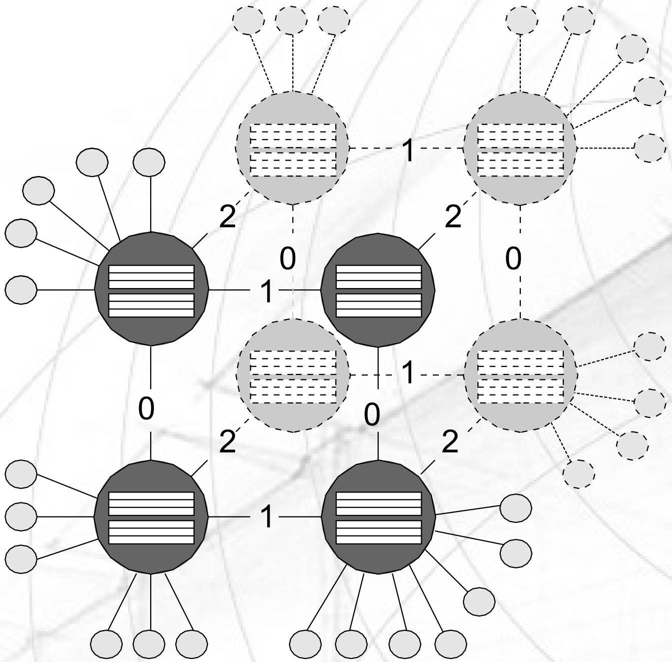
Observation: Peers vary significantly in availability, bandwidth, processing power, etc.

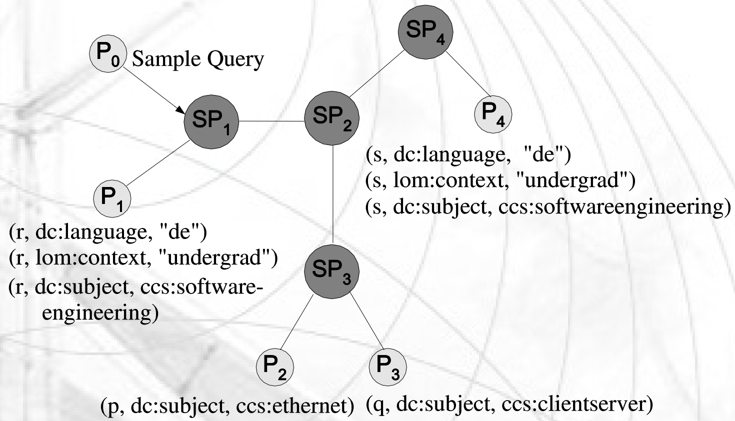
Create network backbone from highly available and powerful peers to distribute load better.

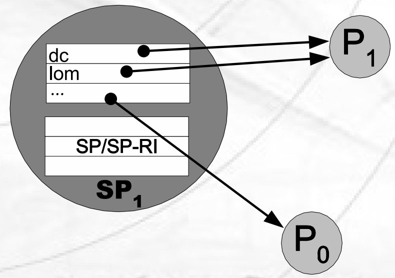
see also Yang, Garcia-Molina: Improving Search in P2P Systems, Intl. Conf. on Distributed Computing Systems, Vienna, 2002, or file sharing networks like KaZaa



Super-Peers and Routing Indices







see Nejdl et al. Super-Peer-Based Routing and Clustering Strategies for RDF-Based Peer-To-Peer Networks. WWW 2003

Top-k Optimization

- Querying often results in too many answers
- Goal: Use (meta)data and ranking methods in the context of P2P-networks to retrieve only the *k* ‚best' resources
- Scenario: To prepare an exam, John searches appropriate exercises in Prolog. Annotated resources can be queried in RDQL (simple attribute/constraint queries):

```
SELECT ?X, ?S, ?T
WHERE (?X, rdf:type, lom-edu:Exercise)
  AND (?X, dc:subject, ?S)
  AND (?X, dc:description, ?T)
  AND (?S == acm_ccs:LogicProgramming)
  AND (?T =~ ".*prolog.*")
```

Top-k Query Extensions

```
SELECT ?X, ?S, ?T
WHERE (?X, rdf:type, lom-edu:Exercise)
  AND (?X, dc:subject, ?S)
  AND (?X, dc:description, ?T)
  AND (?S NEARBY acm_ccs:LogicProgramming)
  AND (?T CONTAINS "prolog")
ORDER BY (0.7*?S+0.3*?T)
STOP AFTER 12
```

1. Introduce ‚weak‘ operators
2. Weight the used ranking methods
3. Limit to top-*k* results

Ranking, Merging and Routing

Assume a super-peer network where queries are routed using the HyperCuP-protocol:

Ranking

- Each peer does local ranking of its resources with respect to the query and return the local top-k to its super-peer

