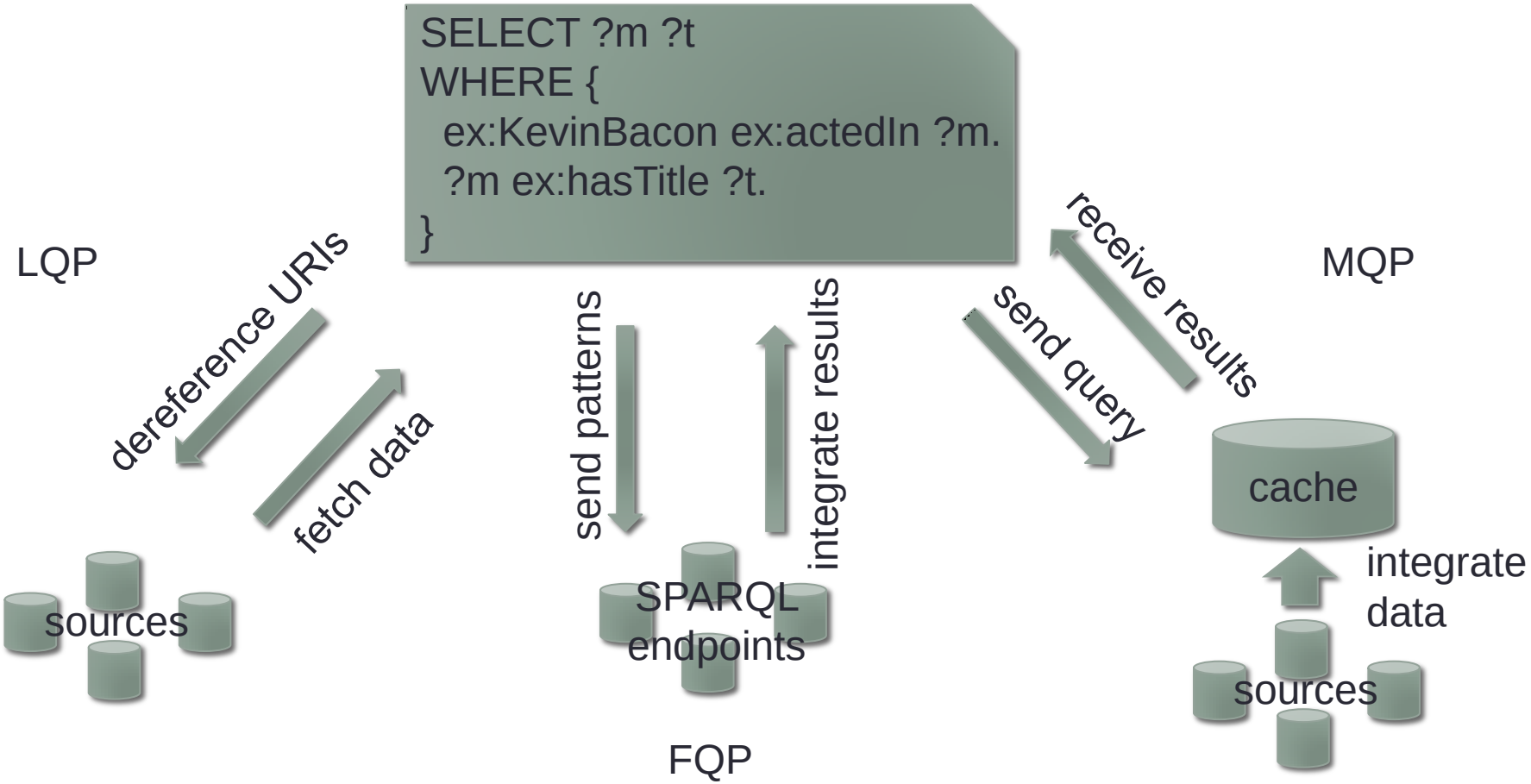
LEARNING FROM THE HISTORY OF DISTRIBUTED QUERY PROCESSING

A Heretic View on Linked Data Management

Heiko Betz[1], **Francis Gropengießer**[1], Katja Hose[2], Kai-Uwe Sattler[1]
[1]Ilmenau University of Technology, Germany
[2]Aalborg University, Denmark

Motivation



Many LQP and FQP approaches rely to some extent on or resemble techniques developed in the context of distributed and federated database systems!

Agenda

What can we learn from distributed and federated database systems?

