Strabon
A Semantic Geospatial DBMS

Kostis Kyzirakos, Manos Karpathiotakis, and Manolis Koubarakis

Dept. of Informatics and Telecommunications,
National and Kapodistrian University of Athens, Greece

Outline

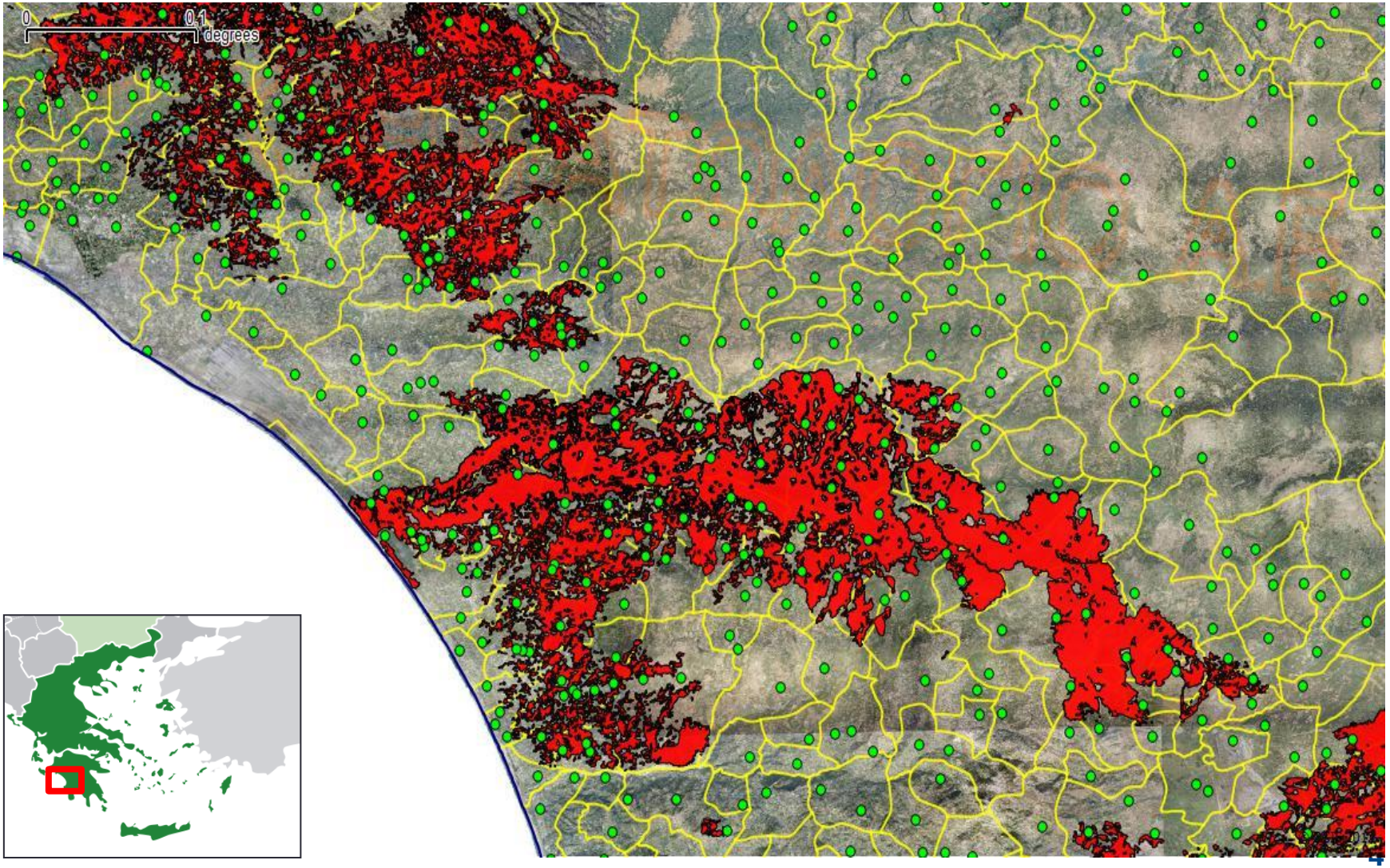
- Introduction
- The data model stRDF
- The query language stSPARQL and a comparison to GeoSPARQL
- The system Strabon for stSPARQL and GeoSPARQL
- Experimental Evaluation
- Related Work & Conclusions

Main idea

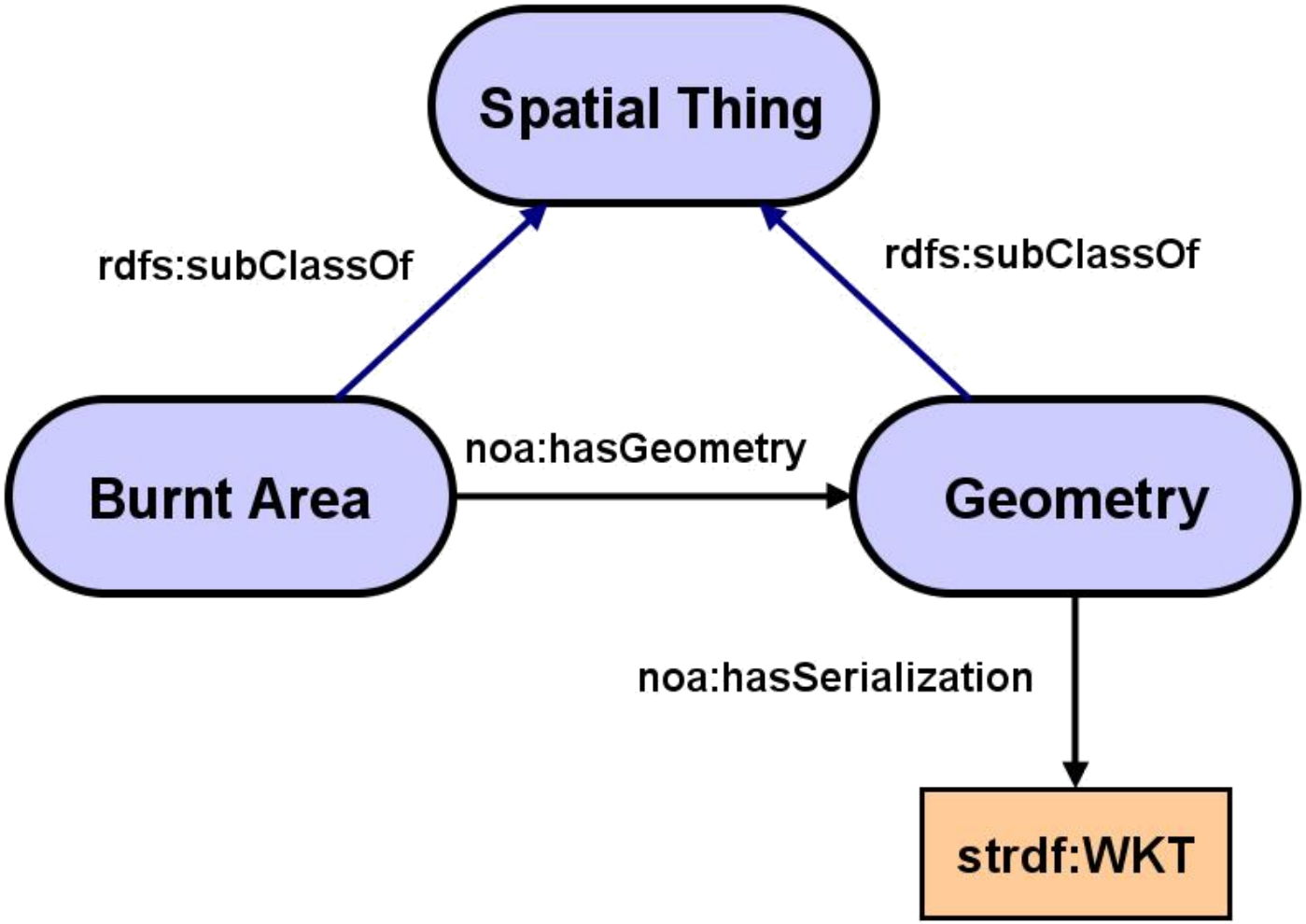
How do we **represent** and **query geospatial information** in the Semantic Web?

