




Article

Assessment and Benchmarking of Spatially Enabled RDF Stores for the Next Generation of Spatial Data Infrastructure

Weiming Huang ^{1,*} , Syed Amir Raza ¹, Oleg Mirzov ^{1,2}  and Lars Harrie ^{1,2} 

¹ Department of Physical Geography and Ecosystem Science, Lund University, 223 62 Lund, Sweden

² ICOS Carbon Portal, Lund University, 223 62 Lund, Sweden

* Correspondence: weiming.huang@nateko.lu.se

Received: 27 May 2019; Accepted: 12 July 2019; Published: 19 July 2019



Abstract: Geospatial information is indispensable for various real-world applications and is thus a prominent part of today's data science landscape. Geospatial data is primarily maintained and disseminated through spatial data infrastructures (SDIs). However, current SDIs are facing challenges in terms of data integration and semantic heterogeneity because of their partially siloed data organization. In this context, linked data provides a promising means to unravel these challenges, and it is seen as one of the key factors moving SDIs toward the next generation. In this study, we investigate the technical environment of the support for geospatial linked data by assessing and benchmarking some popular and well-known spatially enabled RDF stores (RDF4J, GeoSPARQL-Jena, Virtuoso, Stardog, and GraphDB), with a focus on GeoSPARQL compliance and query performance. The tests were performed in two different scenarios. In the first scenario, geospatial data forms a part of a large-scale data infrastructure and is integrated with other types of data. In this scenario, we used ICOS Carbon Portal's metadata—a real-world Earth Science linked data infrastructure. In the second scenario, we benchmarked the RDF stores in a dedicated SDI environment that contains purely geospatial data, and we used geospatial datasets with both crowd-sourced and authoritative data (the same test data used in a previous benchmark study, the Geographica benchmark). The assessment and benchmarking results demonstrate that the GeoSPARQL compliance of the RDF stores has encouragingly advanced in the last several years. The query performances are generally acceptable, and spatial indexing is imperative when handling a large number of geospatial objects. Nevertheless, query correctness remains a challenge for cross-database interoperability. In conclusion, the results indicate that the spatial capacity of the RDF stores has become increasingly mature, which could benefit the development of future SDIs.

Keywords: linked data benchmark; RDF stores; geospatial data; GeoSPARQL; spatial data infrastructure

1. Introduction

Geospatial information is indispensable for spatially informed decision-making and analyses and is thereby a prominent part of today's data science landscape. Significant progress in geospatial data availability and sharing has been achieved as a result of the development of spatial data infrastructures (SDIs) that aim to make geospatial data available for the benefit of the economy and the society [1]. In Europe, the INSPIRE directive—a legal framework and standardization body for SDI development—sets the data specifications, and it mandates its member states to provide data mainly using Open Geospatial Consortium (OGC) web services [2].

Despite the significant progress, SDIs still face a number of limitations, especially in terms of discovery, reuse, and integration of the data. SDIs have partially achieved dissolving environmental

and geospatial data held in silos, but the data is still largely isolated from other information domains [3]. For example, the OGC web features service (WFS) can make geospatial data available through its data query protocol, yet such data cannot be discovered by search engines or, more importantly, linked by other data resources. This makes the data lying in the so-called deep web [4].

Today's geospatial data is available and used not only in dedicated SDIs but also in various general data infrastructures/projects that are not dedicated to geospatial data. One open data example is the general-purpose knowledge graph DBpedia (<https://wiki.dbpedia.org/>), which has a large number of geospatial objects. In other words, geospatial data has become a part of today's big data landscape; thus, siloed data management and delivery should be revisited [5]. This is also in line with the development and vision of open SDIs, which highlight the integration and harmonization with other data [6].

Another significant issue in SDIs is semantic heterogeneity, which is an impediment to integrating multi-source geospatial data and fusing geospatial data with other types of data, as the semantics of metadata, schemas, and data content are not usually harmonized for multi-source geospatial data or with other types of data [7].

Semantic Web technologies, particularly the parts relevant to linked data, provide a promising way to resolve the aforementioned limitations. Linked data is built around a set of data publishing best practices and facilitates data access, interlinking, and integration on the web. A recent survey conducted in 2018 by EuroSDR demonstrated that linked data is seen as one of the most important research issues and key factors moving SDIs toward the next generation [8]. Linked data was also voted one of the most important SDI research topics during the AGILE 2018 workshop 'SDI research and strategies towards 2030' [9]. An increasing amount of geospatial data has been delivered as linked data on the web and has become part of the linked open data (LOD) cloud (<https://lod-cloud.net/>).

Linked data is organized in the data model Resource Description Framework (RDF) [10], which is a generic graph-based data model that describes entities and relations. Linked data is also built upon formally defined ontologies, providing the means to define the concepts and relations in data, in order to make explicit any underlying assumptions regarding the data, and make it easier to understand and reuse the data. In practice, linked data needs to be managed, stored, and delivered by utilizing RDF stores (also known as triplestores), which are databases for storing and retrieving RDF data (linked data) through semantic queries (SPARQL queries [11]). The OGC extended SPARQL to develop the query language for geospatial linked data—GeoSPARQL, which comprises a lightweight vocabulary to represent and query geospatial data [12]. The number of spatially enabled RDF stores (RDF stores that handle geospatial queries) is currently growing, and their compliance with GeoSPARQL has progressed. Therefore, there is a need to survey the status of spatially enabled RDF stores in terms of both geospatial query performance and GeoSPARQL compliance.

The aim of this study is to assess and benchmark several well-known and popular spatially enabled RDF stores for potential use in future SDIs and the geospatial linked data community at large (see supplementary files). In this context, we performed benchmarking in two different scenarios in future SDIs. The first scenario is one in which geospatial data plays an important role in and constitutes a part of a large data infrastructure; here, the focus is on the integration of geospatial data with other data. Two issues must be resolved here: the ontology of the geospatial components of the data should conform to the GeoSPARQL standard, and the RDF stores should be able to efficiently perform geospatial queries on a large volume of data that is a mixture of geospatial and other data. To evaluate the first scenario, we used data from the Integrated Carbon Observation System (ICOS) carbon portal (ICOS CP) [13]—a large-scale Earth Science scientific data infrastructure. The second scenario illustrates a dedicated SDI with purely spatial data; for this case, we used test datasets from Geographica, a previous geospatial benchmark for RDF stores [14]. These datasets include crowd-sourced (e.g., GeoNames, DBpedia, and LinkedGeoData) and authoritative geospatial data.

Following this introduction, the background and related work are presented in Section 2. The data used in this study is illustrated in Section 3, including the ICOS CP's ontology design. Section 4 describes the assessment and benchmarking methodology, and the results are presented in Section 5 (for

qualitative evaluation) and Section 6 (for quantitative evaluation). The paper ends with a discussion (Section 7) and conclusions (Section 8).

2. Background and Related Work

2.1. Geospatial Semantic Web and Linked Data

The Semantic Web is a common framework that allows data to be shared and reused across application, enterprise, and community boundaries [15]. In order to make the Semantic Web a reality, it is important to make a huge amount of data on the web available with recommended best practices for exposing, sharing, and connecting pieces of data, information, and knowledge. These best practices, as well as the delivered data, are also referred to as linked data. At the core of the linked data principles are the ideas of globally unique identifiers, i.e., Uniform Resource Identifiers (URIs) for data elements and a universal graph data model Resource Description Framework (RDF). By reusing the addressing system used for web pages, one can uniquely identify and link to data elements and datasets anywhere on the web [16]. The appreciation of Semantic Web technologies and linked data has increased considerably in the geospatial domain in the last decade, and they have fostered a promising approach to connecting SDIs with mainstream IT to augment the application of geospatial data [3]. Semantic Web technologies, especially linked data, provide a promising means to address some long-standing challenges in the geospatial domain, e.g., data integration (e.g., [3]) and knowledge formalization (e.g., [17]).

