SP²Bench: A SPARQL Performance Benchmark

M. Schmidt*, T. Hornung*, G. Lausen*, and C. Pinkel#

* Freiburg University Database Group, Germany
# MTC Infomedia OHG, Germany

25th International Conference on Data Engineering, 03/30/2009

Motivation

- New technologies: RDF and SPARQL
  - RDF: Data format to encode information in a machine-readable way
  - SPARQL: Query language for RDF
- SPARQL query evaluation is a non-trivial task
  - Same expressiveness as Relational Algebra
  - Several optimization schemes proposed so far
  - No comprehensive benchmark for the SPARQL query language existing

Motivation

- SP²Bench **SP**ARQL **P**erformance **Bench**mark
  - Language-specific benchmark, designed to test engines in a comprehensive way
  - Data generator + real-world benchmark queries
  - Queries pose various challenges to SPARQL engines
  - Allows to compare optimization approaches and to assess strength and weaknesses of implementations

Outline

I. RDF and SPARQL

II. Data Generation

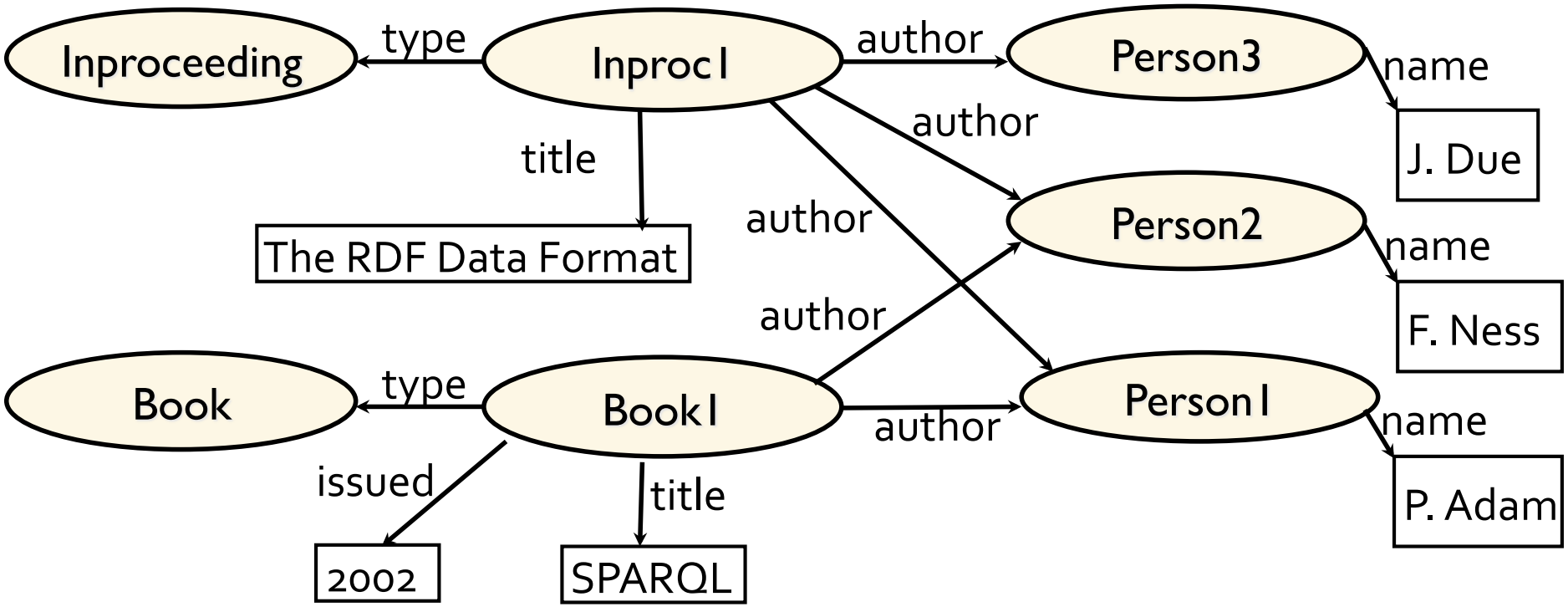
III. Benchmark Queries

IV. Experimental Results

V. Conclusion

The RDF Data Format

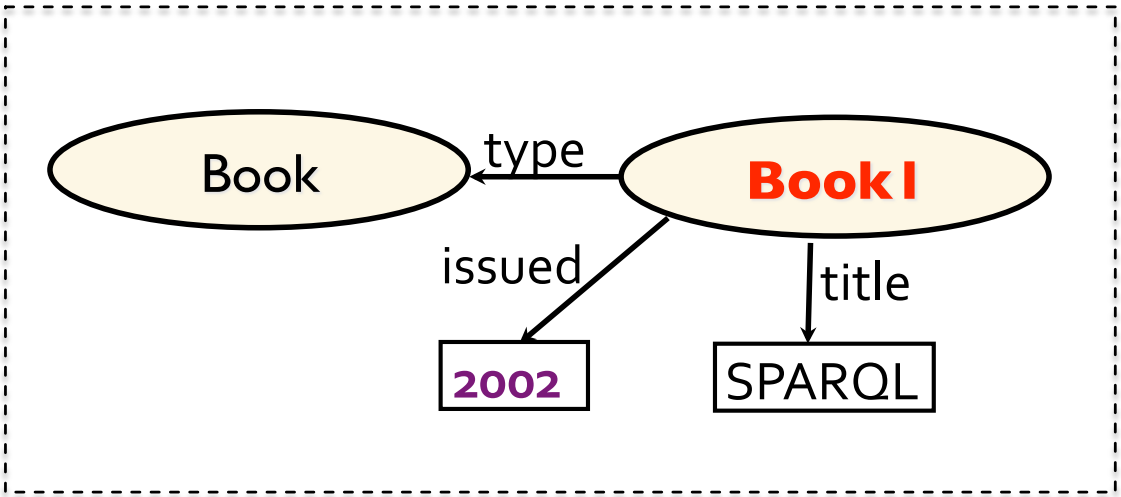
- RDF databases are directed labeled graphs
- Basic constituents: "Triples of Knowledge"



The SPARQL Query Language

- Declarative language for RDF
- Pattern matching against input graph
- Operators: AND, OPTIONAL, UNION, FILTER
- Solution modifiers: ORDER BY, LIMIT, DISTINCT, OFFSET

```
SELECT ?yr
WHERE {
 ?book type Book.
 ?book title "SPARQL"
 OPTIONAL {
  ?book issued ?yr
 }
}
```



Benchmark Scenario

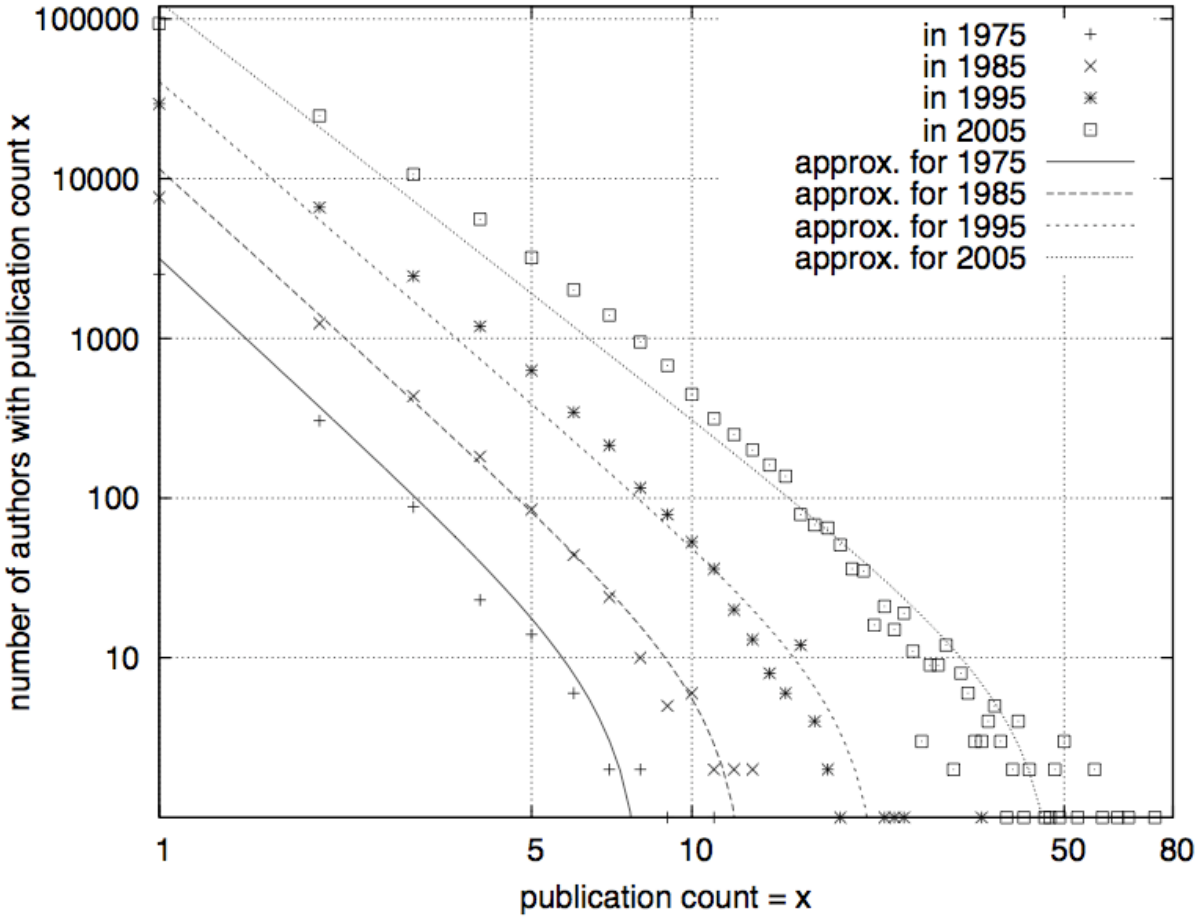
- DBLP Bibliographic Library
- Well-suited for several reasons
  - Meta-data fits the philosophy of RDF data format
  - Scenario well-known, which makes data and queries understandable
  - Mirrors many interesting social-world distributions
- Data generator that allows to create documents of arbitrary size