- Develop appropriate **vocabularies** and **ontologies**
- **Extend RDF** to take into account the **geospatial** dimension
- **Extend SPARQL** to **query** the new kinds of data
- Use **Open Geospatial Consortium (OGC)** and other **geospatial industry standards**

Example



National Observatory of Athens: Fire Products Ontology



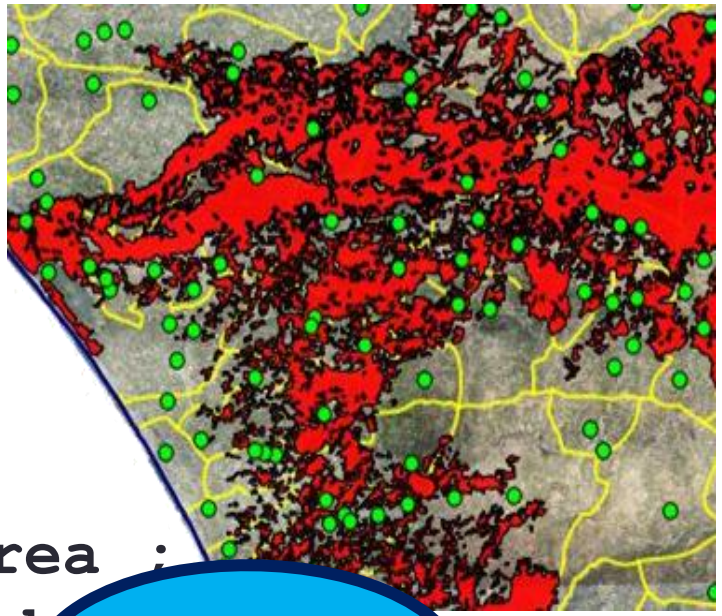
- Introduction
- The data model stRDF
- The query language stSPARQL and a comparison to GeoSPARQL
- The system Strabon for stSPARQL and GeoSPARQL
- Experimental Evaluation
- Related Work & Conclusions

The data model stRDF

The Data Model stRDF

- stRDF extends RDF with: [ESWC'10]
  - **Spatial literals** encoded by Boolean combinations of linear constraints
  - New datatype for spatial literals (`strdf:geometry`)
  - **Valid time of triples** encoded by Boolean combinations of temporal constraints
- stRDF (most recent version) [ISWC '12]
  - **Spatial literals** encoded in Well-Known Text/GML (OGC standards)
  - Valid time of triples ignored for the time being

Burnt Area Products



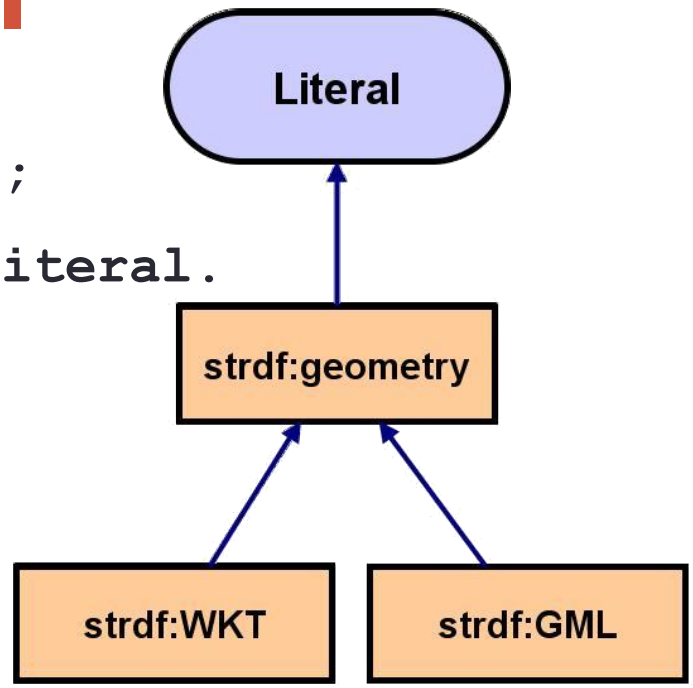
```
@prefix strdf:
<http://strdf.di.uoa.gr/
ontology#>.
@prefix noa:
<http://teleios.di.uoa.gr/
ontologies/noaOntology.owl#>.

noa:ba_15 rdf:type noa:BurntArea ;
          noa:hasGeometry noa:b

noa:ba_15g noa:hasSerialization
  "MULTIPOLYGON(((393801.42 4198827.92,
                ..., 393008 424131)));
<http://www.opengis.net/d  /srs/EPSG/0/2100>"
                                ^^strdf:WKT.
```

Spatial literal

Spatial data type

The stRDF Data Model

```
strdf:geometry rdf:type rdfs:Datatype;
               rdfs:subClassOf rdfs:Literal.
```




```
strdf:WKT   rdf:type rdfs:Datatype;
            rdfs:subClassOf strdf:geometry.

strdf:GML   rdf:type    rdfs:Datatype;
            rdfs:subClassOf   strdf:geometry.
```

- Introduction
- The data model stRDF
- The query language stSPARQL and a comparison to GeoSPARQL
- The system Strabon for stSPARQL and GeoSPARQL
- Experimental Evaluation
- Related Work & Conclusions

The query language stSPARQL

and a comparison to GeoSPARQL

stSPARQL: Geospatial SPARQL 1.1

We define a **SPARQL extension function** for each function defined in the **OpenGIS Simple Features Access** standard

- **Basic functions**
  - Get a property of a geometry (e.g., **`strdf:srid`**)
  - Get the desired representation of a geometry (e.g., **`strdf:AsText`**)
  - Test whether a certain condition holds (e.g., **`strdf:IsEmpty`**, **`strdf:IsSimple`**)
- Functions for **testing topological spatial relationships** (e.g., **`strdf:equals`**, **`strdf:intersects`**)
  - OGC Simple Features Access, Egenhofer, RCC-8
- **Spatial analysis** functions
  - Construct new geometric objects from existing geometric objects (e.g., **`strdf:buffer`**, **`strdf:intersection`**, **`strdf:convexHull`**)
  - Spatial metric functions (e.g., **`strdf:distance`**, **`strdf:area`**)
- **Spatial aggregate** functions (e.g., **`strdf:union`**, **`strdf:extent`**)

stSPARQL: Geospatial SPARQL 1.1

Select clause

- Construction of new geometries (e.g., `strdf:buffer(?geo, 0.1)`)
- Spatial aggregate functions (e.g., `strdf:union(?geo)`)
- Metric functions (e.g., `strdf:area(?geo)`)

Filter clause

- Functions for testing topological relationships between spatial terms (e.g., `strdf:contains(?G1, strdf:union(?G2, ?G3))`)
- Numeric expressions involving spatial metric functions (e.g., `strdf:area(?G1) ≤ 2*strdf:area(?G2)`)
- Boolean combinations

Having clause

- Boolean expressions involving spatial aggregate functions and spatial metric functions or functions testing for topological relationships between spatial terms (e.g., `strdf:area(strdf:union(?geo))>1`)

Updates

stSPARQL: An example (1/2)

Find coniferous forests that have been affected by fires


```
SELECT ?forest ?burntArea
WHERE {
   ?burntArea   rdf:type noa:BurntArea;
                noa:hasGeometry [
                noa:hasSerialization ?baGeo].
   ?forest      rdf:type noa:Region;
                clc:hasLandCover noa:coniferousForest;
                clc:hasGeometry [
                clc:hasSerialization ?fGeom].
   FILTER(strdf:intersects(?baGeom ?fGeom) )
}
```

Spatial Function

stSPARQL: An example (2/2)