Pilot studies have been performed releasing INSPIRE-compliant data as linked data, and draft guidelines and vocabularies have been developed [18]. The development of INSPIRE linked data's URIs leveraged previous work on the standardization of unique identifiers for geospatial objects [19]. In the meantime, an increasing amount of geospatial data has been delivered as linked data, mainly by governmental agencies and large-scale data infrastructures [20]. The UK is a pioneer to this end; Ordnance Survey, Great Britain's national mapping agency (NMA), released several geospatial datasets as linked data nearly a decade ago [21]. However, the data relied on unstandardized methods to represent data semantics and thus lacked usability. In the Netherlands, Kadaster delivered several key geospatial datasets, e.g., building data and address data, as linked data on the web, together with other governmental open data, e.g., statistical data [22]. In Finland, the National Land Survey piloted the delivery of geographic name data, authoritative data, and building data as linked data [23]. In Norway, Kartverket also released some geospatial datasets as linked data [24]. A recent report summarized and reflected on the development of geospatial linked data in the Netherlands, Finland, Norway, and Spain. The fact that different projects use different RDF stores also renders the aim of this study necessary [25]. In the US, several geospatial linked data projects have been conducted: a pilot of design and development of linked data from The National Map was performed [26]; the Geographic Names Information System was served as linked data, and its geospatial visualization was enabled [20]; the GeoLink knowledge graph was published following linked data principles and served through a SPARQL endpoint, including Earth Science information captured by oceanographic cruises, physical sample metadata, etc. [27]. Along with these linked data, development endeavors from authorities, crowd-sourcing projects have also produced several geospatial linked datasets, and some of them are serving as central hubs of the LOD cloud, e.g., GeoNames (<https://www.geonames.org/>) and LinkedGeoData (a linked data distribution of OpenStreetMap [28]). Moreover, van den Brink et al. [29] proposed the best practice of delivering geospatial linked data, and they bridged the OGC web services and the Semantic Web. In the Earth Science domain, there have also been several discussions about how to utilize linked data for data integration and discovery (e.g., [30]).

Semantic Web technologies and linked data have also been utilized in a number of studies in the geospatial domain. The studies on this subject span several research areas, e.g., geoprocessing, information retrieval, and visualization. For example, Hofer et al. [31] developed a knowledge base to support the composition of geoprocessing workflows with ontologies and Semantic Web rule language (SWRL). Keßler et al. [32] leveraged linked data, ontologies, and SWRL rules for geospatial information

retrieval with context awareness. Wiemann and Bernard [33] used linked data for data integration in the environment of SDIs. Huang et al. [34] leveraged linked data and ontologies to realize the relative positioning of geospatial data, thus enabling geometrically self-adapting web maps. Huang and Harrie [17] used linked data, ontologies, and semantic rules to realize knowledge-based visualization of geospatial data, thereby formalizing some visualization knowledge on the aspects of cartographic scale, data portrayal, and geometry source. To realize the potentials revealed by the above studies (e.g., the use of ontological reasoning, rule-based reasoning, and spatial operations), we need RDF stores with capabilities such as semantic query, semantic reasoning, and geospatial query. Therefore, we used these capabilities in this study as part of the RDF store selection criteria (cf. Section 4.1).

2.2. Assessment and Benchmarking of Spatially Enabled RDF Stores

As the Semantic Web evolved into the mainstream of the web and has been adopted in many scientific domains (e.g., life sciences, geosciences), assessments and benchmarks of RDF stores have been abundant, mainly on synthetic and artificial test datasets. Popular benchmarks include, in chronological order, the Lehigh University Benchmark (LUBM) [35], the SPARQL performance benchmark (SP²Bench) [36], and the Berlin SPARQL Benchmark (BSBM) [37]. The DBpedia SPARQL benchmark (DBSB) [38] is a popular benchmark used for real-world linked data and queries (the queries are extracted from actual server logs). However, these benchmarks are mainly for common-use data and data from other domains, not geospatial data and queries. In addition, benchmarks based on synthetic data have been criticized because they have very little in common with the needs of real application domains [39].

For the assessment of spatially enabled RDF stores, in which an even higher level of complexity arises [40,41], Kolas [42] proposed and performed a benchmark for the geospatial query capacity of RDF stores; however, since it was proposed before the standardization of GeoSPARQL, not much from that work can be applied to today's developments. Battle and Kolas [43] demonstrated the geospatial capacity of Parliament and successfully ran a number of GeoSPARQL-compliant queries. Garbis et al. [14] presented the benchmark Geographica to assess several spatially enabled RDF stores in which spatial queries were written in both GeoSPARQL and stSPARQL (the spatiotemporal query language in the RDF store Strabon). In that benchmark, three RDF stores were evaluated, i.e., Strabon, uSeekM, and Parliament, in a micro-benchmark and a macro-benchmark. The micro-benchmark aims to test the efficiency of primitive spatial functions in spatially enabled RDF stores; the macro-benchmark aims to test the performance of the stores in some certain application scenarios, e.g., reverse geocoding, map search, etc. This benchmark's datasets and queries have been published online (<http://geographica.di.uoa.gr/>), and the benchmark was based on both real-world geospatial data (e.g., LinkedGeoData) and synthetic data. The GeoKnow project, which dealt with geospatial Semantic Web and linked data, released a thorough survey and evaluation of spatially enabled RDF stores, with a partial focus on GeoSPARQL compliance [44]. The stores evaluated in GeoKnow include Virtuoso, Parliament, OWLIM, uSeekM, and Strabon, as well as spatially enabled relational databases, i.e., Oracle Spatial and PostgreSQL with PostGIS extension. Bellini and Nesi [45] assessed several well-known RDF stores, including Virtuoso, GraphDB, Oracle, and Stardog, for semantically enabled smart city services. The geospatial capacity of these RDF stores was one of the focuses of this study, as smart city services also have the need for capabilities such as temporal data query. The benchmark was based on the Florence Smart City model; the used datasets and tools are available online. These benchmarks clearly demonstrated the sparse support for spatial operations in RDF stores, and the RDF stores supporting GeoSPARQL were very few. Specifically, many RDF stores, e.g., Virtuoso, used their own syntaxes for geospatial queries rather than GeoSPARQL, and most RDF stores supporting GeoSPARQL queries were developed in academic environments, e.g., Parliament. Furthermore, the query performance was generally unsatisfactory, which also undermined the usability of these very few spatially enabled RDF stores.

The abovementioned previous works provide useful grounds for this study to evaluate the geospatial query capacity of RDF stores for future SDIs and for the geospatial linked data community at large. However, these previous studies have some limitations. First, the results are now mostly outdated, as the status of the tested RDF stores have changed considerably: some of them have developed with more advanced support for geospatial queries and increased GeoSPARQL compliance, and some of them have become obsolete and are rarely used. Second, the assessments and benchmarks targeting geospatial query (i.e., Geographica and GeoKnow benchmarks) depended on either synthetic data or purely geospatial data (in which nearly all the data objects have geometric information and are involved in spatial indexing/search). Our first test scenario, which uses data from ICOS CP, is, however, an Earth Science data infrastructure with a portion of geospatial data, which is more in line with the current role of geospatial data in large data infrastructures (open SDI). In addition, we provide a reproducible benchmark with deliverables that others can use to assess the RDF stores on their own datasets. Additionally, one shortcoming of previous spatially enabled RDF stores' benchmarking works is that they fully focused on evaluating the query performance (response time), but they did not assess the correctness of the returned results. In this paper, we assess query correctness in the first scenario.

3. Benchmarking Datasets

3.1. ICOS Carbon Portal Metadata

In the first scenario, we used data from ICOS CP (see supplementary files). ICOS is a Pan-European research infrastructure that currently has 12 member countries and a legal status of European Research Infrastructure Consortium (ERIC) (https://ec.europa.eu/info/research-and-innovation/strategy/european-research-infrastructures/eric_en). It is a European measurement system for high-quality and precision greenhouse gas observations and environmental monitoring. Currently, there are 135 measurement stations (including co-located ones), with 33 atmosphere stations, 81 ecosystem stations, and 21 ocean stations (Figure 1 shows the geographic locations of the stations).