Discuss myths about Linked Data and MQP

Sketch directions for future research

DISTRIBUTED AND FEDERATED DATABASES

A Retrospective

Systems

- SDD–1:
  - Bernstein, P.A., Goodman, N., Wong, E., Reeve, C.L., Rothnie, J.B.: Query Processing in a System for Distributed Databases (SDD-1). ACM Trans. Database Syst. 6 (1981) 602–625
- INGRES:
  - Stonebraker, M.: The Design and Implementation of Distributed Ingres. In: The INGRES Papers. (1986) 187–196
- R*:
  - Lohman, G.M., Mohan, C., Haas, L.M., Daniels, D., Lindsay, B.G., Selinger, P.G., Wilms, P.F.: Query Processing in R*. In: Query Processing in Database Systems. Springer (1985) 31–47

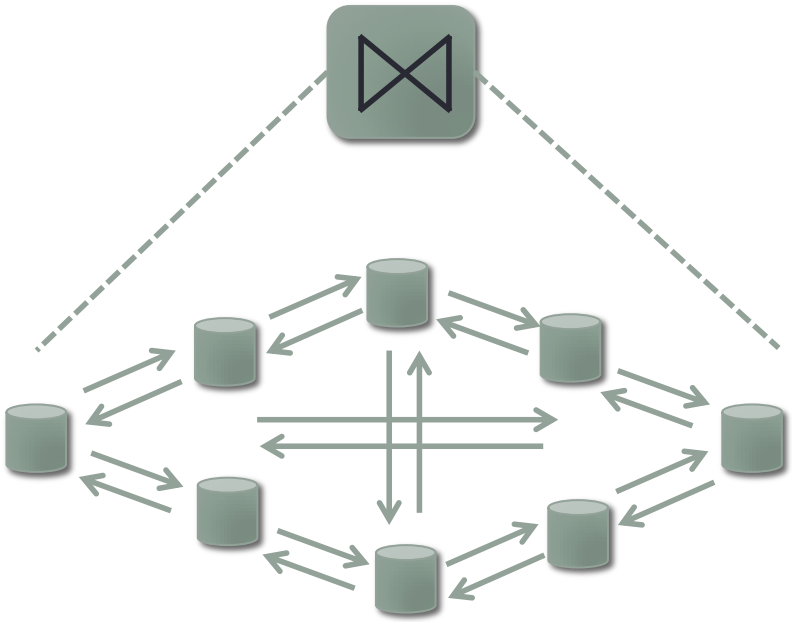
Techniques

- Reducing communication costs during join processing:
  - Hyunchul Kang, Nick Roussopoulos: Using 2-way Semijoins in Distributed Query Processing, ICDE (1987), pages 644-651
  - Judy C. R. Tseng, Arbee L. P. Chen: Improving Distributed Query Processing by Hash-Semijoins, In: J. Inf. Sci. Eng. 8 (1992), pages 525-540
- Dealing with bursty or delayed data arrival:
  - Tolga Urhan, Michael J. Franklin: XJoin: A Reactively-Scheduled Pipelined Join Operator, In: IEEE Data Eng. Bull. 23 (2000), pages 27-33
- Dealing with data, schema, and system heterogeneities:
  - M. T. Roth and P. M. Schwarz. Don't Scrap It, Wrap It! A Wrapper Architecture for Legacy Data Sources. In VLDB '97, pages 266–275

Still Unsolved Problems

Scalability

Distributed query optimization
(cost estimation)

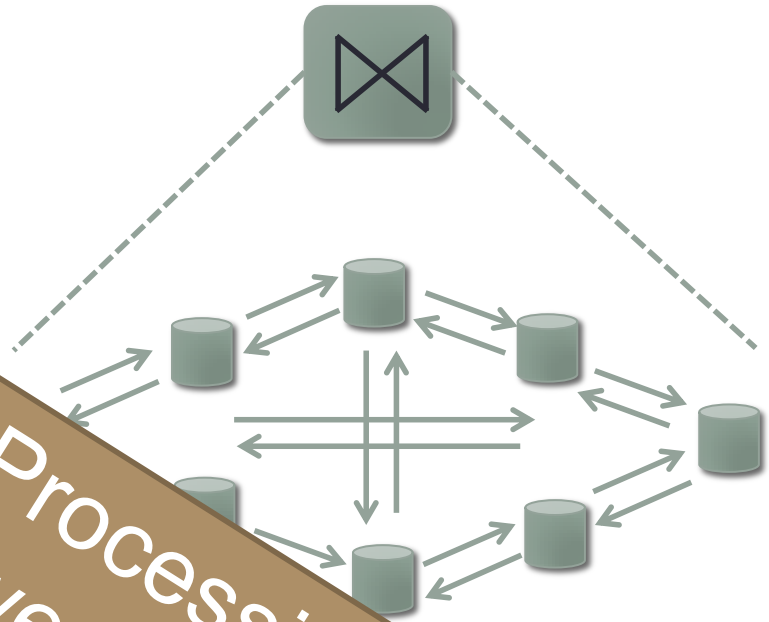


FQP in Linked Data scenarios inherits the challenges and problems of federated database systems

Even more complex, due to loosely structured RDF format and very different SPARQL endpoint implementations

Still Unsolved Problems

Materialized Query Processing could be an answer?!