DBLP Characteristics

- Data generator relies on a study of DBLP
  - Different types of entities (e.g. Articles, Proceedings, Inproceedings, ...)
  - Structure of these entities
  - Quantity of entities (development over time)
  - Citation system
  - Persons (authorship, coauthorship, editors)
- Approximation of real-world distributions by natural function families

DBLP Characteristics

- Distribution of publications: power law



Function prototype:

$$f_{powerlaw}(x) = ax^k + b$$

with

a $\in R_{>0}$: affects x-axis intercerpt
b $\in R$: shift in y-direction
k $\in R_{<0}$: gradient

Observation:
Publication count of leading authors increases over time
-> parameters a and k modeled as functions over time

Data Generator Implementation

- Bases upon extracted approximation functions
- Deterministic (random functions with fixed seed)
- Platform-independent
- Scales linearly to size of generated documents
- Gets by with constant main memory consumption

Query Design

- Real-world requests on top of DBLP data
- Queries vary in a broad range of characteristics
  - Operator constellation and complexity
  - Solution modifiers
  - RDF access patterns
  - Result size (constant, linear, superlinear)
  - Number of variables
  - Possible optimizations
  - …

Example Query

Return the names of all persons that occur as author of at least one inproceeding and at least one article document

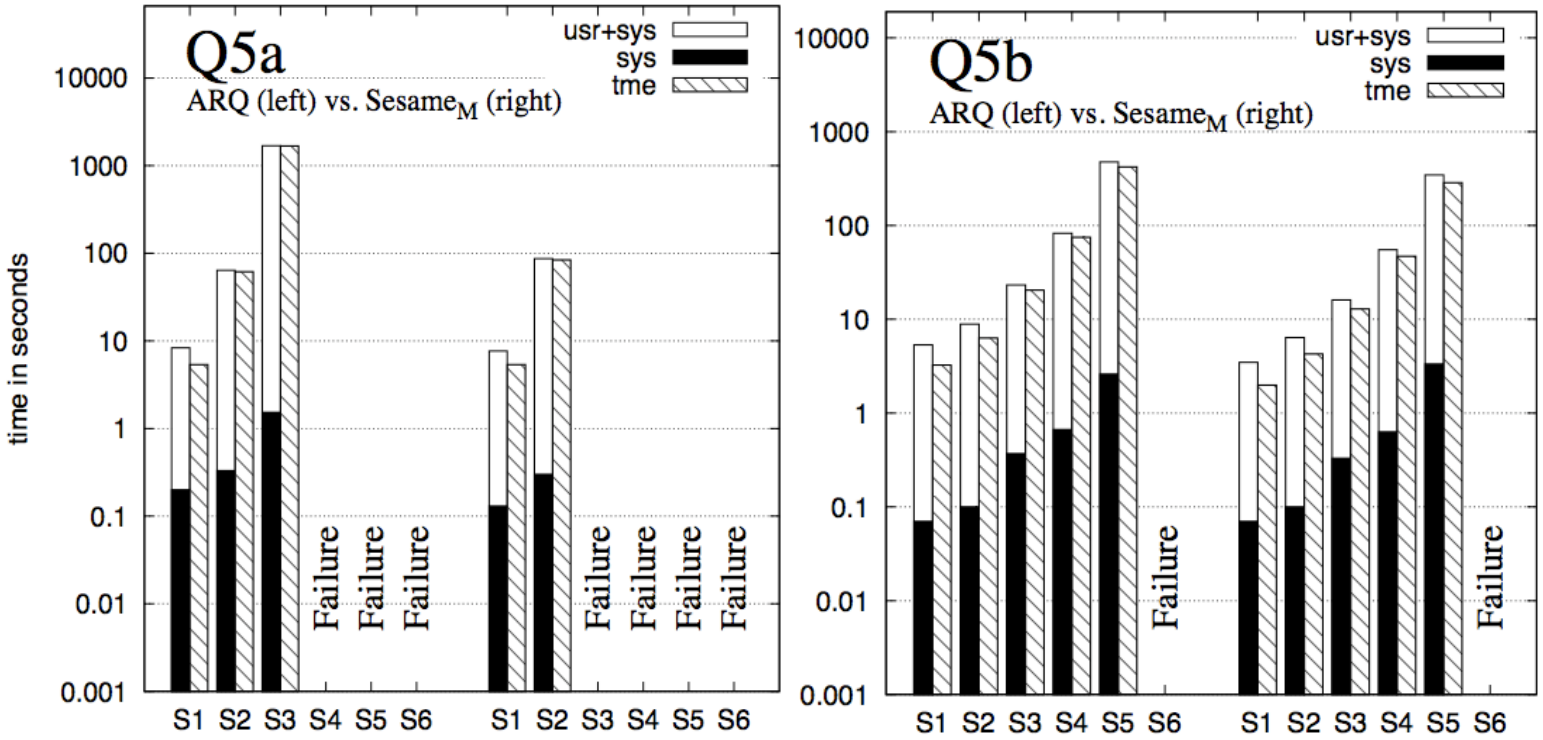
Q5

(a)
```
SELECT DISTINCT ?person ?name
WHERE { ?article rdf:type bench:Article.
        ?article dc:creator ?person.
        ?inproc rdf:type bench:Inproceedings.
        ?inproc dc:creator ?person2.
        ?person foaf:name ?name.
        ?person2 foaf:name ?name2
        FILTER(?name=?name2) }
```

(b)
```
SELECT DISTINCT ?person ?name
WHERE { ?article rdf:type bench:Article.
        ?article dc:creator ?person.
        ?inproc rdf:type bench:Inproceedings.
        ?inproc dc:creator ?person.
        ?person foaf:name ?name }
```

Benchmark Results

- Desktop PC (2.13GHz CPU, 4GB RAM)
- Failure means timeout (30min) or memory exhaustion
- Main memory engines: ARQ, Sesame$_M$



Document size in #triples:

S1: 10k
S2: 50k
S3: 250k
S4: 1M
S5: 5M
S6: 25M

Example Query

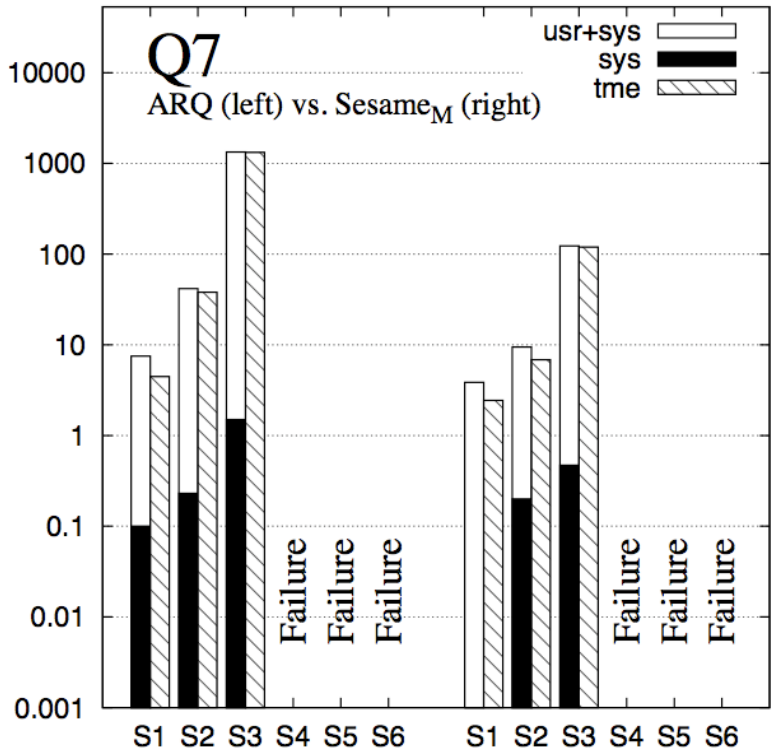
Return the titles of all papers that have been cited at least once, but **not** *by any paper that has* **not** *been cited itself*