ICOS CP is the data portal that provides free and open access to all ICOS datasets. ICOS data products include quality-controlled observational data, elaborated (model) products, and synthesis reports, which is material for policymakers. The users of ICOS CP span various domains, e.g., (Earth Science) researchers, education users, policymakers, and stakeholders in the negotiation of carbon reduction policies. ICOS produces around 25–30 TB of sensor data per year, together with about 1 GB of processed data products and 5–20 TB of elaborated data products. Additionally, as ICOS CP has become a well-recognized data sharing and distribution platform, some other data initiatives and producers, e.g., SOCAT (<https://www.socat.info/>), have also contributed by publishing their data through ICOS CP. The observation data at ICOS CP is linked to georeferenced locations. The atmospheric and ecosystem observations are connected to the coordinates of the measurement stations. For the ocean data, ship trajectories are stored as lists of XY coordinate pairs. The huge amount of data delivered and the complex organizational structure and responsibility raise the importance of data cataloging and discovery.

ICOS CP is an active practitioner of the FAIR principles, which aim to make data Findable, Accessible, Interoperable, and Reusable [46,47]. In this context, ICOS CP has adopted linked data for delivering and publishing all its metadata (including metadata for ICOS data and other data harvested by ICOS CP, e.g., SOCAT data) to make such data more discoverable. The metadata is available through, among others, a SPARQL endpoint (<https://meta.icos-cp.eu/sparqlclient>). Geospatial data forms a part of the ICOS CP metadata. As the size of ICOS CP metadata is constantly growing because observational data is continually ingested, query performance will become a notable issue. To accelerate the spatial search of ocean data, each trajectory is simplified into a line string or a polygon (concave hull of the trajectory) containing a maximum of 20 coordinate pairs (by an in-house developed

streaming algorithm that extends the algorithm from [48]). These simplified geometries are stored in the ICOS CP metadata.

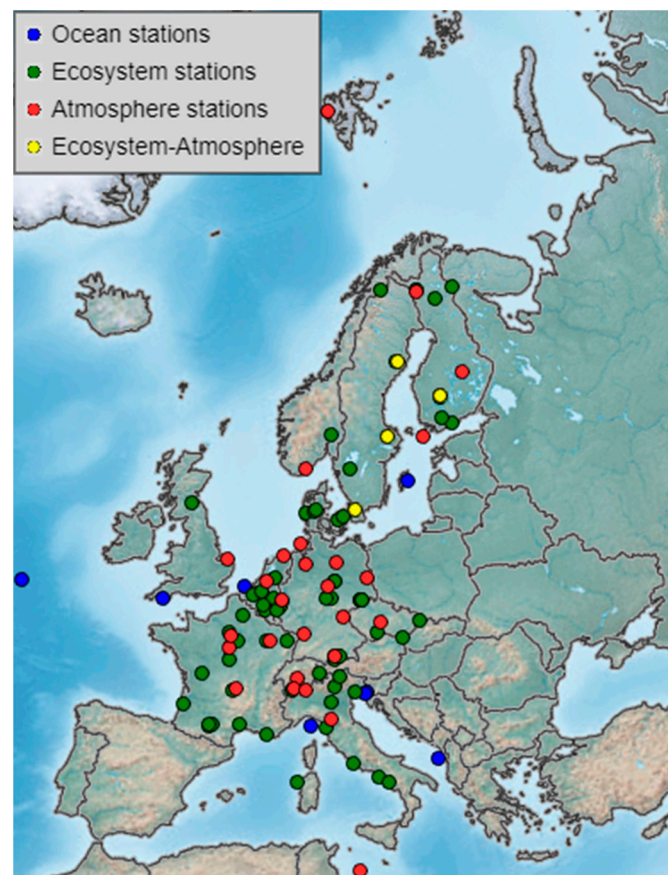


Figure 1. Geographic locations of Integrated Carbon Observation System (ICOS) measurement stations.

The linked data implementation is built upon a set of ontologies for different scopes of the data portal responsibility. Among them, the most important ontology is the ICOS CP metadata ontology (with the prefix *cpmeta* (<https://meta.icos-cp.eu/ontologies/cpmeta/>)). The ICOS CP metadata ontology relies on and has strong interoperability with some W3C standard ontologies, e.g., W3C PROV ontology [49] and W3C organization ontology [50]. For the details of ICOS CP ontologies, please refer to its GitHub repository (<https://github.com/ICOS-Carbon-Portal/meta/tree/master/src/main/resources/owl>) or the online description (<http://static.icos-cp.eu/share/slides/dataServiceWorkshop/#/>).

In the ICOS CP metadata ontology, the instances of the class *DataObject* can be associated with the instances of the class *SpatialCoverage*, and the instances of *SpatialCoverage* can be associated with the serialization of the corresponding geometries (Figure 2 demonstrates a part of the ICOS metadata ontology that is relevant to spatial information.). Currently, the ICOS metadata ontology is not GeoSPARQL-compliant (the GeoSPARQL classes are not introduced into ICOS metadata ontology, and the geometries are serialized in GeoJSON, which is not supported by GeoSPARQL). To support geospatial (GeoSPARQL) queries, we redesigned the ontology to accomplish GeoSPARQL compliance, as illustrated in Figure 2 (we use *geo* for the prefix of GeoSPARQL). That is, we built an inheritance relation in which *SpatialCoverage* is a subclass of *geo:Geometry*, and the instances can thereby be associated with the geometries in Well-Known Text (WKT) to enable GeoSPARQL-compliant geospatial queries. Afterward, we transformed all the geometries from GeoJSON to WKT using several SPARQL CONSTRUCT queries (the queries are available online at <https://github.com/RightBank/Benchmarking-spatially-enabled-RDF-stores/tree/master/TransformationSPARQLQueries>).

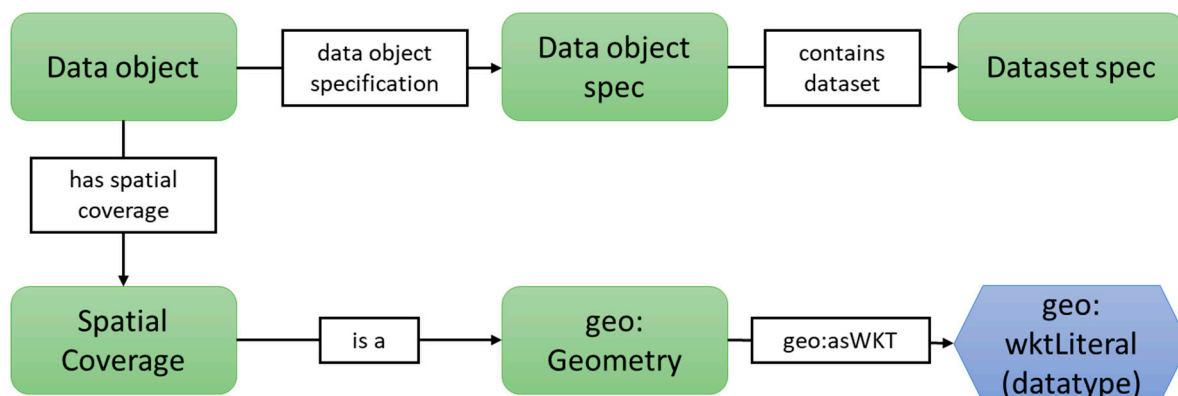


Figure 2. Geospatial part of the ICOS metadata ontology. The concepts and relations without prefix annotation are from ICOS metadata ontology.

The test data for RDF store assessment and benchmarking is the entire set of metadata of ICOS CP, which has 2,194,299 RDF statements as of 18 March 2019. The dataset has been published online [51]. Among the data, there are 1068 spatial objects (88 polygons, 853 polylines, and 127 points). We believe that this situation mirrors the current development of geospatial data that it forms a part of a large-scale information infrastructure. Therefore, the results of this study can also be used as a reference for other linked data implementations with similar situations. Technically, extracting and querying on relevant geospatial data from mass data, including relevant and irrelevant data, is costlier for query planners in the RDF stores than merely operating without query-irrelevant data.