Distributed query
(cost estimati

FQP in Linked Data scenarios inher and
problems of federated database

Even more complex, due to loosely structured R mat
and very different SPARQL endpoint implementations

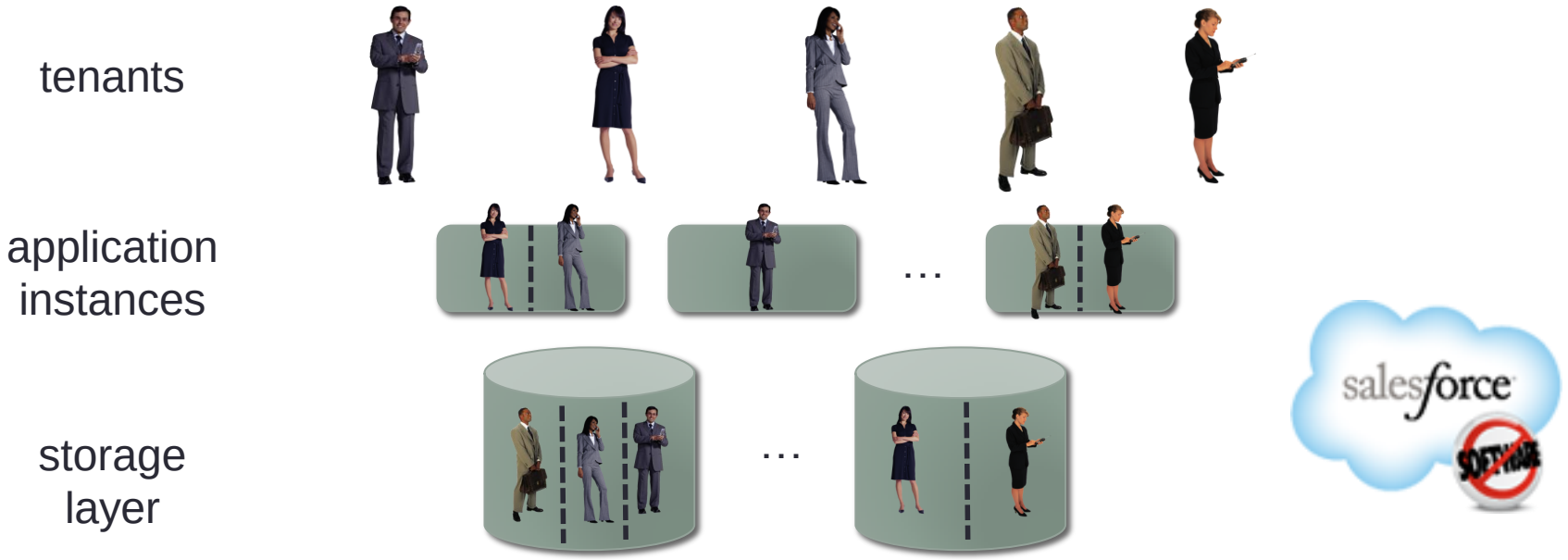
THESES

1. The Volume of Linked Data Is Too Big for Centralized Management

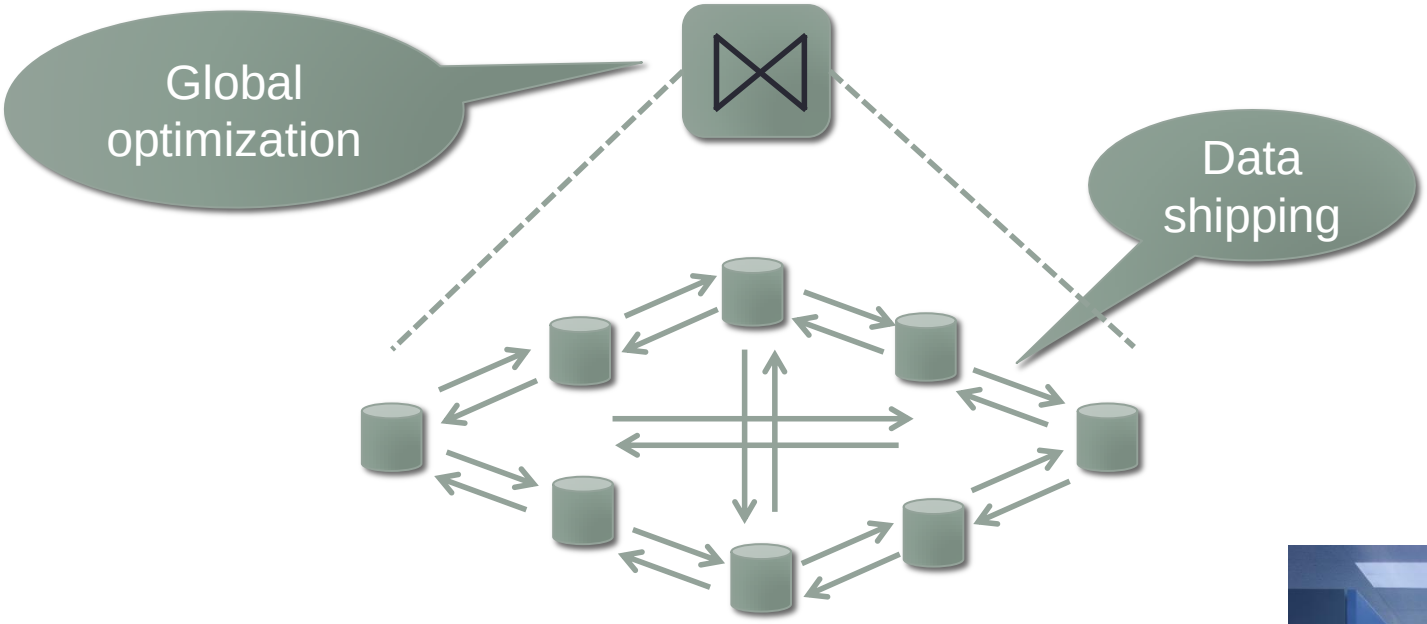
- Linked Open Data Volume is continuously increasing
- However: encoding approaches can reduce data volume tremendously
- Example - English DBpedia :
  - 50 GB uncompressed (386,546,905 triples)
  - Much redundancy: only 23,645,703 subjects, 51,583 predicates, and 75,867,838 objects are unique
  - With dictionary encoding: approximately 10 MB (4 MB for the dictionary, 6 MB for triple data)
- Currently: whole LOD cloud would require only 800 GB of storage space

2. Materialization in Centralized Repositories Violates Data Authority

- Multi-tenancy techniques in order to control access