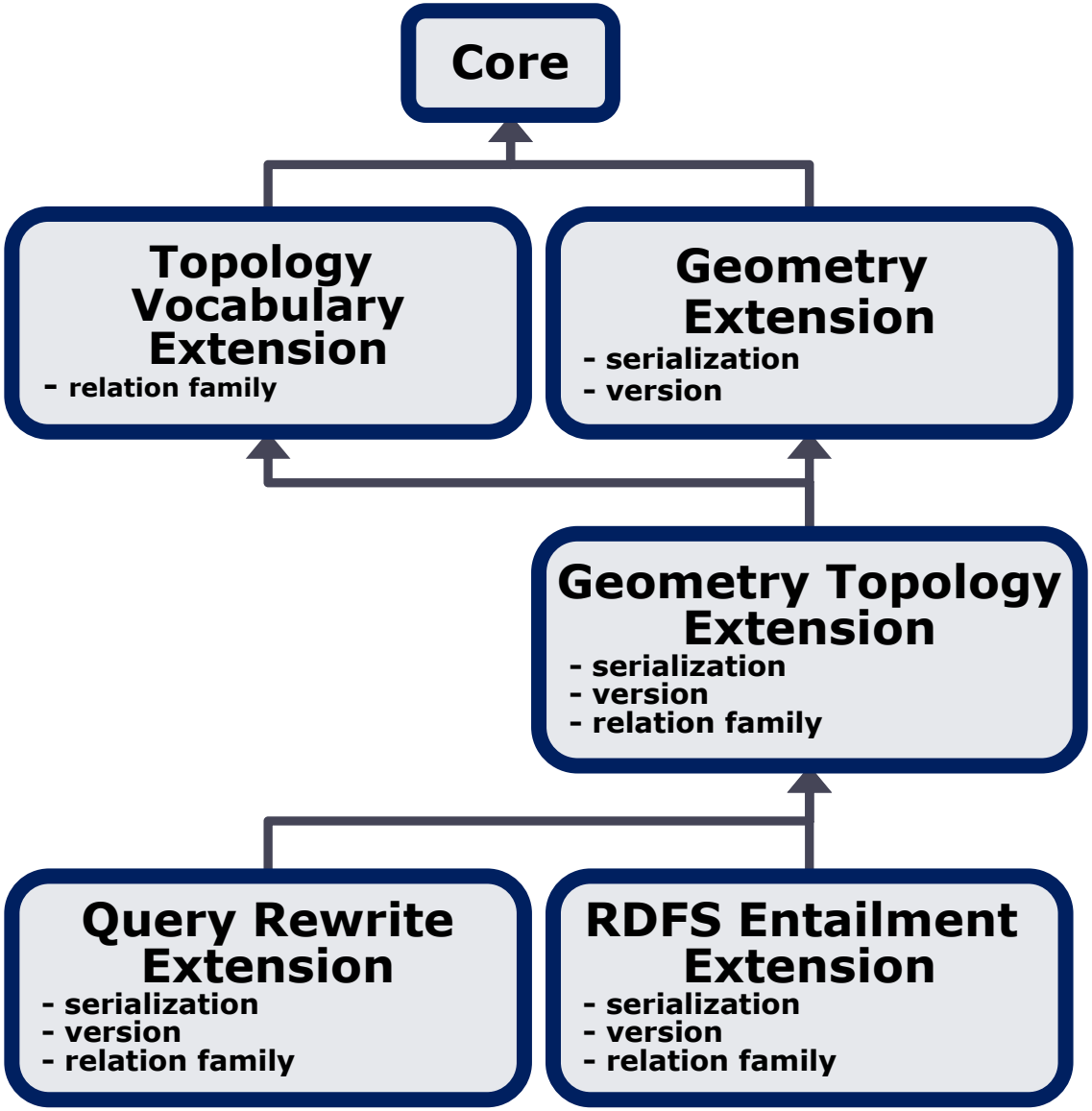
Isolate the parts of the burnt areas that lie in coniferous forests.

Spatial Aggregate

```
SELECT ?burntArea
(strdf:intersection(?baGeom
                    strdf:union(?fGeom))
 AS ?burntForest)
WHERE {
   ?burntArea   rdf:type noa:BurntArea;
                noa:hasGeometry [
                noa:hasSerialization ?baGeo].
   ?forest      rdf:type noa:Region;
                clc:hasLandCover noa:coniferousForest;
                clc:hasGeometry [
                clc:hasSerialization ?fGeom].
   FILTER(strdf:intersects(?baGeom ?fGeom))
}
GROUP BY ?burntArea ?baGeom
```

Spatial Function

The OGC Standard GeoSPARQL



Parameters

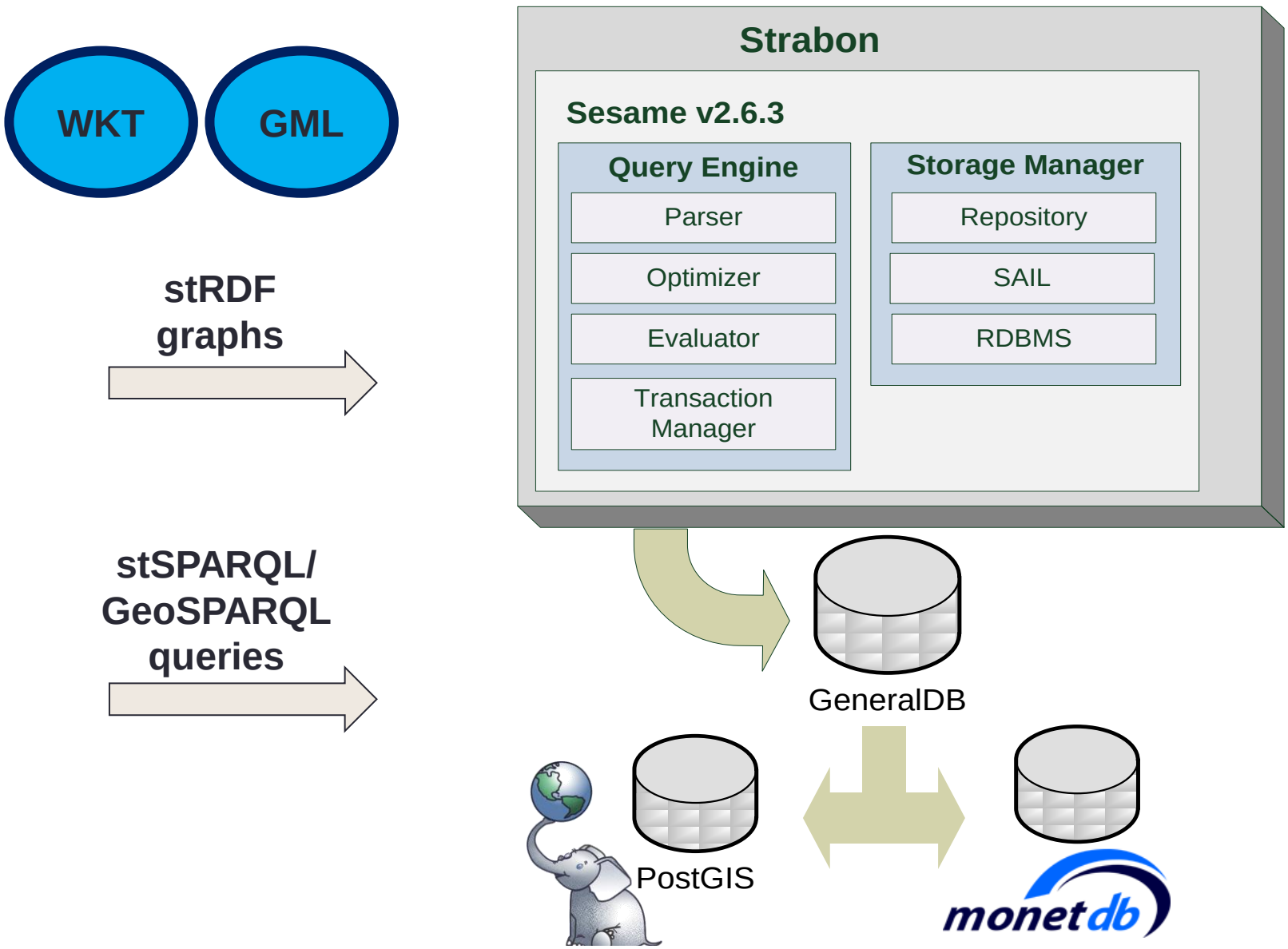
- **Serialization**
  - WKT
  - GML
- **Relation Family**
  - Simple Features
  - RCC-8
  - Egenhofer

