On the Role of the GRAPH Clause in the Performance of Federated SPARQL Queries

David Chaves-Fraga, Ontology Engineering Group
Universidad Politécnica de Madrid, Spain
Claudio Gutierrez, CIWS & Universidad de Chile
Oscar Corcho, OEG - UPM

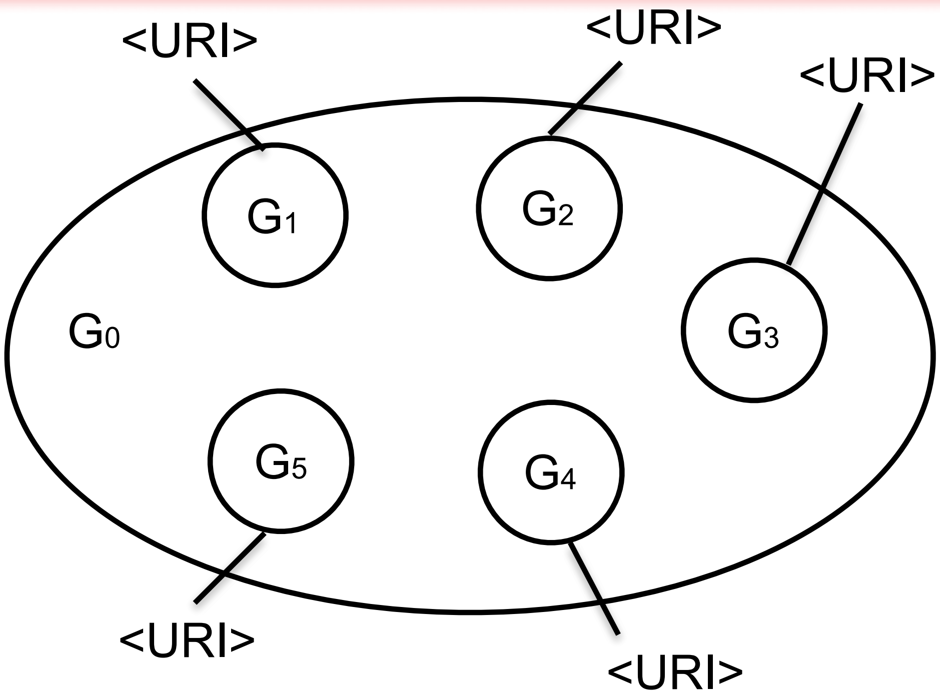
✉ dchaves@email
@dchavesf

📅 22/10/2017
📍 PROFILES17

Federated Query and SPARQL endpoint

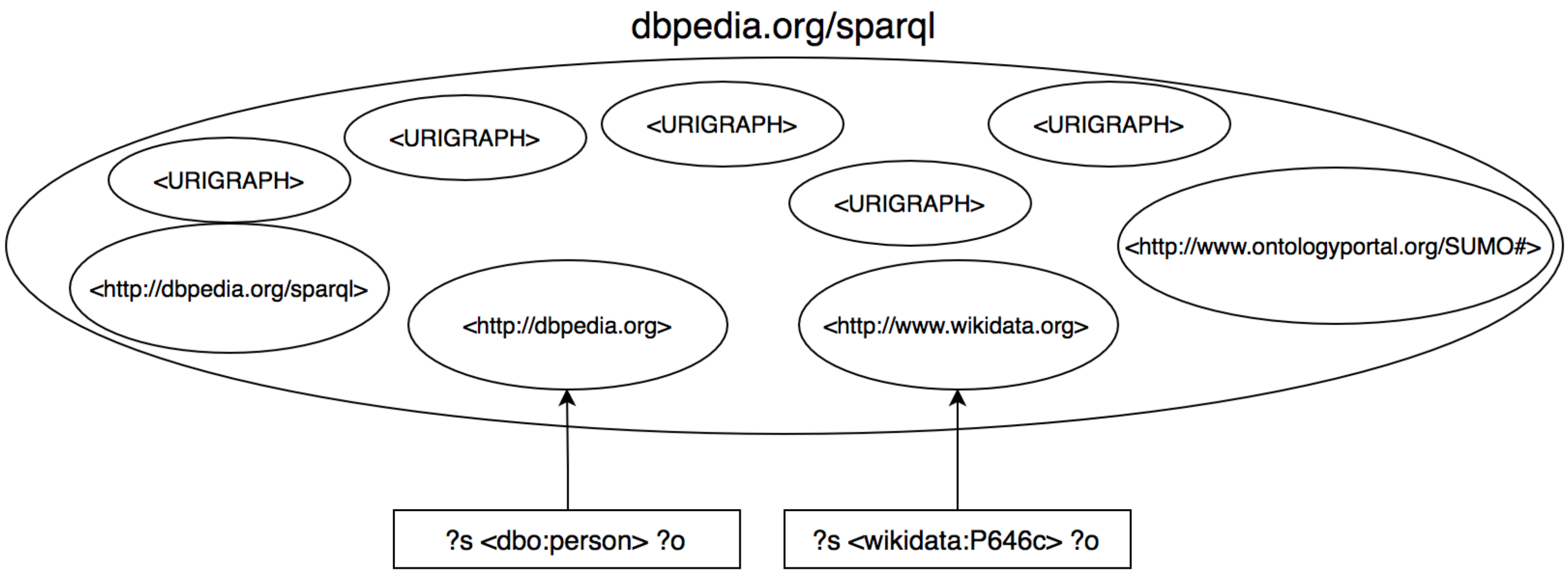
```
select * where {
    service <s1> {GP1}
    service <s2> {GP2}
    service <s3> {GP3}
}
```