  - Stefan Aulbach, Michael Seibold, Dean Jacobs, Alfons Kemper: Extensibility and Data Sharing in Evolving Multi-Tenant Databases, In: International Conference on Data Engineering (2011), pages 99-110
- Changing the license model
  - Creative Commons license



3. Scalability Can Be Achieved Only by Distributed Query Processing



amazon web services™

Cluster-based solutions (Infrastructure as a Service)?!

4. Linked Data Processing Is only about SPARQL Processing

- SPARQL processing significant fraction
- New Trends:
  - Data analytics and reasoning
  - Stream processing
- Need for spatial, temporal, and stream processing extensions
  - D. Le-Phuoc, M. Dao-Tran, J. X. Parreira, and M. Hauswirth. A native and adaptive approach for unified processing of linked streams and linked data. In ISWC'11, pages 370–388, 2011
  - stSPARQL (STRABON system)
  - …

scalability

standardization

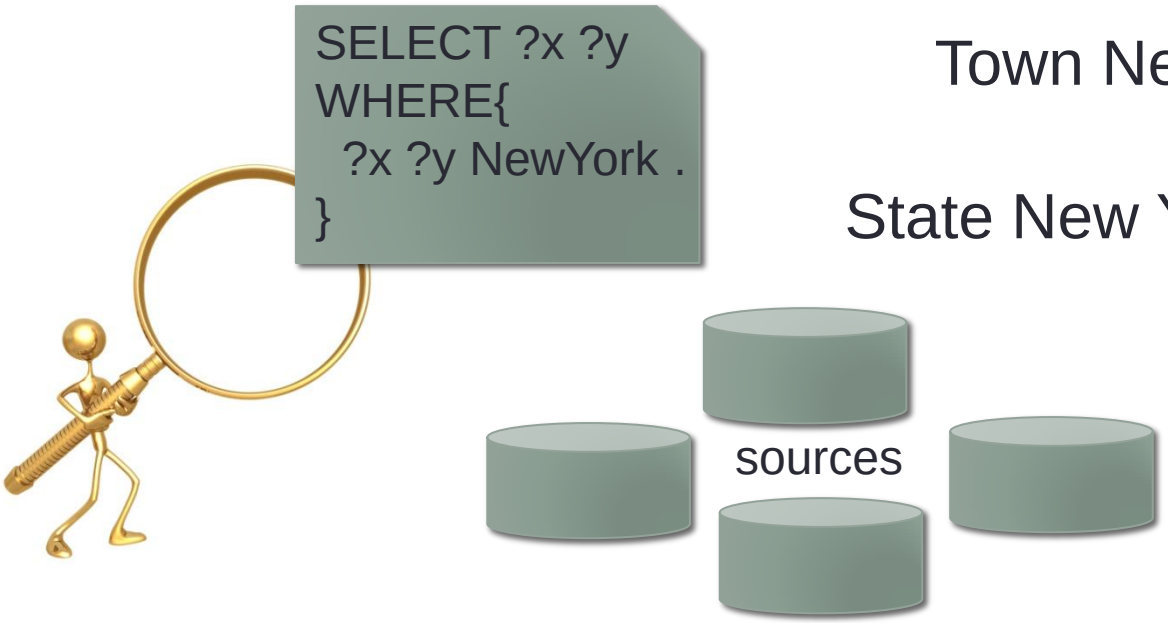
5. The Problem of Semantic Heterogeneity Can Be Solved by Using Ontologies

