Distributed Query Processing in P2P Systems with incomplete schema information

Marcel Karnstedt

Technical University of Ilmenau

marcel.karnstedt@tu-ilmenau.de

Outline

- Motivation, background
- Query model
- Routing approach
- Simulation & preliminary results
- Conclusion
- Current work

Introduction

- Data integration approach using P2P
- New views on data integration techniques
  - No global schema used, each peer provides a local schema
  - Easy to include new peers
- P2P advantages: decentralization, scalability, robustness, ...
- P2P disadvantages: difficult to process queries efficient, incomplete results, ...

Main Challenges

- No central instance at all
- No global knowledge and schema
  - A peer knows only about itself and its neighbours
- Processing of queries that are beyond local schema and correspondences
  - incomplete schema information, unknown elements, incomplete results
- Performance of query evaluation
  - Which of the neighbour peers can provide data at all, which is most suitable for answering?

Contribution

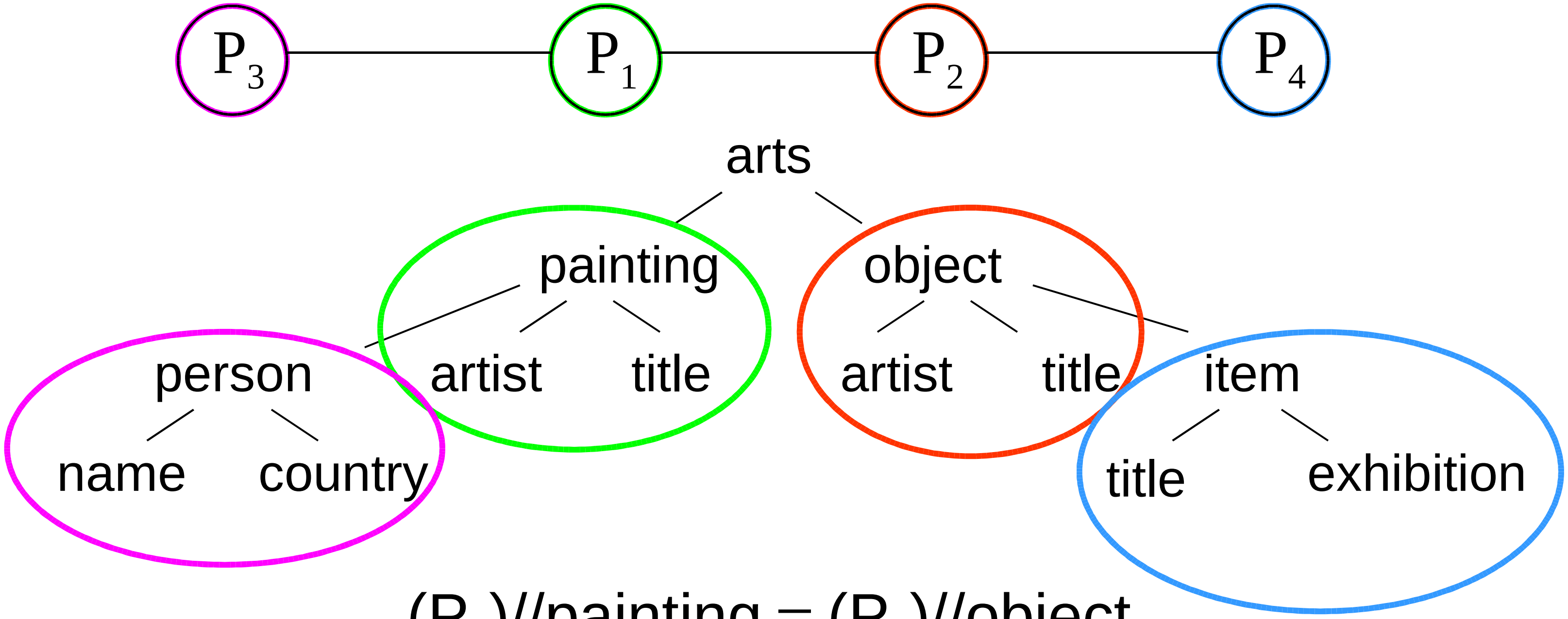
- Data integration approach using schema-based P2P systems
- Efficient processing & routing of queries in such systems
  - Queries referencing unknown schema elements
  - Routing approach based on routing indexes
  - Evaluation
- Open issues of distributed query processor and query engine for P2P systems