The most important geospatial query requirement for ICOS CP is to enable users to directly spatially select different types of data objects (e.g., measurement trajectories) in user-defined geometric ranges, which could be a simple rectangle or an arbitrary complex polygon that is drawn by the users. In this context, the topological relations within, intersects, and overlaps are useful, but we also would like to support other geospatial functions available in GeoSPARQL, such as buffer, disjoint, and crosses, for specific user needs and requirements. Therefore, we tested the available spatial functions in some RDF stores that are not restricted to the functions for spatial selections (cf. Section 4.2).

3.2. Geographica Benchmarking Datasets

For the second scenario, in which the benchmarking is performed on a large amount of purely geospatial data, we used real-world datasets from the Geographica benchmark. Six real-world geospatial datasets in RDF were used: DBpedia, GeoNames, road networks and rivers from Greece, the Greek Administrative Geography dataset, the CORINE Land Use/Land Cover dataset, and wildfire hotspots from the National Observatory of Athens. The geographic coverages of the six datasets are in Greece. The six datasets contain more than 30,000 points, 12,000 polylines, and 104,000 polygons. Details of the datasets are provided in [14] and its online repository (<http://geographica.di.uoa.gr/>).

4. Evaluation Methodology

The evaluation of spatially enabled RDF stores was carried out in two stages. In the first stage, we selected the RDF stores using a set of criteria and deeply analyzed the geospatial features provided by the selected stores (e.g., GeoSPARQL compliance, licensing, spatial indexing, etc.). The successive second stage applied a benchmark to the RDF stores in the above-discussed two scenarios. It is based on a set of SPARQL queries that are capable of testing the geospatial query performance of the stores.

4.1. RDF Store Selection and Analysis

The selection of the tested RDF stores is based on the needs both of large-scale information infrastructures (ICOS CP in this case) and dedicated SDIs. First, general selection criteria were applied:

- The RDF store should be popular, well-known, and actively supported by a community or backed by a commercial vendor.
- The RDF store should support W3C standards, e.g., SPARQL 1.1.
- The RDF store should support semantic reasoning, which can be either triple materialization at load time or at query time (query rewriting), and the widely used reasoning types should be supported (e.g., RDFS, OWL, OWL2, OWL2-DL, etc.). Additionally, rule-based reasoning should be supported.
- The RDF store should have geospatial query capacity, preferably with GeoSPARQL support and compliance.

On the basis of these criteria, a pre-selection was made. The final selection was then based on a qualitative analysis of the pre-selected RDF stores by reading the documentation (we contacted the vendor for Stardog, as we could not find information about its spatial index technique in its documentation). The key aspects of this analysis include the following:

- Software components, architecture, deployment, and licensing;
- The means of data loading, query, and management;
- Utilization of software components from other solutions (e.g., if it is based on open-source frameworks);
- Supported semantic reasoning types;
- Geospatial query capacity and GeoSPARQL compliance;
- The employment of spatial indexing for geospatial data and the types of indexing;
- The popularity of the RDF stores is partially consulted from DB-Engines ranking (<https://db-engines.com/en/ranking/rdf+store>).

Through the qualitative analysis, not only can we choose the evaluated RDF stores in our work, but we can also obtain an up-to-date view of the popular RDF stores, especially to gain insight concerning the recent development of spatially enabled RDF stores and their GeoSPARQL compliance.

4.2. Performance Benchmark of Geospatial Query in RDF Stores

In this study, we reused and tailored the micro-benchmark from the Geographica benchmark [14] to evaluate the RDF stores. The micro-benchmark from Geographica aims to test the efficiency of primitive spatial functions in spatially enabled RDF stores. Simple SPARQL queries that consist of one or two triple patterns and a spatial function were used as benchmark queries. This benchmark includes non-topological geometric construction, simple spatial selections, and more complex operations (e.g., spatial join). In the first scenario, we tailored the benchmark queries for ICOS CP metadata; a brief description of the tailored queries can be found in Table 1. For the second scenario, we adopted the original query set from Geographica [14]. In addition, in both scenarios, *Q6* (area calculation), *Q28* (extension constructing), and *Q29* (union constructing) were removed because these functions are not supported by GeoSPARQL and seldom supported by RDF stores. *Q14* (spatial within function to real-time constructed buffers) was also removed, as this query is semantically equivalent to *Q15* but more computationally expensive than *Q15* [14], and this type of nested spatial function is not always supported by RDF stores.

In our benchmark, we first warmed up the RDF stores with warm-up SPARQL queries in order to get the benchmark systems under normal working conditions, as the query performance in a cold state is often unstable and unpredictably low in the beginning because of factors such as the initial interpretation and compilation of codes. The warm-up queries are disjoint from the actual benchmark queries (cf. Table 1), and they are taken from the pre-defined queries at ICOS CP's SPARQL endpoint.

Table 1. Benchmark queries for spatially enabled Resource Description Framework (RDF) stores in the first scenario with Integrated Carbon Observation System carbon portal (ICOS CP) metadata. Q1–Q5 are non-topological construct functions, Q7–Q17 (excluding Q14) are spatial selection queries, and Q18–Q27 are spatial join queries.

	Operation	Query Description
Q1	Boundary	Construct boundary for each polygon
Q2	Envelope	Construct envelope for each polygon
Q3	Convex Hull	Construct convex hull for each polygon
Q4	Buffer	Construct buffer for each line string (polyline)
Q5	Buffer	Construct buffer for each polygon
Q7	Equals	Find all line strings that are spatially equal to a given line string
Q8	Equals	Find all polygons that are spatially equal to a given polygon
Q9	Intersect	Find all line strings that intersect with a given Polygon
Q10	Intersect	Find all polygons that intersect with a given polygon
Q11	Overlaps	Find all polygons that overlap a given polygon
Q12	Crosses	Find all line strings that cross a given line string
Q13	Within Polygon	Find all points that are spatially within a given polygon
Q15	Near a Point	Find all points that are within a fixed distance to a given point
Q16	Disjoint	Find all points that are disjoint from a given polygon
Q17	Disjoint	Find all line strings that are disjoint from a given polygon
Q18	Equals	Find point-to-point equality among all the points
Q19	Intersects	Find all points and lines that intersect with each other
Q20	Intersects	Find all points and polygons that intersect with each other
Q21	Intersects	Find all line strings and polygons that intersect with each other
Q22	Within	Find all points and polygons where the point lies inside the polygon
Q23	Within	Find all line strings, polygons where the line string lies inside the polygon
Q24	Within	Find all pairs of polygons where one polygon is within the other
Q25	Crosses	Find all line strings, polygons where the line string crosses the polygon
Q26	Touches	Find all pairs of polygons where the polygons touch each other
Q27	Overlaps	Find all pairs of polygons where the polygons overlap each other

4.3. Implementation—Reusable Benchmark Deliverables

The benchmarking of the RDF stores was implemented in Java. We encapsulated the SPARQL queries and the codes interoperating with the underlying RDF stores in executable Jar (Java archive) packages that can be directly run with Java Runtime Environment (JRE). The delivered Jar packages request the location of data source, warm-up query iteration times, and benchmark query iteration times. The deliverable programs and source codes (including the benchmark queries) are available online at <https://github.com/RightBank/Benchmarking-spatially-enabled-RDF-stores>.

After benchmarking, text files were generated with comprehensive information regarding data loading time, the execution time of each query in each iteration, and the query results (including resulted object numbers and the resulted features—mainly their geometries). The query execution time refers to the time elapsed between the point a query is sent to the RDF store and the point the query results are completely returned to the benchmark systems. The benchmark systems use the RDF stores in an embedded mode whenever possible.

5. Results of RDF Store Selection and Analysis

Using the selection criteria for testing RDF stores for this work, we thoroughly investigated a number of RDF stores, and we ultimately selected the following RDF stores for evaluation.