- Different sources provide different ontologies
- Toni Gruetze, Christoph Boehm, Felix Naumann: Holistic and Scalable Ontology Alignment for Linked Open Data, In: Linked Data on the Web (2012):
  - 190 out of 295 sources of the LOD cloud use proprietary vocabulary terms
  - Only 15 out of 190 offer mappings to other widely deployed vocabularies
- Ontologies over ontologies are required
- Hence, ontologies do not completely solve the problem of semantic heterogeneity

MQP + Data Integration

6. HTTP URIs Can Be Used to Identify Relevant Endpoints

- Only subjects and predicates are required to be dereferenceable URIs
- Looking up non-URI objects (literals) leads to the problem of source selection
- Additional meta information is needed, e.g., indexes, or centralized caching



7. The Freshness of Data Is Guaranteed only with Distributed Query Processing

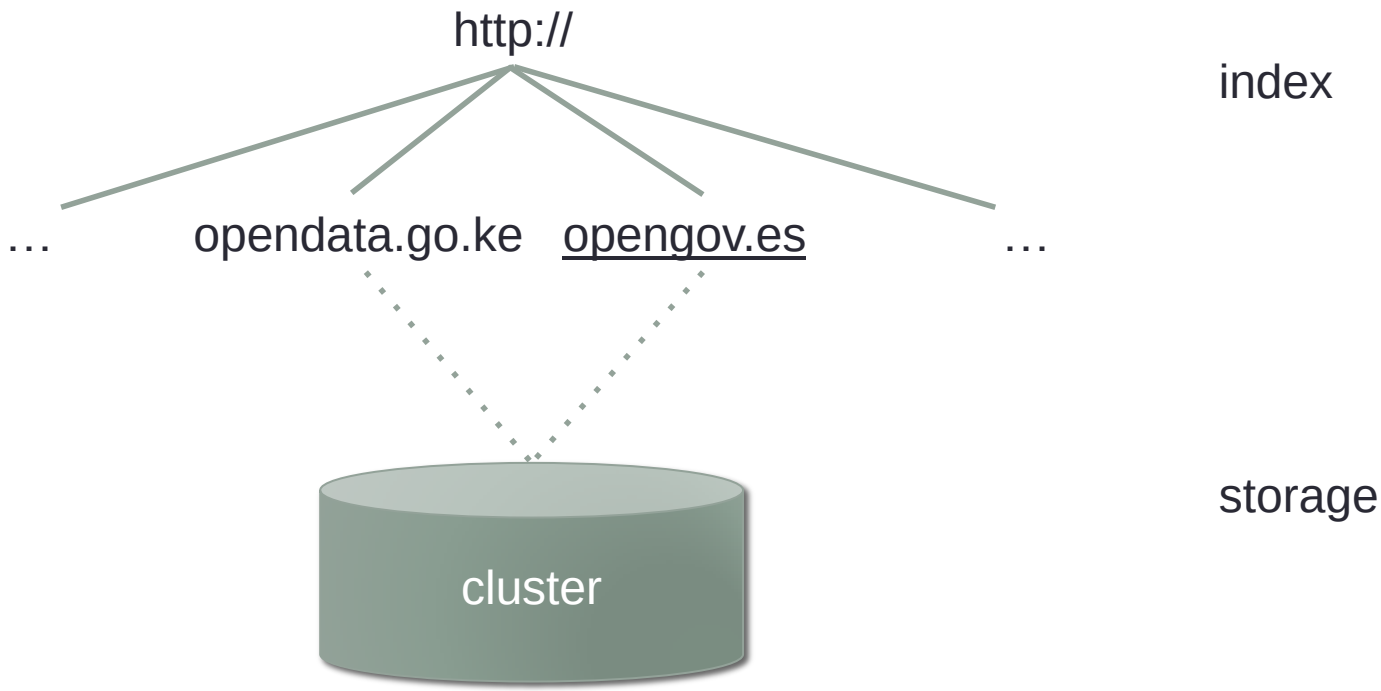
- Clearly:
  - Freshest data by querying sources directly
- However:
  - Many source are only updated rarely
- Examples:
  - US Census – not updated since creation
  - Linked Sensor Data – last update 2008
  - Source Code Ecosystem Linked Data – once per year
- Techniques for update retrieval:
  - E. Oren, R. Delbru, M. Catasta, R. Cyganiak, H. Stenzhorn, and G. Tummarello. Sindice.com: A Document-Oriented Lookup Index for Open Linked Data. Int. J. of Metadata and Semantics and Ontologies, 3:37–52, 2008
  - Incremental updates, e.g., provided by DBpedia