Merging

- At the super-peers all results from the assigned peers are ranked again and merged into one top-k list. This is returned through the super-peer backbone to the querying peers, with merges at all super-peers involved

Ranking, Merging and Routing

Routing

- Super-peer indices store information from which directions the top-k answers were sent for each query. When a ‚known' query comes to a super-peer it passes it to the most promising (super-)peer only with x% probability (e.g. 90%), and with 100-x% probability to other peers

more details in Nejdl, Siberski, Thaden, Balke: Top-k Query Evaluation for Schema-Based Peer-to-Peer Networks, ISWC'04, Thursday, 11:30

Overview

Motivation and Background

- L3S Projects & the Semantic Web

Peer-to-Peer Data Management

- Edutella Framework and Infrastructure
- Super-Peer and Peer-to-Peer Topologies
- Peer-to-Peer Data Retrieval
- Google2Google

Summary and Conclusions

Taking the next step: Google2Google

Potentially better coverage of the Web

- Provision of „Hidden Web“ Content
- Community-driven Search Engines
- Fast Updates of Pages (local knowledge allows for more efficient incremental crawls)

More „democratic“ search infrastructures

- Each web server can crawl and provide its own content or
- Decentralized crawlers can provide a specific view on the Web

To see how far the P2P paradigm can be pushed beyond supporting focussed communities

- „General purpose“ peer-to-peer search on distributed data

Distributed Search Engine Peers

Peers have to provide functionality for

- Crawling, indexing, ranking, query answering
- Grub for example just does distributed crawling (grub.org)

Problem: partition the web graph, possibly with some overlaps to achieve redundancy, without centralized schedulers

- „Parallel Crawlers“, by Cho, Garcia-Molina, WWW 2002
- "Exploiting the block structure of the web for computing PageRank", by Kamvar, Haveliwala, Manning, 2003.

Distributed Ranking Algorithms

Compute ranks in decentralized peer-to-peer network

IR measures: are usually collection dependent (IDF in TF/IDF for example)

- "On the Update of Term Weights in Dynamic Information Retrieval Systems", by Viles and French, CIKM 1995

Computing PageRank like measures:

- "The EigenTrust Algorithm for Reputation Management in P2P Networks", by Kamvar, Schlosser, Garcia-Molina", WWW 2003
- "Knowing Where to Search: Personalized Search Strategies for Peers in P2P Networks", by Chirita, Nejdl, Scurtu, SIGIR-P2P 2004

Distributed Ranking Algorithms II

Computing PageRank variants more suited to decentralized computation:

- "Using SiteRank for Decentralized Computation of Web Document Ranking", by Wu/Aberer, AH 2004
- "Computing PageRank in a Distributed Internet Search System", by Wang/DeWitt, VLDB 2004
- Potential problem: web sites at mass hosters (“the Geocities effect”)

Forschungszentrum L3S

Top-k Retrieval and Optimization

Described in

- „Top-k Query Evaluation for Schema-Based Peer-to-Peer Networks“, by Nejdl, Siberski, Thaden, Balke, ISWC 2004
- "Progressive Distributed Top-k Retrieval in Peer-to-Peer Networks", by Balke, Nejdl, Siberski, Thaden, ICDE 2005

Taking Trust and Malicious Peers into Account

Spamming is easier in a decentralized infrastructure

- Bias towards trusted peers: "Combating Web Spam with TrustRank", by Gyöngyi, Garcia-Molina, Pedersen. VLDB 2004
- How susceptible are distributed algorithms to malicious peers / analyze attack scenarios and counter measures
- "The EigenTrust Algorithm for Reputation Management in P2P Networks", by Kamvar, Schlosser, Garcia-Molina, WWW 2003
- "Knowing Where to Search: Personalized Search Strategies for Peers in P2P Networks", by Chirita, Nejdl, Scurtu, SIGIR-P2P 2004

Plus some more …

Query driven caching and replication are more appropriate for P2P networks

How do I cope with disappearing / reconnecting search peers? (both for search and rank computations)

Simple things get surprisingly difficult, e.g. what changes when I have to work with distributed query statistics? (relevant for optimization but also for basic functionalities like „Did you mean?“)

Summary and Conclusions

Distributed and decentralized information infrastructures are a core ingredient of the Semantic Web

Peer-to-peer data management extends traditional P2P networks and distributed / heterogeneous database research, with a lot of new challenges as well as additional functionalities

Building blocks are query and reasoning languages (for information systems, information retrieval, publish/subscribe), query optimization and caching, distributed ranking and search algorithms, efficient network topologies and routing algorithms, and decentralized access control and trust negotiation infrastructures

There is much to build upon, but even more to work on (and become famous for ☺), so see you at the next Google2Google related session at ISWC'05 or any other conference!