1. RDF4J 2.4.2: an open-source Java RDF framework under the license of Eclipse Distribution License, v1.0, formerly known as Sesame. It supports parsing, storing, inferencing, and querying RDF data. It supports SPARQL 1.1 and both ontological and rule-based reasoning. Inferred statements are materialized. It supports geospatial query in GeoSPARQL, and its spatial queries can be performed without spatial indexing or with Lucene Spatial (currently, Lucene Spatial in

- RDF4J results in errors). RDF4J can be used as an RDF store or a library that communicates and operates with many third-party storage solutions (RDF stores).
2. Jena 3.9.0 + GeoSPARQL-Jena 1.0.3: an open-source Java framework for building Semantic Web and linked data applications. It supports SPARQL 1.1 and both ontological and rule-based reasoning. It provides both RDF API, which manipulates RDF data, and TDB, an RDF store solution. Jena is one of the most widely adopted RDF frameworks in various research and production projects. Jena itself has very limited spatial query capacity and does not support GeoSPARQL. The recently developed open-source plugin GeoSPARQL-Jena (<https://github.com/galbiston/geosparql-jena>) provides fully GeoSPARQL-compliant spatial query capacity with a custom spatial indexing technique. Both Jena and GeoSAPRQL-Jena are under Apache License 2.0.
 3. Virtuoso Enterprise 8.2: one of the most well-known RDF stores because of its adoption by DBpedia. It supports SPARQL 1.1 and ontological and rule-based reasoning. The reasoning is performed by query rewriting, so inferred statements are not materialized. It has had geospatial query support for a few years, and it started to support GeoSPARQL in its commercial version in 2018 (it also claimed to support GeoSPARQL in its open-source edition, but, to date, no release has appeared, so we chose to use the commercial version). It uses R-tree as its spatial indexing technique. A proprietary license for the commercial edition and a GPL 2 license for the open-source version are used.
 4. Stardog 6.0.1: a commercial knowledge graph product that supports parsing, storing, inferencing, and querying RDF data. It supports SPARQL 1.1 and both ontological and rule-based reasoning with a query rewriting strategy. It supports a few GeoSPARQL query functions with Lucene Spatial for spatial indexing. It is actively supported by a commercial company and uses proprietary licenses.
 5. GraphDB 8.8.0: a linked data platform built upon RDF4J. It is a commercial solution that provides support for SPARQL 1.1 and ontological and rule-based reasoning. It supports GeoSPARQL with spatial indexing of Lucene Spatial (specifically, quad-prefix-tree and geohash-prefix-tree). It utilizes different strategies for handling queries with and without using a spatial index. GraphDB is under proprietary licenses.

The rationale for not selecting the formerly assessed and benchmarked spatially enabled RDF stores Parliament, Strabon, and uSeekM is that they are currently not actively supported by the community, and some of them have limited capacity for reasoning, particularly rule-based reasoning. That is, we only evaluated fully fledged and popular RDF stores with spatial query support.

The qualitative analysis of the selected stores resulted in a cross-store qualitative comparison. Table 2 compiles the results of qualitative analysis with a focus on spatial query capacity and GeoSPARQL compliance. The storage solutions adopted by the RDF stores are mainly divisible into two types: native (designed from scratch) and RDBMS-based (based on an existing relational database management system). Four of the five tested stores utilize native solutions for storage; only Virtuoso relies on an underlying RDBMS. All tested RDF stores support spatial operations for geometries serialized in WKT; only GraphDB and GeoSPARQL-Jena support GML as well. RDF4J, GeoSPARQL-Jena, Virtuoso, and GraphDB currently provide full support for GeoSPARQL functions (the queries with spatial relations in the simple features relation family), including non-topological construct functions (*Q1–Q5* in Table 1), spatial selection functions (*Q7–Q17* in Table 1), and spatial join functions (*Q18–Q27* in Table 1). Stardog only supports the functions that find the relations within, nearby, intersect, contains, disjoint, and equal, and it uses its own spatial query syntax. With regard to the spatial index technique, Lucene Spatial is commonly used because of its fast development and active support from the community. GeoSPARQL-Jena indexes and caches intermediate spatial query results to accelerate queries with similar graph patterns thereafter, and it supports dataset-custom spatial index constructing, which cannot be migrated to other datasets. Virtuoso uses R-tree for spatial indexing. In Virtuoso and Stardog, there is no way to switch off spatial queries with a spatial index, while the others support switching off spatial indexing. GeoSPARQL-Jena has been very recently

developed, and it supports transformation between different spatial reference systems (SRSs), whereas the other stores only support WGS84. This usually entails SRS transformation before importing into the stores.

Table 2. Qualitative analysis results of geospatial query support of the selected RDF stores.

	RDF4J	GeoSPARQL-Jena	Virtuoso	Stardog	GraphDB
Storage	Native	Native	RDBMS	Native	Native
Geometry serialization	WKT	WKT, GML	WKT	WKT	WKT, GML
GeoSPARQL-compliance ¹	Full	Full	Full	Partly	Full
Use of spatial index	Optional ²	Optional	Must	Must	Optional
Spatial index technique	Lucene Spatial	Custom	R-tree	Lucene Spatial	Lucene Spatial
Supported SRS	WGS84	Geographic and project SRSs	WGS84	WGS84	WGS84

¹ It refers to the compliance with spatial functions in the simple features relation family; e.g., it does not include support for SRS and GML. ² The support of Lucene Spatial in RDF4J currently has problems.

6. Results of the Spatially Enabled RDF Store Benchmark

6.1. Experimental Setup

We ran the benchmark in a machine with the processor Intel Core i7-6700 (8M Cache, up to 4.00 GHz), 24 GB of RAM, and the operating system Ubuntu 18.04.1 LTS.

In the first scenario, the ICOS CP metadata was exported from its current RDF4J-based store into an RDF dump file with the 2.2 M triples. In the second scenario, the Geographica data was downloaded from its online repository as dump files. The benchmark programs first loaded the dump files into each store and recorded the loading time (including the spatial index construction time).

Each query in the benchmark (Table 1) was run three times after a number of warm-up queries were finished. In order to test the difference between using and not using a spatial index, we tested GraphDB in both modes (the queries Q1–Q5 and Q15 do not differ in either manner, as spatial indexing cannot be used in these queries in GraphDB). To determine the influence of the means of communication with the stores, we tested different communication interfaces with Virtuoso and Stardog. We tested Virtuoso's native interface Java Database Connectivity (JDBC) and RDF4J for operation and communication (as RDF4J is also commonly used as a library to manipulate other stores). We also tested Stardog's native interface SNARL and RDF4J for communication. We set a 1-h timeout for all queries.

6.2. Benchmark Results with ICOS CP Metadata

6.2.1. Query Performance

Table 3 summarizes the loading time for the ICOS CP metadata of each store. All the stores import, and possibly construct, the spatial index for the 2.2 M triple dataset in a reasonable time. Notice that the loading time is for the entire ICOS CP dataset, which contains around 1000 spatial objects and many other object types.

Table 3. Loading time of each store for ICOS CP metadata.

	RDF4J	GeoSPARQL-Jena	Virtuoso	Stardog	GraphDB
Loading time	62.4 s	88.0 s	94.5 s	134.1 s	154.1 s

Table 4 summarizes the results for the average query execution time regarding RDF4J, GeoSPARQL-Jena, Virtuoso (connected through JDBC and RDF4J), Stardog (connected through

SNARL and RDF4J), and GraphDB (with and without using a spatial index). For non-topological functions (Q1–Q5), GraphDB generally triumphs over the other stores. The performance of RDF4J is comparable to that of GraphDB. Compared with the other stores, GeoSPARQL-Jena and Virtuoso take much more time to calculate buffers of polylines and polygons, which might be the result of their more complex custom implementations. Stardog does not support any of the non-topological functions. For spatial selection queries (Q7–Q17), RDF4J provides generally good performance in terms of query response time. GraphDB also has comparable performance records, and it is much faster than the other stores for Q7 (equal polyline finding). Virtuoso has the best performance for Q13 (i.e., find all points in a given polygon, which is a very useful query for ICOS CP and many other linked data-based projects). Stardog has a reasonable performance but is much slower for Q7 using its native SNARL interface. For spatial join queries (Q18–Q27), RDF4J provides the best performance for four queries (Q20, Q21, Q22, Q27), and it is generally fast at intersection queries. GeoSPARQL-Jena is fastest at Q23, Q24, and Q25 and is generally superior at within functions. GraphDB is the best at Q18 (without using a spatial index), Q19 (with a spatial index), and Q26 (with an index), and it generally provides reasonable performance for all queries. Virtuoso and Stardog are relatively slow for Q19, Q20, Q23, and Q24, which are mainly within and intersection queries; for these queries, the query performance differs by nearly three orders of magnitude, which indicates that some stores (Stardog and Virtuoso) may not be suited to the tasks of conducting spatial join queries.