8. Open Data Is Accessible as Linked Data

- Many open data platforms do not meet Tim Berners-Lees rules
- Open Data Survey (http://wwwdb.inf.tu-dresden.de/opendatasurvey/):
  - Very different file formats
  - Different APIs
  - Different schemas
  - Only 8% act as SPARQL/SQL endpoints
- Data integration techniques necessary

9. Centralized Linked Data Is not Linked Data anymore

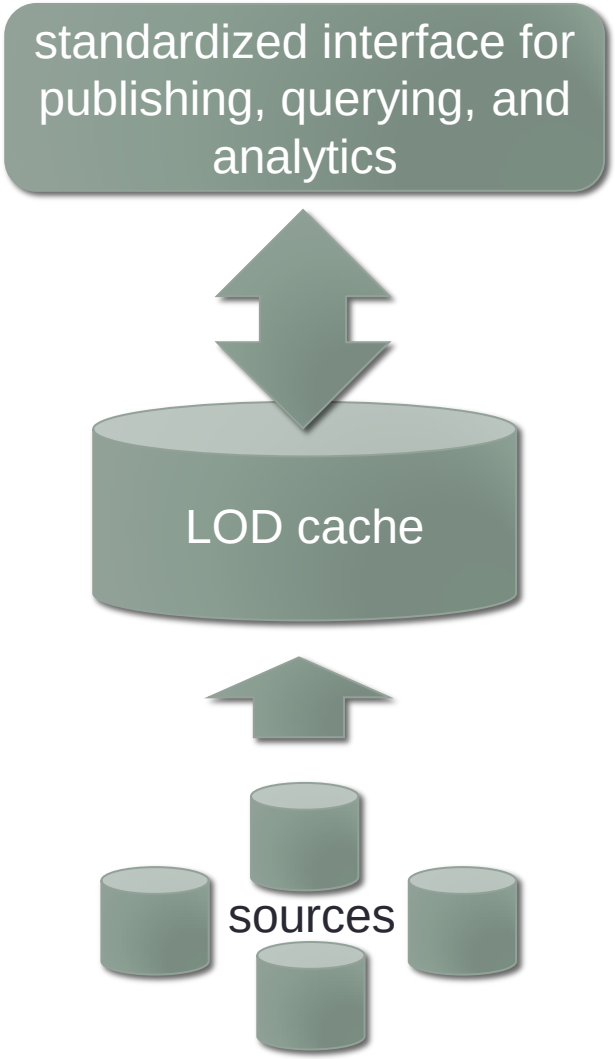
- Using URIs not necessarily implies data distribution
- URIs can act as index keys instead of real physical location descriptions



