GeoFedBench: A Benchmark for Federated GeoSPARQL Query Processors

Antonis Troumpoukis[1], Stasinos Konstantopoulos[1], Giannis Mouchakis[1], Nefeli Prokopaki-Kostopoulou[1], Claudia Paris[2], Lorenzo Bruzzone[2], Despina-Athanasia Pantazi[3], and Manolis Koubarakis[3]

[1] Institute and Informatics and Telecommunications, NCSR "Demokritos", Greece
{antru,konstant,gmouchakis,nefelipk}@iit.demokritos.gr
[2] Dept Information Engineering and Computer Science, University of Trento, Italy
{claudia.paris,lorenzo.bruzzone}@unitn.it
[3] National and Kapodistrian University of Athens, Greece
{dpantazi,koubarak}@di.uoa.gr

Abstract. Performance benchmarks are invaluable for evaluating and comparing federated query processing systems, but it is hard to design benchmarks that are both realistic and informative about the systems being tested. In this paper we present GeoFedBench, a benchmark that has been obtained from an actual, practical application of geospatial and linked data querying and uses GeoSPARQL constructs that challenge all phases of federated query processing. The benchmark is publicly available as part of the Kobe suite.

Keywords: Benchmarking, GeoSPARQL, Federated querying

1 Introduction and Motivation

Performance benchmarks are invaluable for evaluating and comparing systems, but designing benchmarks is subject to considerations that are difficult to satisfy simultaneously. One potential tension is the creation of a realistic benchmark that accurately reflects how the benchmarked systems will behave in real-world use cases against the design of a benchmark that is informative with respect to system characteristics we know in advance that we need to test and measure.

Given the above, we are excited to present a benchmark that has been obtained from an actual, practical application of geospatial and linked data querying. The benchmark federates a database of Earth Observation data about land usage and a database of ground observations about land usage, to search for pairs between them that simultaneously satisfy geospatial and thematic (land usage) constraints (Section 2).

Besides being extracted from a real workflow in the Earth Observation domain, the benchmark queries use GeoSPARQL constructs that challenge all

phases of federated query processing, from source selection to query planing and execution. Besides a detailed analysis of the queries, we also present emprical tests demonstrating that the benchmark is both challenging but feasible (Section 3). Finally, we recap and conclude (Section 4).

2 Use case: Validating land usage data

Detailed land usage data is crucial in many applications, ranging from formulating agricultural policy and monitoring its execution, to conducting research on climate change resilience and future food security. Land usage can be inferred from Earth Observation images or collected through self-declaration, but in either case needs to be validated against land surveys. The standard approach for this validation is to match each instance in the land survey dataset (GPS points) with the nearest land parcel (a GIS shape) and compare the crops observed in the survey against the crops declared or inferred for the matching parcel.

Although conceptually straightforward, in operational scenarios this rule can be misleading. Ground observations are geo-referenced to a point on the road adjacent to the field, which is often ambiguous in agricultural areas with several adjacent parcels; further exacerbated by GPS accuracy. However, a more sophisticated (and also computationally demanding) approach can estimate the error rate of the land usage data: for every survey point there must be at least one parcel with the same label in reasonable proximity; otherwise at least one nearby parcel is mis-labelled (although we cannot automatically infer which one).

For our benchmark, we use the *Invekos* dataset, the Austrian administration's Land Parcel Identification System with owners' self-declaration about the crops grown in each parcel, compared against the observations from the 2018 *Land use and cover area frame statistical survey (Lucas)*. Table 1 gives more details about these datasets.

Besides geospatial processing, using these datasets also introduces a data integration aspect to our benchmark. Specifically, Lucas annotations follow the *Land Cover Classification System (LCCS)* whereas Invekos follows its own codelist of 212 crop types. There is no one-to-one mapping between instance labels (e.g., Invekos *grassland* can be Lucas E10, B55, or E30, while Lucas B13 can be Invekos *spring barley* or *winter barley*).

Table 1. Dataset details

	All triples	Geospatial triples	Thematic triples
Lucas	30,379	4,325	26,054
https://esdac.jrc.ec.europa.eu/projects/lucas			
Invekos	14,036,799	2,005,257	12,031,542
https://www.data.gv.at/katalog/dataset/e21a731f-9e08-4dd3-b9e5-cd460438a5d9			

3 Benchmark queries

In order to estimate the reliability of the Invekos dataset, we used queries that, for each given Lucas instance, check if: (Q1) the closest Invekos instance is under 10 meters away and their crop labels match; (Q2) the closest Invekos instance is under 10 meters away and their crop labels do *not* match; or (Q3) there is no Invekos instance within 10 meters.