Table 4. Average query response time of selected stores of benchmark queries with ICOS CP metadata (shortest response times in bold). Time unit is millisecond. The results that are different from the results produced from JTS (ArcGIS for Q15) are shaded (see Section 6.2.2).

Query Time (ms)	RDF4J	GeoSPARQL-Jena	Virtuoso		Stardog		GraphDB	
			JDBC	RDF4J	SNARL	RDF4J	Indexed	Non-Indexed
Q1	1.70	4.04	3.76	7.37				1.51
Q2	1.27	2.62	2.14	2.14				1.15
Q3	1.44	6.85	4.29	5.02				1.19
Q4	1.45	100.93	944.95	979.93				1.12
Q5	1.29	3.68	64.98	70.41				2.51
Q7	21.48	12.84	53.11	56.62	142.72	33.58	3.57	5.83
Q8	7.13	4.34	8.97	10.20	11.93	4.23	13.13	2.58
Q9	1.93	4.83	21.02	22.80	10.19	5.20	5.57	19.32
Q10	1.10	3.68	10.13	11.71	11.90	4.02	4.27	2.53
Q11	1.20	3.39	9.39	12.54			9.73	2.80
Q12	1.19	3.64	55.17	47.79			2.83	2.95
Q13	2.54	5.04	1.85	4.05	10.05	4.49	4.03	7.17
Q15	2.35	20.20	2.10	4.80	24.57	3.30		1.78
Q16	1.47	2.51	1.78	4.63	8.96	3.39	2.83	5.31
Q17	1.37	1.87	28.15	26.82	10.80	3.73	2.34	2.24
Q18	136.81	71.19	39.92	45.84	280.10	196.99	31.01	9.33
Q19	2569.42	454.32	776.08	666.98	5786.66	5363.58	4.55	29.82
Q20	1.75	7.40	1536.15	1541.63	621.66	583.33	19.10	3.61
Q21	1.51	2.20	47.86	45.06	10.95	4.17	14.20	3.57
Q22	1.25	10.13	759.84	783.62	11.27	6.21	14.97	11.86
Q23	422.94	3.57	277.79	279.90	1605.72	1499.08	3.58	3.81
Q24	76.92	2.80	111.08	90.40	211.97	226.99	18.05	5.24
Q25	2.09	1.73	42.27	32.47			6.93	5.29
Q26	719.31	165.46	619.50	629.43			19.11	23.49
Q27	2.19	2.62	58.81	52.85			5.34	11.43

We also observe that the performance with Virtuoso’s native JDBC interface is similar to that with the RDF4J interface. With Stardog, using RDF4J as the interface generally leads to better performance than using its native interface SNARL, as RDF4J caches some intermediate query results. From the results, we observe that GeoSPARQL-Jena and RDF4J demonstrate a significant caching effect, i.e., the query time of the second and third times substantially drops compared with that of the first time. This

is in line with their means of implementation: they cache a lot of intermediate query results. Other stores do not show a clear caching effect.

6.2.2. Query Correctness

Evaluating query correctness for spatial queries is complex, particularly when the queries deal with a large amount of data. However, query correctness is an important aspect in the assessment of the selected stores, especially because it is common for different stores to implement the spatial query functions differently. In this paper, we partially evaluate and discuss the query correctness by observing the results from the above-described benchmarking.

For topological queries, GeoSPARQL follows the definitions of topological relations in the dimensionally extended nine-intersection model DE-9IM [52]. A well-known and reliable implementation of DE-9IM is the Java library JTS Topology Suite, JTS (<https://github.com/locationtech/jts>). In this study, we performed all the benchmark queries using the JTS library, and we treat the returned results as reference results for the evaluation of the RDF stores. Queries whose number of returned results from the RDF stores differs from the number returned from JTS are shaded in Table 4. One exception is *Q15*, which is not supported by JTS (as JTS does not support distance calculation in geographic SRSs). Thus, we calculated it in ArcGIS 10.3.1 as reference results.

For *Q1–Q9*, all the evaluated stores provide the same number of returned results as JTS. For *Q10*, we find that Stardog handles the spatial relation intersect (for polygons) in a manner that differs from the other stores; it returns the same results as the other stores return for *Q11*, which queries all the polygons that overlap a given polygon. That is, the intersect function for polygons in Stardog is actually equivalent to the overlap function in other stores, and Stardog does not have the function overlap. For *Q15*, only GraphDB provides the same results as ArcGIS (10 results); RDF4J, Virtuoso, and Stardog return 11 results (probably linked to precision settings); and GeoSPARQL-Jena fails to give any result in spite of the relatively long time it takes on this query. For *Q18*, RDF4J fails to return any result, and this problem is potentially linked to the precision setting in RDF4J when finding equal points. For *Q21* and *Q25*, RDF4J, Stardog (only for *Q21*), and GraphDB (using spatial indexing) return 563 results; Virtuoso returns 567 results; and GeoSPARQL-Jena, and GraphDB (without using spatial indexing) return 565 results. This divergence may be linked to Lucene Spatial filtering out some results because of factors such as precision settings in different stores. JTS returns 565 results for these queries.

6.3. Benchmark Results with Geographica Datasets

In the second scenario, we tested the selected RDF stores with large geospatial datasets. This scenario is more in line with conventional SDIs, in which geospatial data dominates. Therefore, benchmarking the RDF stores with such large datasets to test their scalabilities will potentially benefit the SDI and geospatial linked data communities, as it is common for a project (especially dedicated SDIs) to have a vast number of geospatial objects.

The loading time of the six datasets in the five selected stores is presented in Table 5, and the query performance is demonstrated in Table 6.

From Table 5, we can observe that a large number of geospatial objects do not lengthen the loading time for RDF4J, GeoSPARQL-Jena, and GraphDB. For RDF4J and GeoSPARQL-Jena, this is because they do not build a spatial index while data loading; for GraphDB, the spatial index construction is completed in a short time. Virtuoso takes longer (more than 10 min) to load and construct a spatial index for the data. For Stardog, the spatial indexing process is slow, as the whole loading and index construction process takes nearly five hours.

Table 5. Loading time of each store for Geographica datasets.

	RDF4J	GeoSPARQL-Jena	Virtuoso	Stardog	GraphDB
Loading time	48.6 s	89.6 s	620.0 s	4.6 h	89.7 s

Table 6. Average query response time of selected stores of benchmark queries with Geographica datasets (shortest response time in bold). Time unit is second unless specified as hour.