- Introduction
- The data model stRDF
- The query language stSPARQL and a comparison to GeoSPARQL
- The system Strabon for stSPARQL and GeoSPARQL
- Experimental Evaluation
- Related Work & Conclusions

The system Strabon for stSPARQL and GeoSPARQL

strabon.di.uoa.gr

Strabon Architecture



Storage Scheme

Triples

SUBJECT	PREDICATE	OBJECT
noa:ba_15	rdf:type	noa:BurntArea
noa:ba_15	noa:hasGeometry	noa:ba_15g
noa:ba_15g	rdf:type	noa:Geometry
noa:ba_15g	noa:hasserialization	"MULTIPOLYGON(...)"^^ strdf:WKT

Storage Scheme

Triples

SUBJECT	PREDICATE	OBJECT
1	2	3
1	4	5
5	2	6
5	7	8

Dictionary

ID	VALUE
1	noa:ba_15
2	rdf:type
3	noa:BurntArea
4	noa:hasGeometry
5	noa:ba_15g
6	noa:Geometry
7	noa:hasSerialization
8	"MULTIPOLYGON(...)"^^ strdf:WKT

Storage Scheme

type_2

SUBJECT	OBJECT
1	3
5	6

hasgeom_4

SUBJECT	OBJECT
1	5

hasserial_7

SUBJECT	OBJECT
5	8

Dictionary

ID	VALUE
1	noa:ba_15
2	rdf:type
3	noa:BurntArea
4	noa:hasGeometry
5	noa:ba_15g
6	noa:Geometry
7	noa:hasSerialization
8	"MULTIPOLYGON(...)"^^ strdf:WKT

Storage Scheme

type_2

SUBJECT	OBJECT
1	3
5	6

hasgeom_4

SUBJECT	OBJECT
1	5

hasserial_7

SUBJECT	OBJECT
5	8

uri_values

ID	VALUE
1	noa:ba_15
2	rdf:type
3	noa:BurntArea
4	noa:hasGeometry
5	noa:ba_15g
6	noa:Geometry
7	noa:hasSerialization

label_values

ID	VALUE
8	"MULTIPOLYGON(...)"^^ strdf:WKT

datatype_values

ID	VALUE
8	strdf:WKT

Storage Scheme

type_2

SUBJECT	OBJECT
1	3
5	6

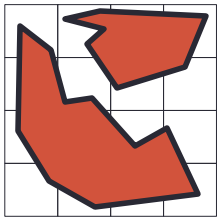
hasgeom_4

SUBJECT	OBJECT
1	5

hasserial_7

SUBJECT	OBJECT
5	8

geo_values

ID	VALUE
8	

uri_values

ID	VALUE
1	noa:ba_15
2	rdf:type
3	noa:BurntArea
4	noa:hasGeometry
5	noa:ba_15g
6	noa:Geometry
7	noa:hasSerialization

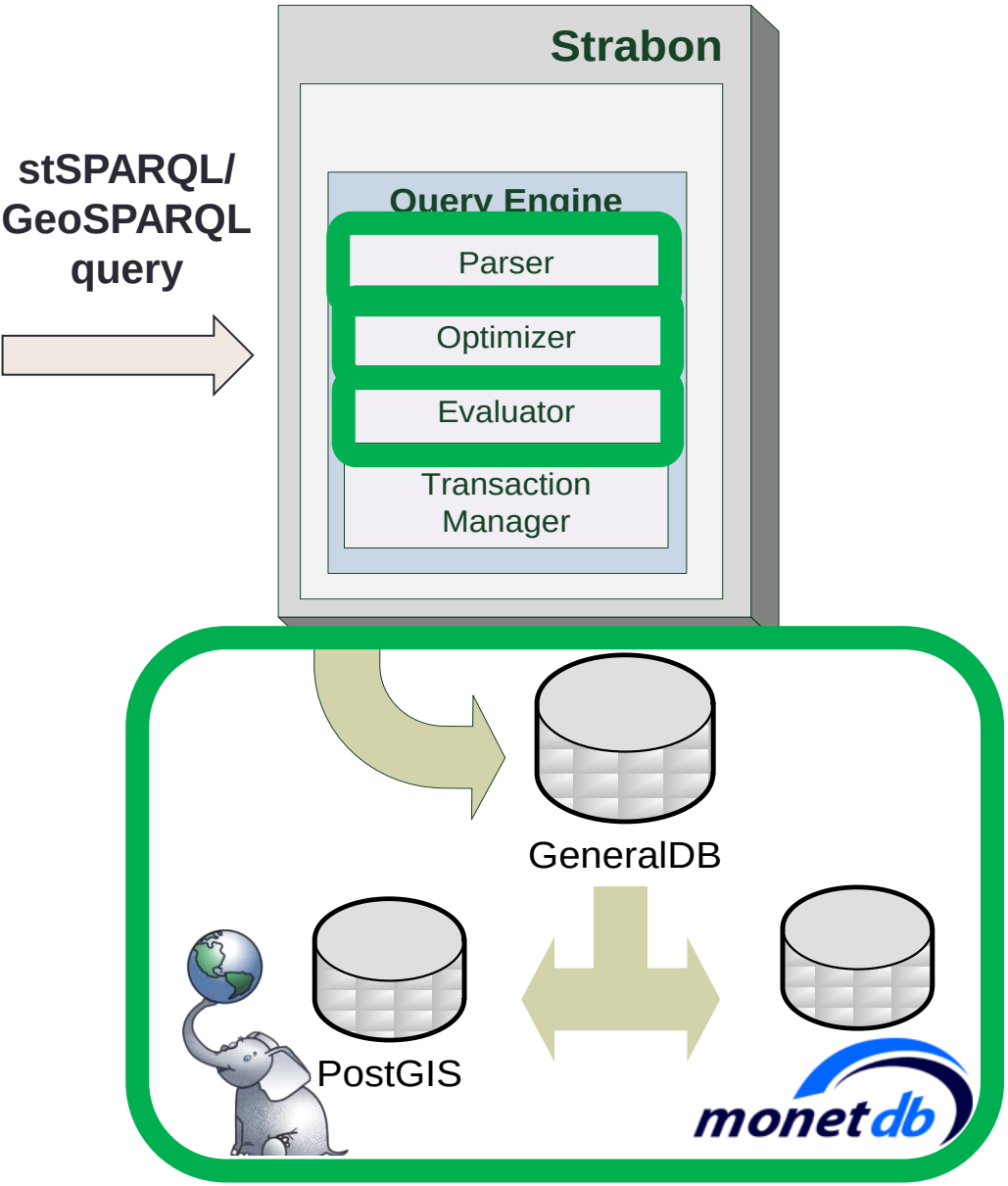
label_values

ID	VALUE
8	"MULTIPOLYGON(...)"^^ strdf:WKT

datatype_values

ID	VALUE
8	strdf:WKT

Query Processing



- Parser generates abstract syntax tree
- Abstract syntax tree mapped to internal algebra of Sesame
- Standard optimizations performed
- Evaluator produces corresponding SQL query
- DBMS evaluates the SQL query
- Post-processing

Query Processing (cont'd)

- **Deviate** from the evaluation strategy of Sesame for SPARQL extension functions
- **Push** the evaluation of extension functions **to underlying DBMS**
  - **Spatial predicates** evaluated by PostGIS
  - **Spatial joins** now affect query plan
    - Avoid Cartesian products
- Results may be returned in well-known industry formats
  - KML/KMZ
  - GeoJSON
  - GML

- Introduction
- The data model stRDF
- The query language stSPARQL and a comparison to GeoSPARQL
- The system Strabon for stSPARQL and GeoSPARQL
- Experimental Evaluation
- Related Work & Conclusions

Experimental Evaluation

Experimental Evaluation

- **Goal**: Evaluate the performance of Strabon vs other systems
- Real workload based on geospatial linked datasets
  - **150 million** triples
- Synthetic workload
  - **Half a billion** triples