Since geo-linked data vocabularies link instances with a geometry object (which then has as an attribute the actual shape), these queries (and geoSPARQL queries in general) challenge `FILTER` optimizers because it presents them with comparisons between variable groundings (as opposed to constant values), and because these comparisons are non-standard extensions (the geospatial extensions of GeoSPARQL).

In most benchmarks, filters are either not present at all [LUBM, 2] or only have unary functions or comparisons against constants [FedBench, 5] that can always be pushed into one data source. LargeRDFBench [4] includes multi-variable filters that compare values from different repositories, challenging the optimizer to select the correct strategy: to fetch the left-hand side values and push the filter into the right-hand side source or to fetch both sides and apply the filter locally. Both approaches are valid, but can vary dramatically in terms of efficiency. Our benchmark presents the same challenge in a geospatial context; the federator is tested not only on correctly selecting the best strategy but also on the efficiency of its local implementation of the GeoSPARQL extension.

Properties of standard vocabularies, which can appear possibly in all sources of a federation, present another challenge in the efficient evaluation of a query. When evaluating a triple pattern that contains a property such as `rdf:type` or `owl:sameAs` the source selector is prone to overestimate the set of relevant sources, thus increasing both network traffic and the overall query processing time. Current benchmarks already contain such commonly used properties. But GeoFedBench stresses source selections more on this direction by exploiting a query characteristic that appears frequently in Geospatial data; a resource `?x` is linked with its geometry representation `?wkt` using *chains of known properties* of the form `?x geo:hasGeometry ?g . ?g geo:asWKT ?wkt`, where all members of the chain usually appear in the same dataset. The federation engine is tested on distinguishing which geospatial triple patterns refer to which dataset, thus avoiding to fetch redundant bindings for the variable in the middle of the chain.

Finally, the complex nature of our queries challenges query planning. Current benchmarks usually contain simple queries consisting only joins between triple

Table 2. Query prossesing time (msec) of the benchmark queries for three different Lucas instances.

Lucas point	Q1	Q2	Q3
1	157,209	146,325	152,999
2	143,751	145,049	139,447
3	156,127	152,502	136,762

patterns and `FILTER` operations, or some additional operators such as `UNION`, `ORDER`, `LIMIT`, etc. In GeoFedBench, Q1 and Q2 use a *subquery* for discovering the *closest* Invekos instance. Also, Q2 and Q3 use *negation*, in the form of the `FILTER NOT EXISTS` operator to check that there does not exist and matching Invekos instance. Both subqueries and negation are not present in any of the currently existing federated SPARQL benchmarks.

To demonstrate that the benchmark is feasible but challenging, we tested on Semagrow [1], to the best our knowledge the SPARQL federation engine that supports geospatial operators. The datasets are served by Strabon geospatial RDF stores [3]. Table 2 gives the query execution time for three runs of each query, where each run grounds the query with a different Lucas point.

4 Conclusions

We presented GeoFedBench, a benchmark for federated geospatial query processing. GeoFedBench is based in openly available datasets and queries challenge all phases of federated query processing. The benchmark is distributed as part of the benchmark suite of the KOBE Open Benchmark Engine, available from https://github.com/semagrow/kobe

Acknowledgement

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825258. Please see http://earthanalytics.eu for more details.

Bibliography

[1] Charalambidis, A., Troumpoukis, A., Konstantopoulos, S.: SemaGrow: Optimizing federated SPARQL queries. In: Proceedings of the 11th International Conference on Semantic Systems (SEMANTiCS 2015), Vienna, Austria, 16–17 September 2015 (2015)

[2] Guo, Y., Pan, Z., Heflin, J.: LUBM: a benchmark for OWL knowledge base systems. Web Semantics 3(2) (Jul 2005)

[3] Kyzirakos, K., Karpathiotakis, M., Koubarakis, M.: Strabon: A semantic geospatial DBMS. In: P. Cudré-Mauroux et al. (eds.) Proceedings of ISWC 2012, Boston, MA, USA, 11-15 November 2012. LNCS vol. 7649, Springer (2012), https://doi.org/10.1007/978-3-642-35176-1_19

[4] Saleem, M., Hasnain, A., Ngomo, A.N.: LargeRdfBench: A billion triples benchmark for SPARQL endpoint federation. J. Web Semant. 48, 85–125 (2018), https://doi.org/10.1016/j.websem.2017.12.005

[5] Schmidt, M., Görlitz, O., Haase, P., Ladwig, G., Schwarte, A., Tran, T.: FedBench: A benchmark suite for federated semantic data query processing. In: Proceedings ISWC 2011, Bonn, Germany, 23-27 October 2011. LNCS vol. 7031. Springer (2011), https://doi.org/10.1007/978-3-642-25073-6_37