Data Model

- All peers provide XML data, described by schemes (DTD, XML-Schema)
  - Maybe use wrappers
- Two issues:
  - (1) formulate schema correspondences
  - (2) formulate queries without complete schema information

- Ad (1): Three correspondence operations:
  - Equivalence (horizontal), child-of/part-of (vertical), transformation

Example Scenario



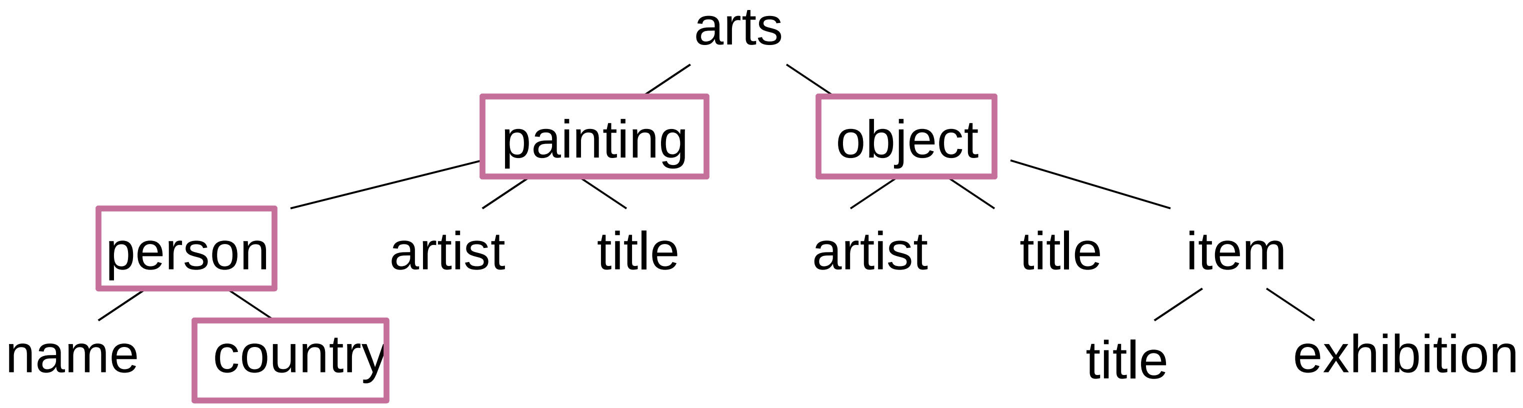
$(P_1)//\text{painting} \equiv (P_2)//\text{object}$

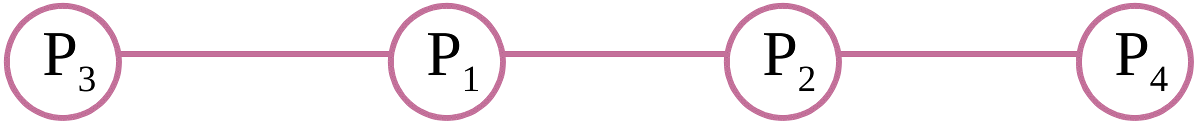
$(P_1)//\text{painting} <_{\text{artist=name}} (P_3)//\text{person}$

$(P_2)//\text{object} <_{\text{title=title}} (P_4)//\text{item}$

Example Query

$$\sigma_{[object]person/country=\text{‘}Netherlands\text{‘}}(\mu_{[\perp]//object}(P_2.xml))$$





Distributed Processing

- Data shipping vs. query shipping
  - Classical strategies in distributed systems
  - Data shipping bad choice for investigated systems
  - Query shipping reduces the transferred data volume by far
- Also hybrid shipping techniques
- Special query shipping approaches:
  - Query decomposition
  - “Mutant Query Plans“ – MQP

P2P Query Processing

- In P2P systems these classical strategies must be adapted!
- Query decomposition
  - Each peer processes part(s) of the query
- Query transformation
  - Use correspondences for rewriting
- Query routing
  - Next slides...

Query Routing

- Problem: incomplete information about data placement and schemes - Which of the known peers is (most) suitable for answering queries?

- Distributed Indexes (e.g. DHTs), ...