RESEARCH AGENDA

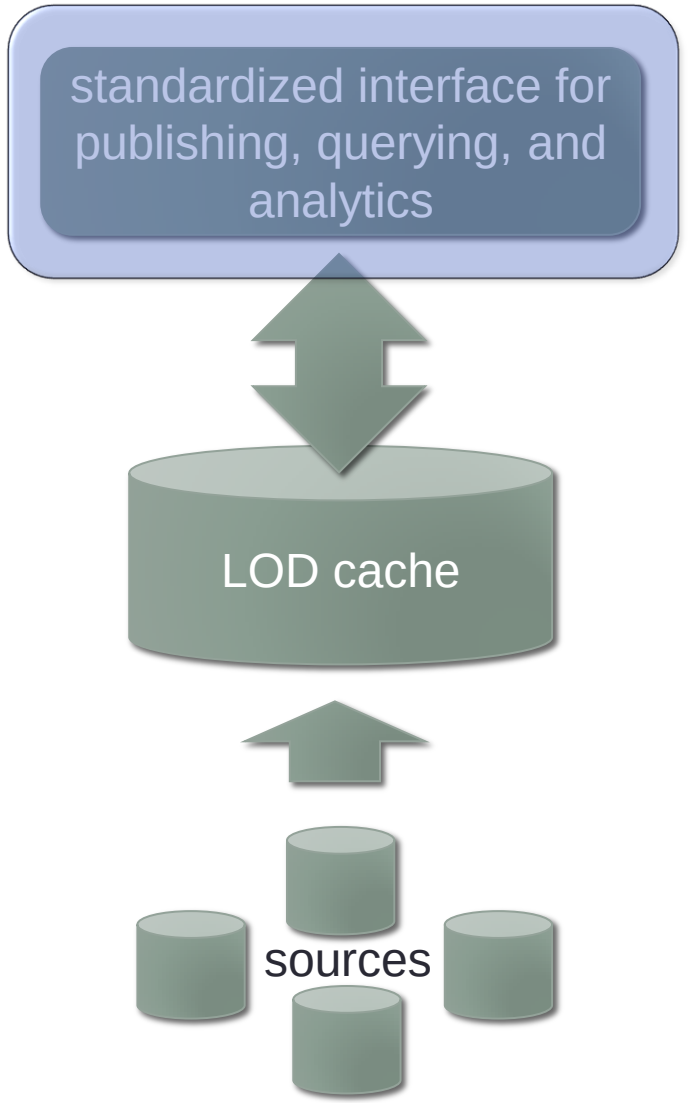
Linked Data as a Service

- Centralized storage for commonly used data (data centers)
- Not applicable for all sources but could be an alternative for a large number
- Pioneers: Flickr, YouTube
- Examples: Windows Azure Marketplace Data-Market, InfoChimps, Factual



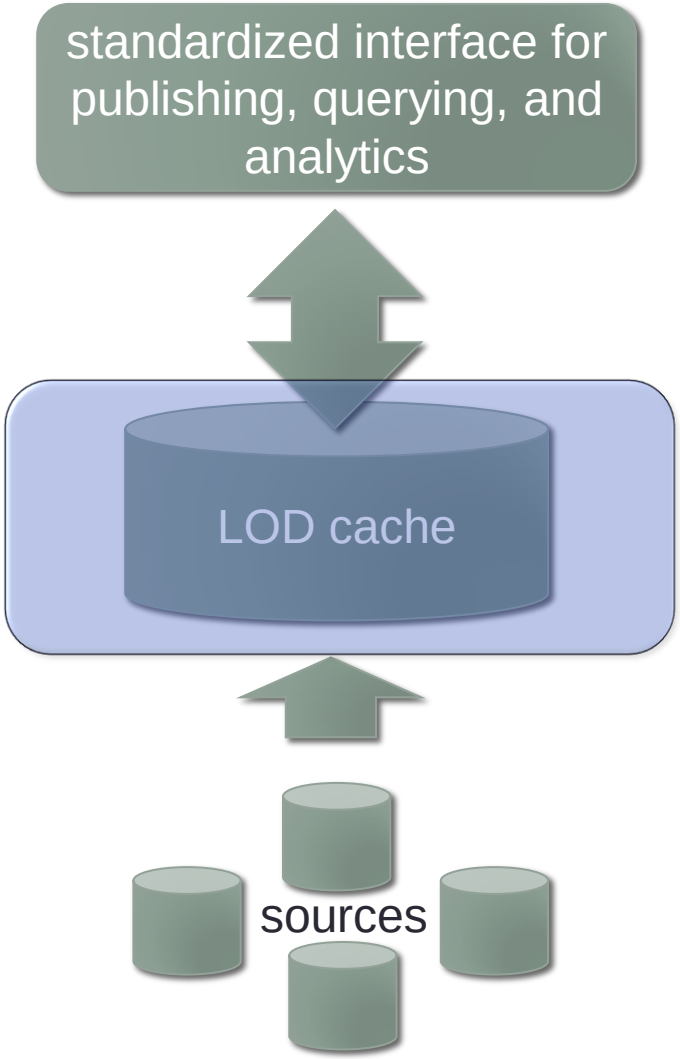
1. Linked Data Processing beyond SPARQL Processing

- Developments show the need for data analytic & stream processing functionality
- Currently, no standardization (SPARQL 1.1 still a draft)
- Integration of necessary techniques in LDaaS platform + standardized access language