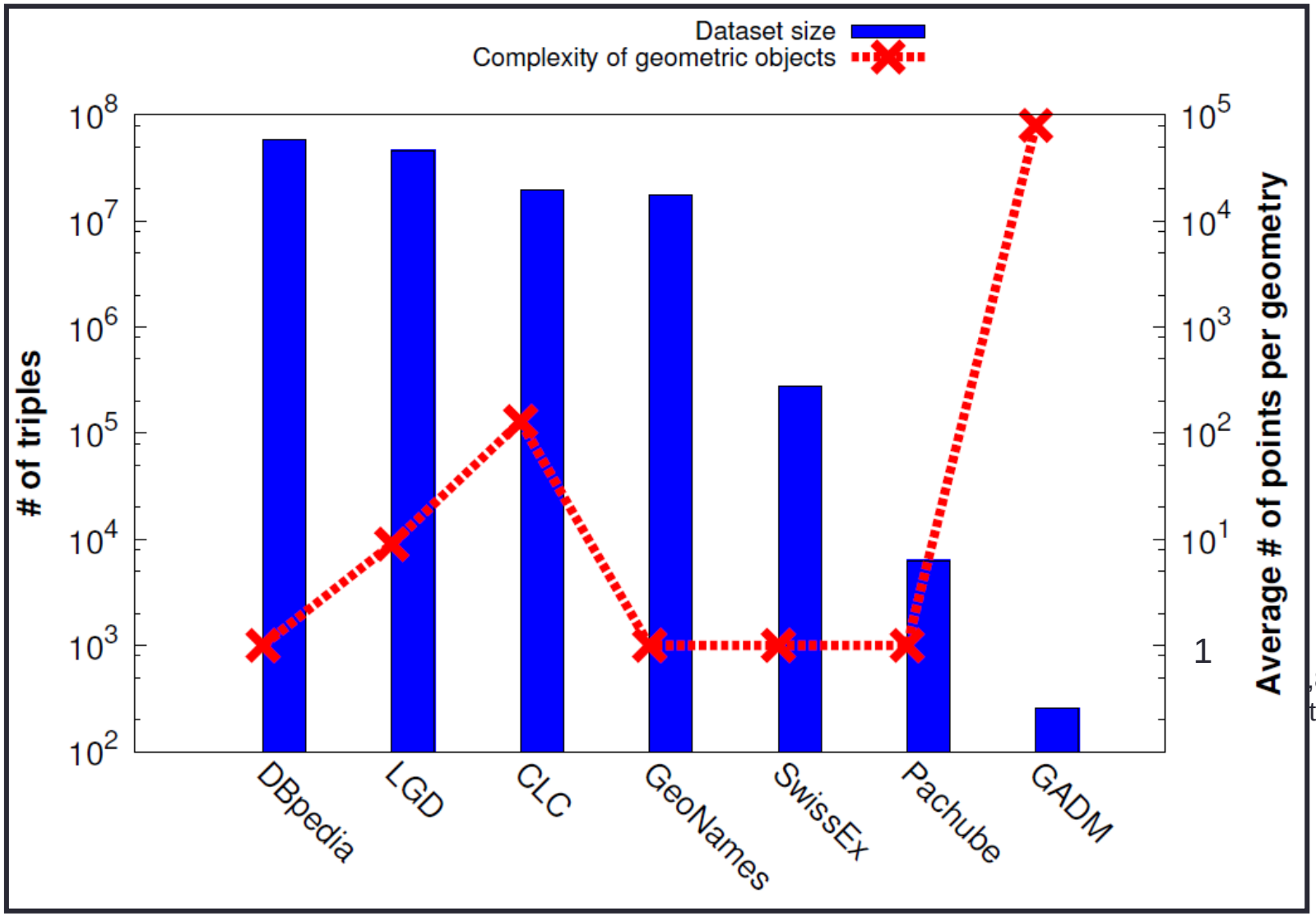
Strabon vs other systems

- Strabon over PostgreSQL (Strabon-PG)
- Strabon over System X (Strabon-X)
- Implementation over RDF-3X (Brodt et. al)
- Parliament (BBN Technologies)
- Naive Implementation over Sesame

Real world workload: Data

	DBpedia	GeoNames	LGD	Pachube	SwissEx	CLC	GADM
Size	**7.1 GB**	**2.1 GB**	**6.6 GB**	**828 KB**	**33 MB**	**14 GB**	**146 MB**
Triples	58,722,893	17,688,602	46,296,978	6,333	277,919	19,711,926	255
Spatial Terms	386,205	1,262,356	5,414,032	101	687	2,190,214	51
Distinct Spatial Terms	375,087	1,099,964	5,035,981	70	623	2,190,214	51
Points	375,087	1,099,964	3,205,015	70	623	-	-
Linestrings	-	-	353,714	-	-	-	- 79,831 vertices
Polygons	-	-	1,704,650	-	-	2,190,214	51

Real world workload: Data



,831
tices

Real world workload: Queries

	Commonly Used	Spatial Selection	Spatial Join
Query 1	X		
Query 2	X		
Query 3		X	
Query 4		X	
Query 5			X
Query 6	X		X
Query 7	X		X
Query 8			X

Queries with Spatial Joins	Points	Lines	Polygons
Query 5	X X		X X
Query 6	X	X	X
Query 7			X
Query 8	X	X	X