Q7

```
SELECT DISTINCT ?title
WHERE {
  ?class rdfs:subClassOf foaf:Document.
  ?doc rdf:type ?class.
  ?doc dc:title ?title.
  ?bag2 ?member2 ?doc.
  ?doc2 dcterms:references ?bag2
  OPTIONAL {
    ?class3 rdfs:subClassOf foaf:Document.
    ?doc3 rdf:type ?class3.
    ?doc3 dcterms:references ?bag3.
    ?bag3 ?member3 ?doc
    OPTIONAL {
      ?class4 rdfs:subClassOf foaf:Document.
      ?doc4 rdf:type ?class4.
      ?doc4 dcterms:references ?bag4.
      ?bag4 ?member4 ?doc3 }
    FILTER (!bound(?doc4)) }
  FILTER (!bound(?doc3)) }
```

Benchmark Results

- Desktop PC (2.13GHz CPU, 4GB RAM)
- Failure means timeout (30min) or memory exhaustion
- Main memory engines: ARQ, Sesame$_M$



Document size in #triples:

S1: 10k
S2: 50k
S3: 250k
S4: 1M
S5: 5M
S6: 25M

Benchmark Results

- None of the four tested engines/ configurations scales to large data
- Severe problems for more complex queries, e.g. those involving negation
- Experimental results give interesting hints on possible future optimization approaches

Conclusion

- SP²Bench data generator and queries at
  http://dbis.informatik.uni-freiburg.de/index.php?project=SP2B
- Data generator relies on detailed study of DBLP, mirrors vital social-world relationships
- Queries test typical SPARQL/RDF patterns and also optimization approaches
- More experimental results in

M. Schmidt, T. Hornung, N. Kuechlin, G. Lausen, C. Pinkel. An Experimental Comparison of RDF Data Management Approaches in a SPARQL Benchmark Scenario. In ISWC, 2008.

Conclusion of experiments: still work left to do!

Related Work (selected references)

A. Chebotko et al.. Semantics Preserving SPARQL-to-SQL Query Translation for Optional Graph Patterns. TR-DB-052006-CLJF.

C. Bizer and R. Cyganiak. D2R Server publishing the DBLP Bibliography Database, 2007. http://www4.wiwiss.fu-berlin.de/dblp/.

C. Bizer and A. Schultz. The Berlin SPARQL Benchmark. http://www4.wiwiss.fu-berlin.de/bizer/BerlinSPARQLBenchmark/.

C. Bizer and A. Schultz. Benchmarking the Performance Benchmarking the Performance of Storage Systems that of Storage Systems that expose SPARQL Endpoints. In SSWS, 2008.

R. Cyganiac. A relational algebra for SPARQL. Technical Report, HP Laboratories Bristol.

D.J. Abadi et al.. Scalable Semantic Web Data Management Using Vertical Partitioning. In *VLDB*, pages 411–422, 2007.

D..J. Abadi et al.. Using the Barton libraries dataset as an RDF benchmark. TR MIT-CSAIL-TR-2007-036, MIT.

J. Gray. *The Benchmark Handbook for Database and Transaction Systems*. Morgan Kaufmann, 1993.

S. Groppe, J. Groppe, and V. Linnemann. Using an Index of Precomputed Joins in order to speed up SPARQL Processing. In *ICEIS*, 2007.

S. Harris and N. Gibbins. 3store: Efficient Bulk RDF Storage. In *PSSS*, 2003.

A. Harth and S. Decker. Optimized Index Structures for Querying RDF from the Web. In *LA-WEB*, 2005.

O. Hartwig and R. Heese. The SPARQL Query Graph Model for Query Optimization. In *ESWC*, 2007.

M. Ley. DBLP Database. http://www.informatik.uni-trier.de/ ley/db/.

M. Schmidt, T. Hornung, N. Kuechlin, G. Lausen, C. Pinkel. An Experimental Comparison of RDF Data Management Approaches in a SPARQL Benchmark Scenario. In Proc. ISWC, 2008.

J. Perez, M. Arenas, and C. Gutierrez: Semantics and Complexity of SPARQL. In CoRR cs.DB/0605124, 2006.

A. Polleres: From SPARQL to Rules (and back). In WWW, pages 787-796, 2007.

L. Sidirourgos, R. Gocalves, M. Kerstin, N. Nes, and S. Manegold: Column-store support for rdf data management: not all swans are white. In VLDB 2008.