Federated Query



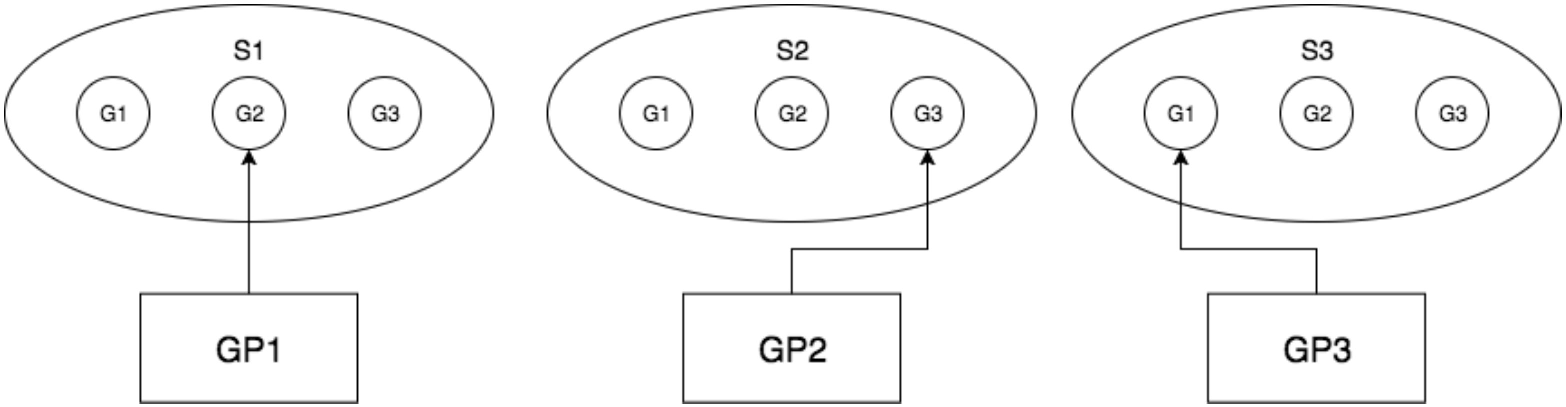
SPARQL endpoint

Improve federated query evaluation by systematically using the GRAPH clause to rewrite a query into a semantically equivalent one



- Many graph patterns bind results from only one of the graphs in the SPARQL endpoints
- The server can have more information and optimize the retrieval

Reducing the space of a federated SPARQL query



```
   select * where {
       service <s1> {GP1}
       service <s2> {GP2}
       service <s3> {GP3}
   }
```

```
select * from named <s1-g2>, <s2-g3>, <s3-g1>{
    service <s1> {graph <s1-g2> {GP1}}
    service <s2> {graph <s2-g3> {GP2}}
    service <s3> {graph <s3-g1> {GP3}}
}
```

- Hypothesis: This method adds efficiency in the evaluation of the query
- Assumption: The identification of the GP

- We use the real public instances of SPARQL endpoints of:
  - DBpedia
  - KB3city
  - Linkedmdb
- 5 queries - two different approaches (with/without GRAPH clause):
  - 5 executions per approach with the disabled cache.
- Queries available at: https://github.com/dachafra/federated-graph-queries

Problems:

- There is not exist any benchmarking for federated SPARQL queries that includes SERVICE and GRAPH clauses.
- Deal with availability and real performance of the SPARQL endpoints

	Q1	Q2	Q3	Q4	Q5
GRAPH	**13,51**	**3,22**	**1,98**	1,73	**8,56**
SERVICE	14,63	4,36	2,88	**1,69**	9,12

- 5%-10% improvement
- Q2-Q3-Q4: kb.3city.com SPARQL endpoint
  - 73 graphs
  - Almost 5 millions of triples
- Q1-Q5: dbpedia.com SPARQL endpoint:
  - 19 graphs
  - 285 millions of triples