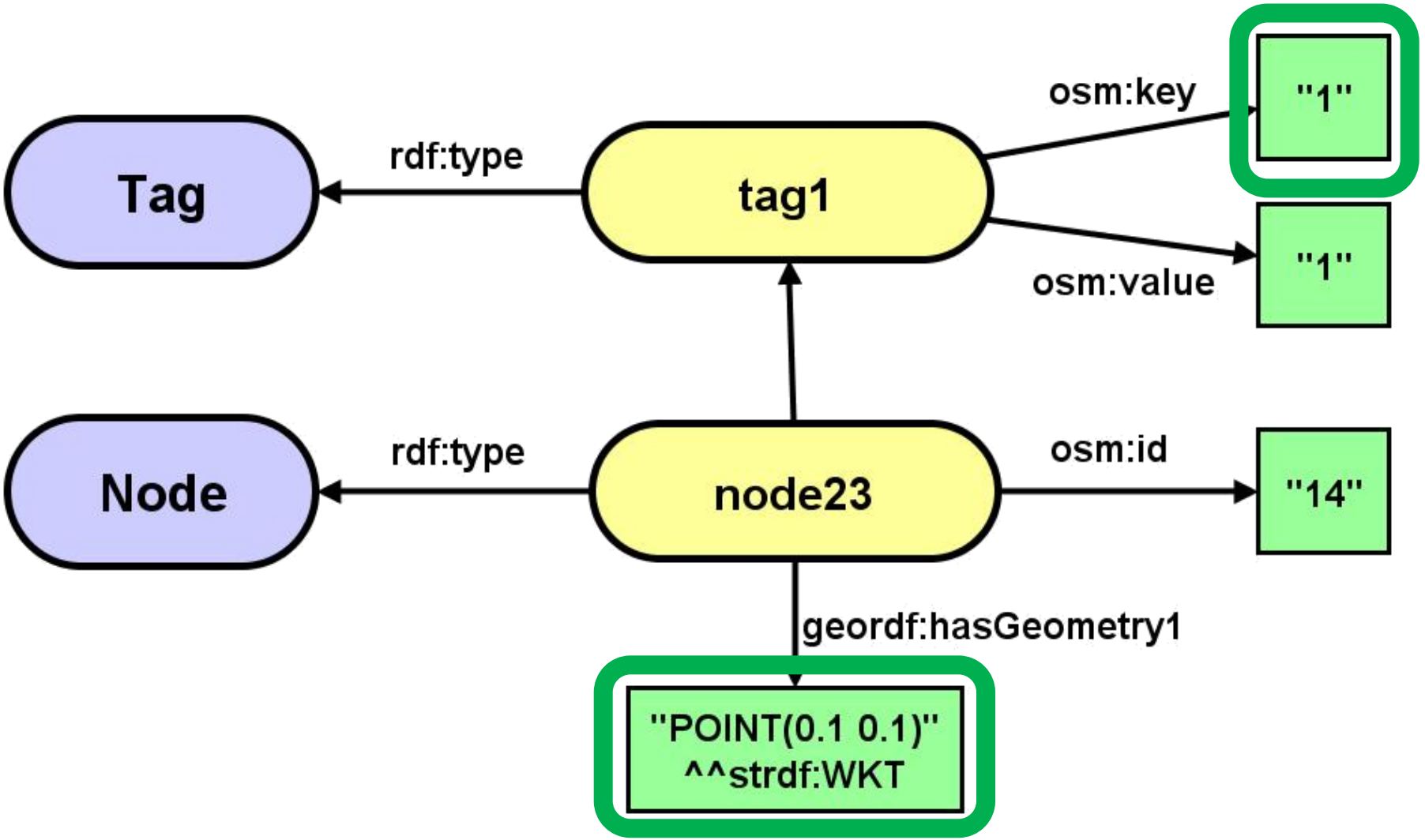
Real world workload: Results

Cache State	System	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Cold (sec.)	Naive	0.08	1.65	>8h	28.88	89	170	844	1.699
	Strabon PG	2.01	6.79	41.39	10.11	78.69	60.25	9.23	702.55
	Strabon X	1.74	3.05	1623	46.52	12.57	2409	>8h	57.83
	Parliament	2.12	6.46	>8h	229.72	1130	872	3627	3786
Warm (sec.)	Naïve	0.01	0.03	>8h	0.79	43.07	88	708	1712
	Strabon PG	0.01	0.81	0.96	1.66	38.74	1.22	2.92	648.1
	Strabon X	0.01	0.26	1604.9	35.59	0.18	3196	>8h	44.72
	Parliament	0.01	0.04	>8h	10.91	358.92	483.29	2771	3502

Synthetic Workload

- Workload based on a synthetic dataset
  - Dataset based on OpenStreetMaps
  - **10 million** triples (**2 GB**) up to **half a billion** triples (**50GB**)
    - Implemented custom bulk loader
  - Triples with **spatial literals**: **1** up to **46 million** triples
- Response time of queries with various **thematic** and **spatial selectivities**

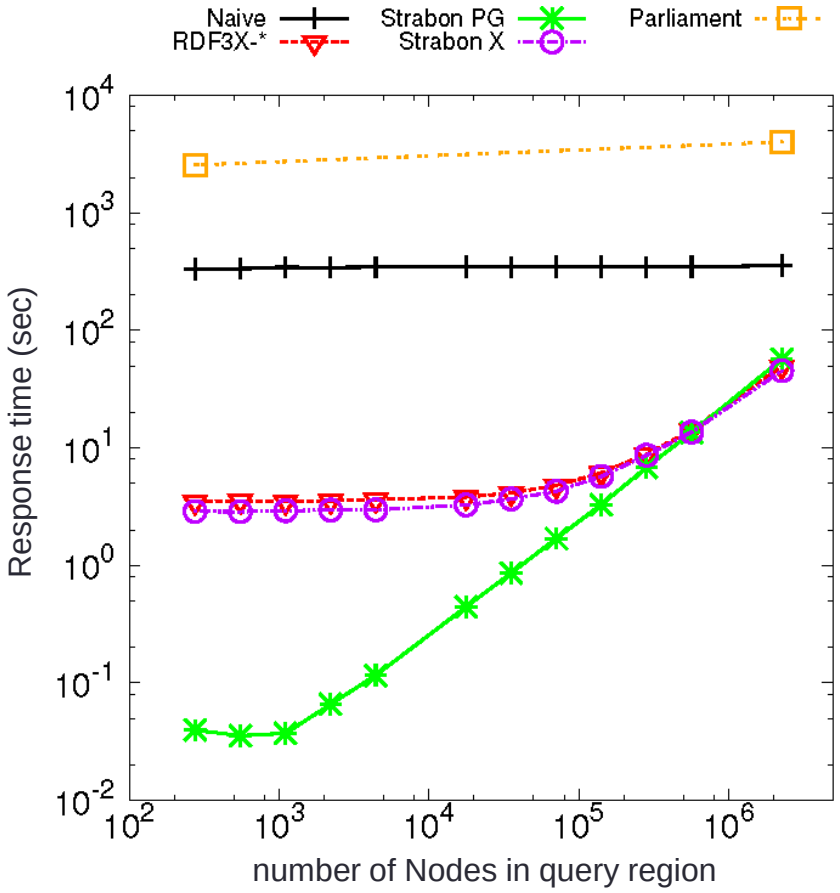
Synthetic Workload: Sample Data



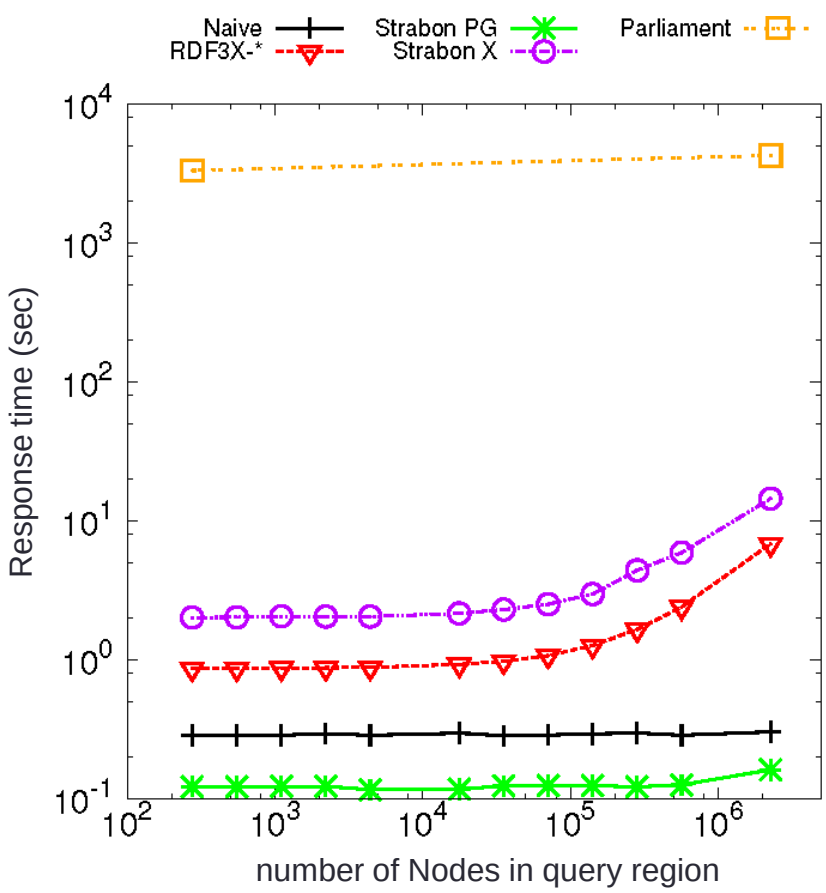
Synthetic Workload: Sample stSPARQL Query

```
SELECT *

WHERE {
     ?tag geordf:key "1" .
     ?node geordf:hasTag ?tag .
     ?node geo:hasGeography1 ?geo .

FILTER (strdf:inside(?geo,
         "POLYGON((-1 -1, 0.056568542 -1,
        0.056568542 0.056568542,
        -1 0.056568542, -1 -1))"^^strdf:WKT
       ))

}
```