- Routing indexes:
  - *Compound Routing Indexes, Hop Count Routing Indexes*
  - Schema level: identifiers
  - Instance level: predicates

Routing Indexes

Routing index at peer P_2

Neighbour-Peer	Category Schema level	Category Instance level	Cardi-nality	#Peers
1	painting	-	520	4
	painting/title	-	520	3
	...	...	...	...
	painting/person	-	210	2
	painting/person/name	-	210	2
	painting/person/birth	[@date<'1800']	132	1
	painting/person/birth	[@date>='1800']	78	1
4	item	-	112	5
...	...	...	...	...

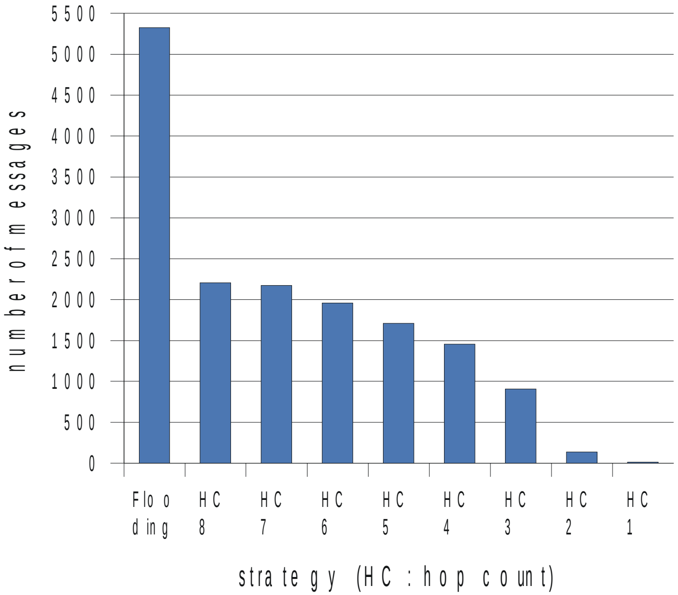
Indicated how unknown schema elements are integrated

Simulation

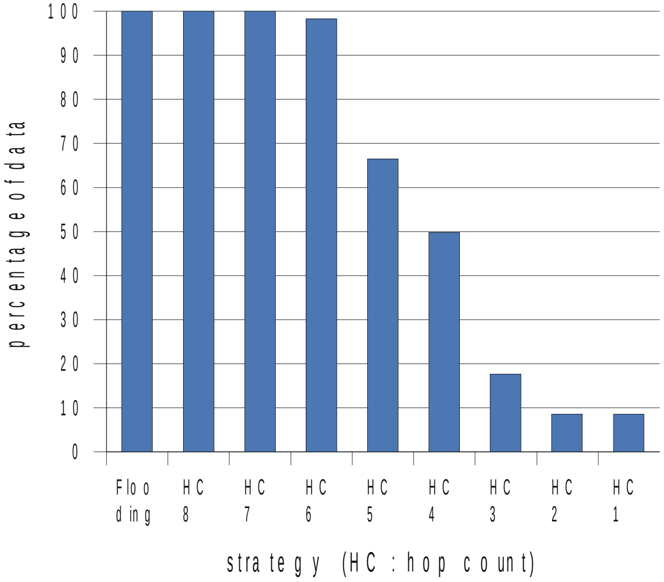
- Main goals:
  - Impact of defined horizon
  - Performance of our approach
  - Percentage of data actually retrieved
- Environment:
  - 40 peers, randomly chosen bidirictional correspondences
  - data distributed horizontally and vertically
  - 64 queries
  - Experiments on data about plays of Shakespeare

Preliminary Results

Query shipping, hop count 1...8, flooding:



Percentage of data retrieved:



Learned: only mappings is bad choice, percentage satisfying at hop count of 4-5, ...

Conclusion

Routing indexes are a powerful tool when used for distributed query processing

- Schema & instance level

Problem: optimal hop count

- We are able to process queries efficient, even if only limited knowledge is available!
- Small modifications in the horizon lead to significant changes in the results
- Small horizon may already be satisfying
- Unsatisfying percentage if hop count too low

Outlook

- (Semi-)Automatic rule-based schema matching
- Efficient techniques for index building and maintenance
- Extended simulation environment
- Improve query engine, e.g. stateless queries
- Dynamic cost model & cost-based decisions
- Adaptive query processing techniques
- Scenarios: virtual observatory, crisis management

Thank you for your attention!

TECHNISCHE
UNIVERSITÄT
ILMENAU