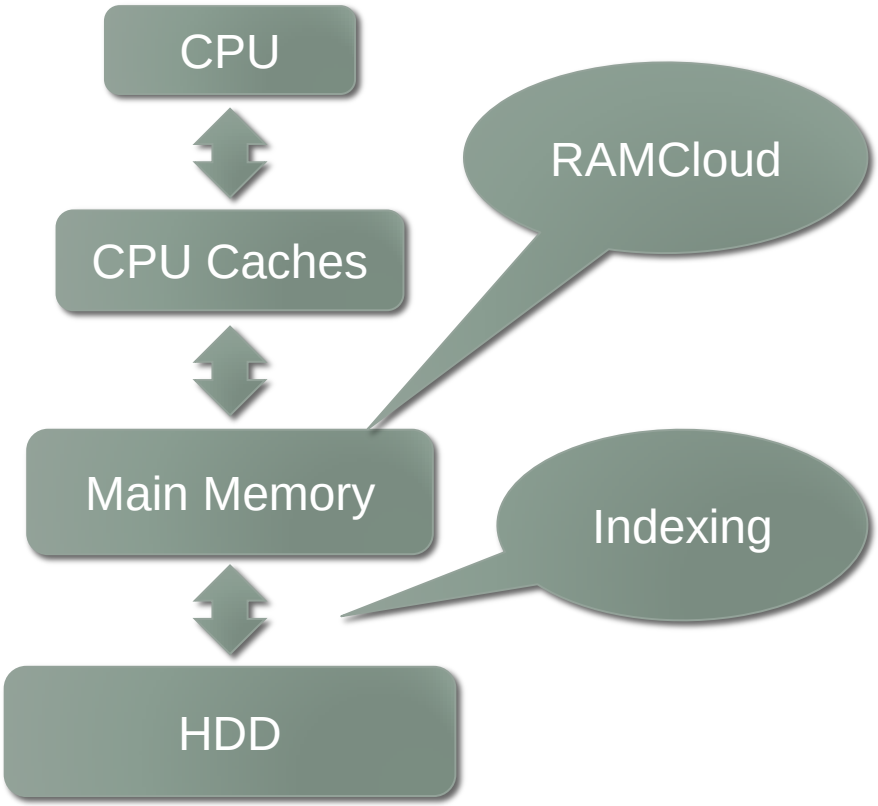


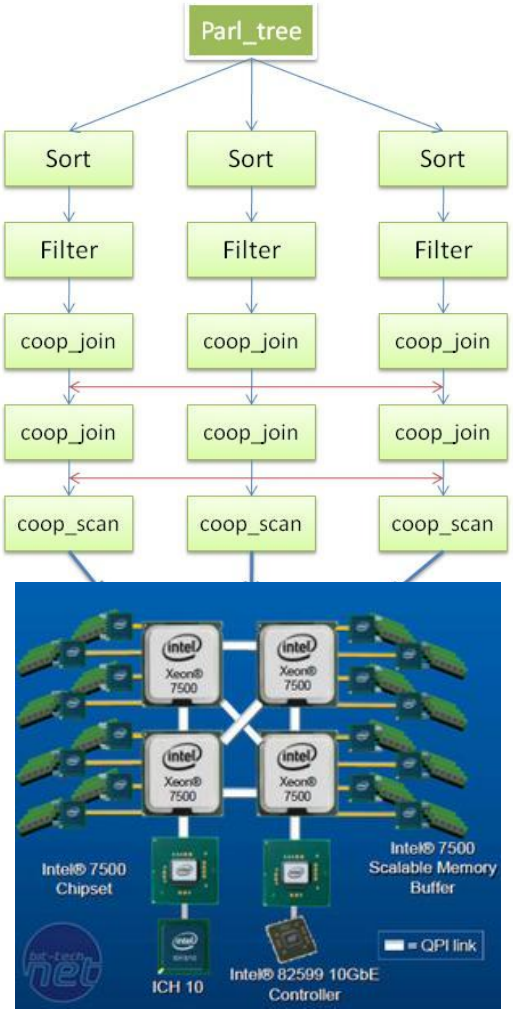
2. Exploit Newly Available Infrastructure/Platform as a Service

- Cloud solutions (Amazon EC2, S3) instead of unreliable P2P environments
- Infinite resources, pay per use, SLAs
- Virtualization + high capacity networks reduces communication costs
- Built in monitoring eases failure recovery (Amazon CloudWatch)
- Automatic scaling and load balancing (Google AppEngine)
- MapReduce support (Amazon Elastic MapReduce)



3. Address the Opportunities of Modern Hardware Architectures for Query Processing





4. Realistic Benchmarking and Metrics

- Benchmark has to model typical scenarios
- Example:
  - Berlin SPARQL models an e-commerce scenario
  - This is not a typical Linked Data application!
- Appropriate metrics are needed:
  - Execution time
  - Throughput
  - Price/performance metrics
- FedBench first step
- However, aspects like aggregation, reasoning, streams have to be considered

5. Simplify Publishing and Exploit Crowdsourcing

- Data publishing often requires conversion to RDF as well as data cleaning
  - Time consuming
  - Human interaction required
- Crowdsourcing could ease publishing
- First examples:
  - Pedantic Web Group (http://pedantic-web.org/)
  - Metzger,S.,Hose,K.,Schenkel,R.: Colledge - A Vision of Collaborative Knowledge Networks. In: 2nd International Workshop on Semantic Search over the Web (SSW) in conjunction with VLDB 2012.
- Still a long way until crowdsourcing can be used easily and efficiently

CONCLUSION

Conclusion

- Distributed Linked Data processing faces similar problems already known from federated database systems
- MQP avoids many of these problems
- MQP becomes attractive due to recent developments in Cloud computing
- We propose Linked Data as a Service
- New research challenges

Federated Query Processing is a good solution for some scenarios... but not for all!
As our discussion of the myths shows other solutions (such as the one we propose) are not as unfeasible as these myths might suggest!

CLOSING WORDS

The next few years will show to what extent re-searchers learned from the history of distributed and federated query processing and which parts of the research agenda were the most challenging ones!