Real-world Workload: 100 million triples – warm caches

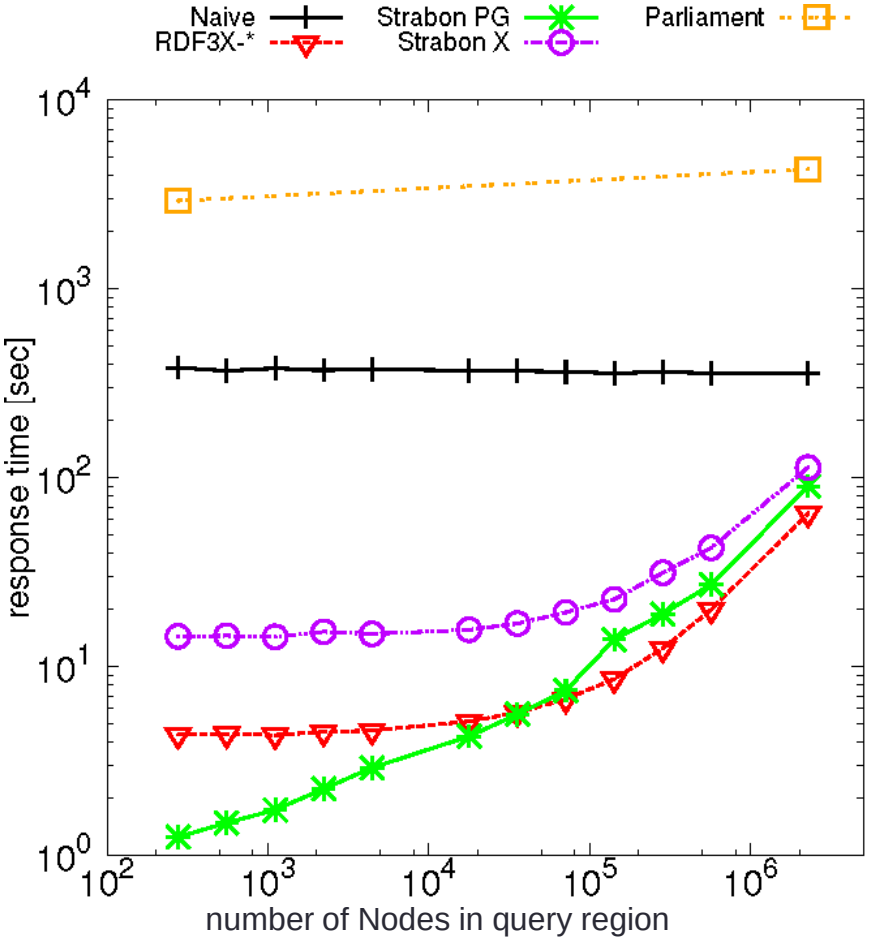


Tag 1

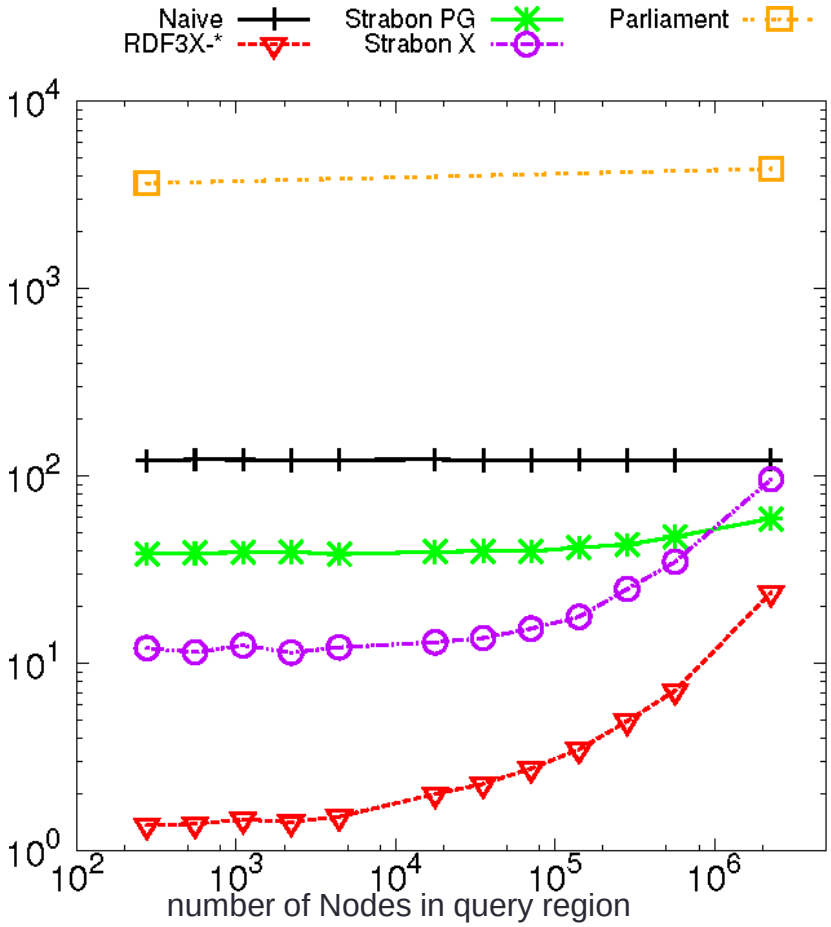


Tag 1024

Real-world Workload: 100 million triples – cold caches

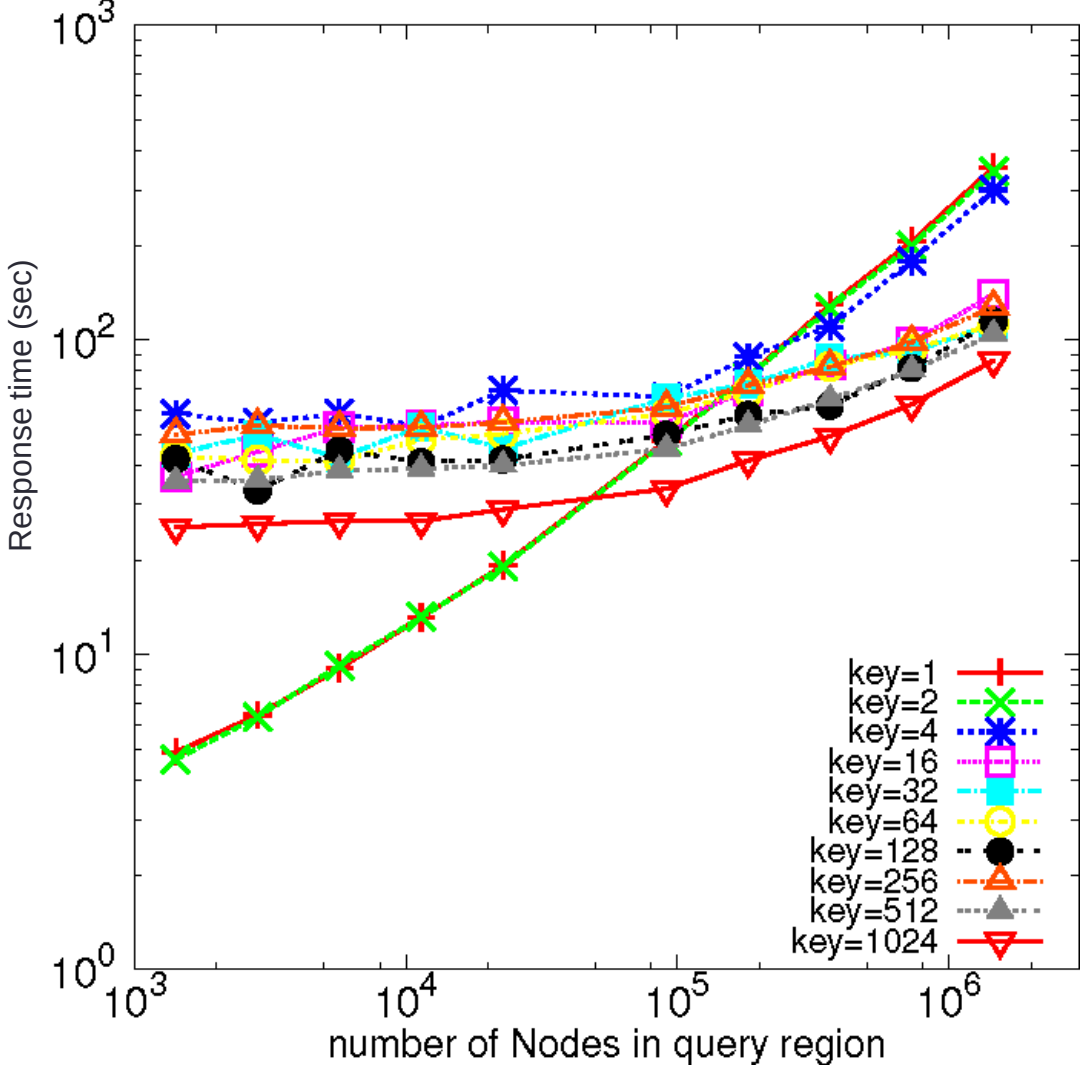


Tag 1

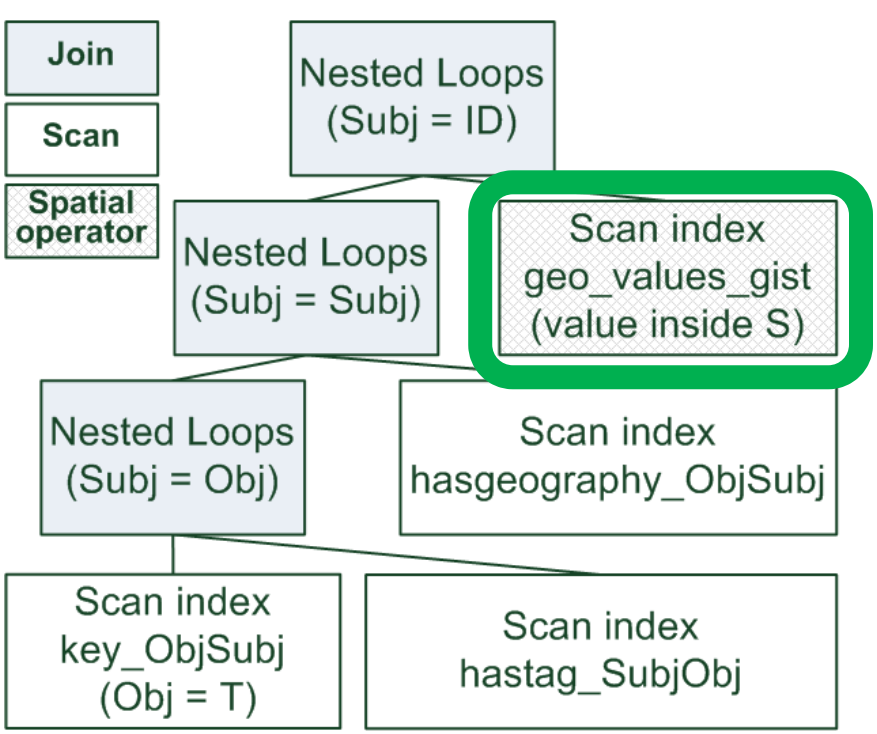


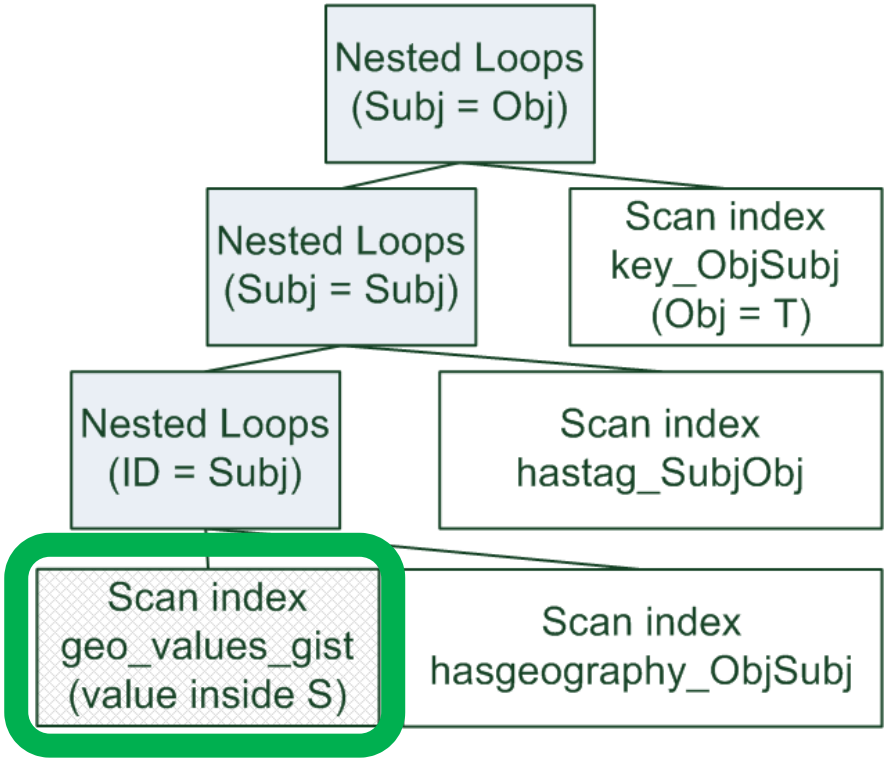
Tag 1024

Strabon-PG: 500 million triples Tags comparison



Real-world Workload: Query Plans





Findings

- Strabon over PostgreSQL outperforms other systems in case of warm caches
- Results in case of cold caches mixed
- PostreSQL optimizer needs to take into account spatial selectivity
  - PostGIS 2.0 moves towards it

System	Language	Index	Geometries	CRS support	Geospatial Function Support
Strabon	stSPARQL/ GeoSPARQL*	R-tree-over-GiST	WKT / GML support	Yes	• OGC-SFA • Egenhofer • RCC-8
Parliament	GeoSPARQL*	R-Tree	WKT / GML support	Yes	•OGC-SFA •Egenhofer •RCC-8
Oracle 12c	GeoSPARQL	R-Tree, Quadtree	WKT	Yes	•OGC-SFA
Brodt et al. (RDF-3X)	SPARQL	R-Tree	WKT support	No	OGC-SFA
Perry	SPARQL-ST	R-Tree	GeoRSS GML	Yes	RCC-8
AllegroGraph	Extended SPARQL	Distribution sweeping technique	2D point geometries	Partial	•Buffer •Bounding Box •Distance
OWLIM	Extended SPARQL	Custom	2D point geometries	No	•Point-in-polygon •Buffer •Distance
Virtuoso	SPARQL	R-Tree	2D point geometries	Yes	SQL/MM (subset)
uSeekM	SPARQL	R-tree-over GiST	WKT support	No	OGC-SFA

- Introduction
- The data model stRDF
- The query language stSPARQL and a comparison to GeoSPARQL
- The system Strabon for stSPARQL and GeoSPARQL
- Experimental Evaluation
- Related Work & Conclusions

Related Work & Conclusions