Preliminary results

	Q1	Q2	Q3	Q4	Q5
GRAPH	**13,51**	**3,22**	**1,98**	1,73	**8,56**
SERVICE	14,63	4,36	2,88	**1,69**	9,12

- 5%-10% improvement
- Q2-Q3-Q4: kb.3city.com SPARQL
  - 73 graphs
  - Almost 5 millions of triples
- Q1-Q5: dbpedia.com SPARQL endpoint:
  - 19 graphs
  - 285 millions of triples

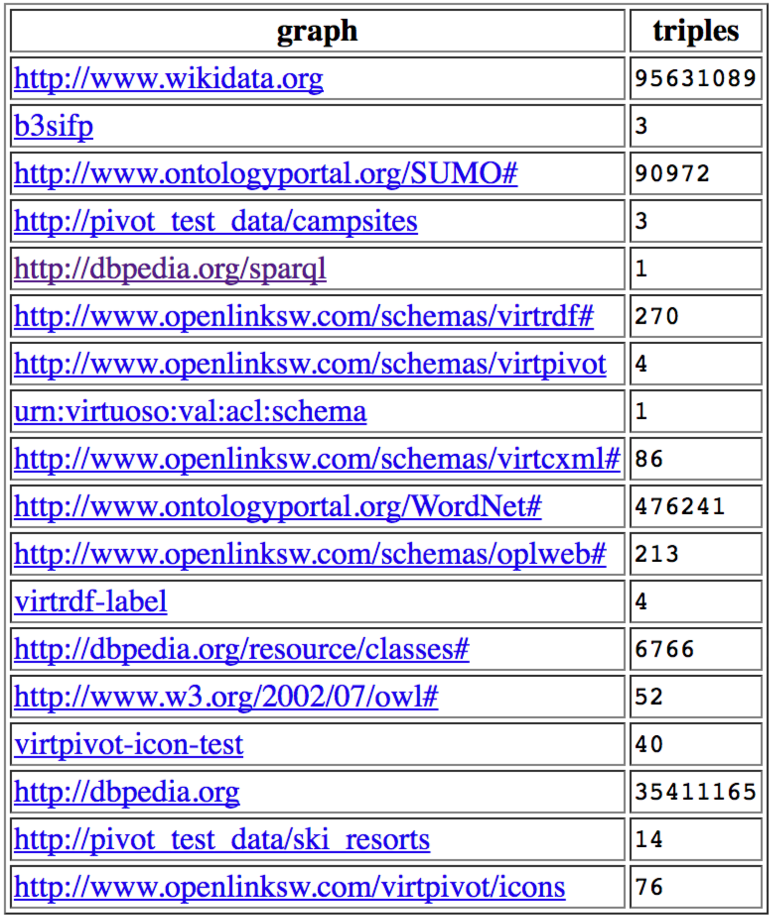
Virtuoso puts the triples in a single table

Conclusions: The role of the GRAPH clause

Conclusions about GRAPH clause:

1. is semantically ambiguous
2. offers a view of how the data is conceptually ordered but not physically
3. is a SPARQL clause that seems to not allow any type of optimization (at the moment)

DBpedia graphs and triples



graph	triples
http://www.wikidata.org	95631089
b3sifp	3
http://www.ontologyportal.org/SUMO#	90972
http://pivot_test_data/campsites	3
http://dbpedia.org/sparql	1
http://www.openlinksw.com/schemas/virtrdf#	270
http://www.openlinksw.com/schemas/virtpivot	4
urn:virtuoso:val:acl:schema	1
http://www.openlinksw.com/schemas/virtcxml#	86
http://www.ontologyportal.org/WordNet#	476241
http://www.openlinksw.com/schemas/oplweb#	213
virtrdf-label	4
http://dbpedia.org/resource/classes#	6766
http://www.w3.org/2002/07/owl#	52
virtpivot-icon-test	40
http://dbpedia.org	35411165
http://pivot_test_data/ski_resorts	14
http://www.openlinksw.com/virtpivot/icons	76

Future work

- Create a Benchmarking for federated queries with SERVICE and GRAPH
- Apply the approach to other triple stores
- Carry out semantic equivalence between open data portal and a SPARQL endpoint
- Link unique graph patterns with their pertained graphs
- Analyze why Virtuoso and Apache Jena can execute the approach

On the Role of the GRAPH Clause in the Performance of Federated SPARQL Queries

David Chaves-Fraga, Ontology Engineering Group
Universidad Politécnica de Madrid, Spain
Claudio Gutierrez, CIWS & Universidad de Chile
Oscar Corcho, OEG - UPM

dchaves@email
@dchavesf

22/10/2017
PROFILES17