Query Time (s)	RDF4J	GeoSPARQL-Jena	Virtuoso		Stardog		GraphDB	
			JDBC	RDF4J	SNARL	RDF4J	Indexed	Non-Indexed
Q1	0.015	0.011	0.020	0.088				0.009
Q2	0.011	0.003	0.023	0.016				0.002
Q3	0.011	0.005	0.059	0.074				0.009
Q4	0.006	0.079	0.043	0.061				0.005
Q5	0.003	0.003	0.203	0.250				0.003
Q7	0.527	0.055	0.120	0.130	0.515	0.533	0.079	2.515
Q8	0.482	0.139	0.148	0.156	0.178	0.140	0.139	5.536
Q9	0.013	0.005	0.022	0.035	0.021	0.017	17.442	0.046
Q10	0.776	0.012	0.120	0.181	0.095	0.083	0.125	0.867
Q11	0.685	0.014	0.077	0.125			0.404	0.034
Q12	0.009	0.005	0.094	0.116			1.880	0.076
Q13	0.093	0.003	0.052	0.054	0.786	0.776	13.163	0.026
Q15	0.222	4.529	0.119	0.147	0.921	0.760		1.645
Q16	0.003	0.384	0.006	0.009	0.013	0.009	124.135	0.003
Q17	0.003	0.002	0.027	0.030	0.008	0.007	148.334	0.002
Q18	0.027	0.060	0.060	0.009	0.014	0.008	0.010	1.491
Q19	>1 h	544.082	938.553	932.699	>1 h	>1 h	1026.021	>1 h
Q20	9.031	0.021	2.677	2.679	2416.730	2439.824	1.013	9.887
Q21	3.985	0.005	1.715	1.673	4.174	4.471	2.969	3.573
Q22	8.569	0.003	2.071	2.130	0.380	0.386	0.441	9.104
Q23	5.940	0.004	2.370	2.463	4.681	4.857	1.677	3.382
Q24	7.875	0.007	2.358	2.529	0.129	0.113	0.099	3.304
Q25	3.940	0.017	6.531	6.596			0.612	62.865
Q26	0.040	0.033	3.460	3.758			0.531	7.431
Q27	18.274	0.111	1.059	0.644			0.077	17.337

The query performance of GraphDB is generally better than that of the others for the non-topological construct queries *Q1–Q5*, and RDF4J, GeoSPARQL-Jena, and Virtuoso have comparable performances. For the spatial selection queries *Q7–Q17*, all the RDF stores respond in a reasonable time, and GeoSPARQL-Jena performs better than the others in most of the queries. The spatial join query *Q19* is the most computationally expensive query in the benchmark: RDF4J, Stardog, and GraphDB without spatial indexing all time out for this query, while GeoSPARQL-Jena provides the shortest time for this query (less than 10 min). For other spatial join queries, *Q20–Q27*, GeoSPARQL-Jena generally performs better than the others, and all stores have reasonable response times. It is observed that different query interfaces do not have much effect on the query response time. For GraphDB, the indexed mode generally returns the results much quicker than the non-indexed mode. The exceptions are *Q16* and *Q17*, for which GraphDB has a very similar performance to that of RDF4J with quick responses; this might be the result of the simplistic implementation of the disjoint function in RDF4J (GraphDB is dependent on RDF4J in the mode that does not use spatial indexing).

7. Discussion

In this paper, we comprehensively assess and benchmark five popular and well-known spatially enabled RDF stores, i.e., RDF4J, GeoSPARQL-Jena, Virtuoso, Stardog, and GraphDB. It is encouraging to see the increasing maturity of the technical environment for the support of geospatial linked data, as well as the increasing compliance with GeoSPARQL compared with previous benchmarks. That is, progressively more mainstream and well-known RDF stores are (partially) supporting GeoSPARQL. Another positive observation is that the syntaxes used for geospatial queries with GeoSPARQL are the same in RDF4J, GeoSPARQL-Jena, Virtuoso, and GraphDB in this benchmark, which implies that the geospatial queries are cross-database interoperable in terms of query syntax (Stardog does not have the same geospatial query syntax as the others). Listing 1 is an example query of *Q23* in the first scenario in RDF4J, GeoSPARQL-Jena, Virtuoso, and GraphDB (without using spatial indexing, as the

filter should be replaced with a triple relation in the query when using spatial indexing in GraphDB, i.e., `?geom1 geo:sfWithin ?geom2`). Listing 2 is the corresponding query used in Stardog.

Listing 1. Query syntax of Q23 in the first scenario in RDF4J, GeoSPARQL-Jena, Virtuoso, and GraphDB (without indexing).

```
PREFIX geo: <http://www.opengis.net/ont/geosparql#>
PREFIX geof: <http://www.opengis.net/def/function/geosparql/>
PREFIX sf: <http://www.opengis.net/ont/sf#>
SELECT ?geom1 ?geom2
WHERE {
  ?geom1 a sf:LineString; geo:asWKT ?wkt1.
  ?geom2 a sf:Polygon; geo:asWKT ?wkt2.
  FILTER(geof:sfWithin(?wkt1,?wkt2)).}
```

Listing 2. Query syntax of Q23 in the first scenario in Stardog.

```
PREFIX geo: <http://www.opengis.net/ont/geosparql#>
PREFIX geof: <http://www.opengis.net/def/function/geosparql/>
PREFIX sf: <http://www.opengis.net/ont/sf#>
SELECT ?geom1 ?geom2
WHERE {
  ?geom1 a sf:LineString.
  ?geom2 a sf:Polygon.
  FILTER(geof:relate(?geom1,?geom2,geo:within)).}
```

The query performance is generally acceptable, and it is much better than previous benchmarking results because RDF stores have developed and computer hardware has advanced. GeoSPARQL was supported in all the stores except for Stardog after 2018, which also makes this paper timely in its contribution to the comprehensive understanding of this subject. We believe the increasingly mature technical environment will benefit the development of the next generation of SDIs, in which linked data will expectedly play an important role.

From the query performance of the evaluated stores in the two scenarios, we observe that GraphDB is generally better than the others at non-topological queries, which are useful in many real-world spatial analyses: e.g., buffering is important for location selection analysis. GeoSPARQL-Jena and RDF4J are generally better than the other RDF stores at spatial selection queries, which are useful for many real-world use cases: e.g., for ICOS CP, the overlap and within functions are the most useful queries for enabling a user-defined spatial search. GeoSPARQL-Jena is superior at spatial join queries—operations used for functions such as establishing relations between the cadaster registries (points) and building objects (polygons).

A prerequisite of (partially) achieving cross-database interoperability is that the GeoSPARQL standard should be used when possible. The lightweight nature of the GeoSPARQL vocabulary means that accomplishing interoperability with GeoSPARQL for other spatial-relevant ontologies does not entail much work since, in most cases, it can be accomplished with subclass/subproperty inheritance. Nevertheless, we believe that GeoSPARQL should support more serializations to realize its wider adoption. It is especially desirable to have support for GeoJSON, which is widely accepted by the web development community.

One lesson learned from the experimental results is that, for a moderate amount of geospatial data (scenario 1 with about 1000 spatial objects), spatial indexing could be an overhead both for data loading and querying, whereas spatial indexing is certainly necessary when querying a large number of geospatial objects (scenario 2 with about 150,000 spatial objects). Most selected RDF stores provide reasonable data loading and spatial index construction times, except for Stardog, which takes nearly five hours to load and index the Geographica datasets. That is, we believe that enabling spatial indexing for querying large geospatial datasets is imperative, and constant change and injection of data

are also feasible as long as the data loading and indexing times are reasonable. In this context, further assessment of the RDF stores with an even larger amount of data is desirable, which is interesting for large-scale geospatial linked data deployment.

From this assessment, we observe that most of the selected RDF stores with spatial indexing use Lucene Spatial for its easy deployment and wide support from the community. We argue that no spatial indexing technique can best fit all applications. In fact, it would be better to also enable developers and geospatial experts to configure specific and optimized spatial indexes tailored for certain datasets. This functionality is already provided by some RDF stores, e.g., RDF4J and GeoSPARQL-Jena.

Despite the promising results and advancements, there are still some challenges. One of the most significant challenges is query correctness. Although the queries are interoperable in terms of query syntax across most of the selected RDF stores, the returned results are sometimes not the same because of different implementations and interpretations of, for example, spatial topological relations. This issue renders the cross-database interoperability problematic for geospatial queries, which is rarely the case for other types of queries following the W3C recommendations. We think further development of the RDF stores might mitigate this issue, but to overcome this problem, we may need a community-backed and commonly used compliance testing suite regarding the OGC Implementation Standard for Geographic Information [52] for the implementation and interpretation of spatial functions. For the query correctness issue, we propose that a major cause is the different strategies for handling precision in the stores. Furthermore, as only four of the five stores support the SRS of WGS84, conducting spatial operations in a geographic SRS and converting data from other SRSs to WGS84 can lead to precision loss and thus incorrect or inaccurate results. Therefore, further investigation of the effect of precision settings in RDF stores is deemed necessary.