Future Work

- Use even larger datasets
  - Test with 1 billion triples successful
- Implement the temporal part of stSPARQL
- Develop a benchmark for geospatial RDF stores
  - Consider more systems
    - GeoSPARQL implementation of Oracle
    - Virtuoso
- stSPARQL query processing in MonetDB
- Go distributed!
  - Federated queries

Thanks! Any Questions?

- Strabon
  - Manolis Koubarakis, Kostis Kyzirakos, Manos Karpathiotakis, Charalampos Nikolaou, Giorgos Garbis, Konstantina Bereta, Kallirroi Dogani, Stella Giannakopoulou and Panayiotis Smeros.
  - Web site: **http://strabon.di.uoa.gr**
  - Mercurial repository: **http://hg.strabon.di.uoa.gr**
  - Trac: **http://bug.strabon.di.uoa.gr**
  - Mailing list: **http://cgi.di.uoa.gr/~mailman/listinfo/strabon-users**

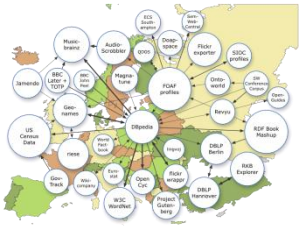








Thanks! Any Questions?



- Strabon
  - Manolis Koubarakis, Kostis Kyzirakos, Manos Karpathiotakis, Charalampos Nikolaou, Giorgos Garbis, Konstantina Bereta, Kallirroi Dogani, Stella Giannakopoulou and Panayiotis Smeros.
  - Web site: **http://strabon.di.uoa.gr**
  - Mercurial repository: **http://hg.strabon.di.uoa.gr**
  - Trac: **http://bug.strabon.di.uoa.gr**
  - Mailing list: **http://cgi.di.uoa.gr/~mailman/listinfo/strabon-users**
- Real Time Fire Monitoring Service, National Obervatory of Athens
  - **http://papos.space.noa.gr/fend_static**
- Greek Linked Open Data
  - **http://www.linkedopendata.gr**
  -
- TELEIOS EU Project
  - **http://www.earthobservatory.eu**
- SemsorGrid4Env EU Project
  - **http://www.semsorgrid4env.eu**

[ISWC '12 poster and challenge]

TELEIOS

SemSorGrid