Another important topic that deserves investigation is the performance comparison between spatially enabled RDF stores and state-of-the-art OGC services (e.g., WFS). We speculate that current OGC services are superior to RDF stores at spatial queries. This raises the question of how much faster OGC services are than RDF stores. The answer to that question will potentially unveil the answers to two other questions: (1) Should we (partly) leave the spatial operations to RDBMS-backed OGC services or other GIS tools, especially since spatial join queries do not perform favorably in the evaluated RDF stores, until their spatial capacities are significantly advanced? (2) Should data publishers or third parties pre-compute important and relevant spatial relations and publish them along with the data, which will greatly diminish the need for real-time spatial operations at the cost of pre-computation and increase in data volume? Our initial opinion is that it will be beneficial to pre-compute some important spatial relations and release the relations together with geospatial linked data.

8. Conclusions

Linked data is a promising means to resolve the limitations concerning data integration and semantic heterogeneity of the current SDI solutions; thus, linked data has been seen as one of the key factors moving SDIs toward the next generation. The technical environment and support are important for deploying geospatial linked data. In this paper, we present an assessment and benchmarking concerning the spatial query capacities of five RDF stores, i.e., RDF4J, GeoSPARQL-Jena, Virtuoso, Stardog, and GraphDB. We tested the selected stores in two scenarios. One scenario involves benchmarking the RDF stores with ICOS CP metadata, a large-scale Earth Science data infrastructure in which geospatial data is integrated with other types of data. The other scenario is in a dedicated SDI environment with a large amount of purely geospatial data, which is a mixture of crowd-sourced and authoritative geospatial data. The queries used in this study are mainly from the Geographica benchmark. The results demonstrate that GeoSPARQL compliance has advanced dramatically in the last several years for the RDF stores, and query performances are generally acceptable. Furthermore, spatial indexing is important when querying a large number of geospatial objects. However, query correctness remains a challenge for cross-database interoperability.

Supplementary Materials: The benchmarking programs used in this study are available at <https://github.com/RightBank/Benchmarking-spatially-enabled-RDF-stores>. The test data from ICOS CP has been published at <https://doi.org/10.18160/9D9W-WT2P>.

Author Contributions: W.H., O.M., and L.H. conceived and designed this study; S.A.R. and W.H. implemented the benchmarking for the RDF stores; W.H. wrote the paper, with revisions from S.A.R., O.M., and L.H.; O.M. is the system architect at ICOS CP and partially proposed the need for this study; all authors read and approved this manuscript.

Funding: The work was supported by Lund University and China Scholarship Council.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Van den Brink, L.; Janssen, P.; Quak, W.; Stoter, J. Towards a high level of semantic harmonisation in the geospatial domain. *Comput. Environ. Urban Syst.* **2017**, *62*, 233–242. [CrossRef]
2. INSPIRE. Available online: <https://inspire.ec.europa.eu/> (accessed on 2 December 2018).
3. Schade, S.; Smits, P. Why linked data should not lead to next generation SDI. In Proceedings of the 2012 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Munich, Germany, 22–27 July 2012; pp. 2894–2897.
4. Parsons, E. If You Can't Link to it ... Does it Exist? Available online: <https://www.edparsons.com/2017/09/cant-link-exist/> (accessed on 24 April 2019).
5. Janowicz, K.; Scheider, S.; Pehle, T.; Hart, G. Geospatial semantics and linked spatiotemporal data—Past, present, and future. *Semant. Web* **2012**, *3*, 321–332.
6. Vancauwenberghe, G.; Valeckaite, K.; Van Loenen, B.; Donker, F.W. Assessing the Openness of Spatial Data Infrastructures (SDI): Towards a Map of Open SDI. *IJSDIR* **2018**, *13*, 88–100.
7. Lutz, M.; Sprado, J.; Klien, E.; Schubert, C.; Christ, I. Overcoming semantic heterogeneity in spatial data infrastructures. *Comput. Geosci.* **2009**, *35*, 739–752. [CrossRef]
8. EuroSDR. EuroSDR Annual Report 2018. Available online: http://www.eurocdr.net/sites/default/files/images/inline/eurocdr_annual_report_2018.pdf (accessed on 12 June 2019).
9. AGILE 2018 Workshop 'SDI Research and Strategies towards 2030'. Available online: <https://kcoappendata.eu/sdi2030/> (accessed on 25 July 2018).
10. W3C. Resource Description Framework (RDF). Available online: <https://www.w3.org/RDF/> (accessed on 6 January 2018).
11. W3C. SPARQL Query Language for RDF. Available online: <https://www.w3.org/TR/rdf-sparql-query/> (accessed on 20 March 2019).
12. Perry, M.; Herring, J. OGC GeoSPARQL—A Geographic Query Language for RDF Data. Technical report, Open Geospatial Consortium, 2012. Available online: <http://www.opengeospatial.org/standards/geosparql> (accessed on 1 May 2019).
13. ICOS Carbon Portal. Available online: <https://www.ICOSCP.eu/> (accessed on 7 January 2019).
14. Garbis, G.; Kyzirakos, K.; Koubarakis, M. Geographica: A benchmark for geospatial RDF stores (long version). In Proceedings of the International Semantic Web Conference, Sydney, NSW, Australia, 21–25 October 2013; pp. 343–359.
15. World Wide Web Consortium (W3C). W3C Semantic Web Activity. Available online: <https://www.w3.org/2001/sw/> (accessed on 25 July 2017).
16. Kuhn, W.; Kauppinen, T.; Janowicz, K. Linked data—A paradigm shift for geographic information science. In Proceedings of the International Conference on Geographic Information Science, Vienna, Austria, 24–26 September 2014; pp. 173–186.
17. Huang, W.; Harrie, L. Towards knowledge-based geovisualisation using Semantic Web technologies: A knowledge representation approach coupling ontologies and rules. *Int. J. Digit. Earth* **2019**. [CrossRef]
18. INSPIRE. Linking INSPIRE Data: Draft Guidelines and Pilots. Available online: <https://inspire.ec.europa.eu/news/linking-inspire-data-draft-guidelines-and-pilots> (accessed on 20 December 2018).
19. INSPIRE. Guidelines for the Encoding of Spatial Data. Available online: https://inspire.ec.europa.eu/documents/Data_Specifications/D2.7_v3.3rc3.pdf (accessed on 28 April 2019).

20. Regalia, B.; Janowicz, K.; Mai, G.; Varanka, D.; Usery, E.L. GNIS-LD: Serving and Visualizing the Geographic Names Information System Gazetteer as Linked Data. In Proceedings of the European Semantic Web Conference, Heraklion, Crete, Greece, 3–7 June 2018; pp. 528–540.
21. Goodwin, J.; Dolbear, C.; Hart, G. Geographical linked data: The administrative geography of great britain on the semantic web. *Trans. Gis* **2008**, *12*, 19–30. [[CrossRef](#)]
22. Folmer, E.; Beek, W.; Rietveld, L. Linked Data Viewing as part of the Spatial Data Platform of the Future. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2018**, *42*, 49–52. [[CrossRef](#)]
23. Hietanen, E.; Lehto, L.; Latvala, P. Providing Geographic Datasets as Linked Data in SDI. *Isprs-Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2016**, *41*, 583–586. [[CrossRef](#)]
24. Shi, L.; Sukhobok, D.; Nikolov, N.; Roman, D. Norwegian State of Estate Report as Linked Open Data. In Proceedings of the OTM Confederated International Conferences on the Move to Meaningful Internet Systems, Rhodes, Greece, 23–27 October 2017; pp. 445–462.
25. Ronzhin, S.; Folmer, E.; Mellum, R.; von Brasch, T.E.; Martin, E.; Romero, E.L.; Kytö, S.; Hietanen, E.; Latvala, P. Next Generation of Spatial Data Infrastructure: Lessons from Linked Data implementations across Europe. Report of Open ELS Project. Available online: https://openels.eu/wp-content/uploads/2019/04/V2_Next_Generation_SDI_Lessons-from-LD-implementations-across-Europe_1.pdf (accessed on 20 May 2019).
26. Usery, E.L.; Varanka, D. Design and development of linked data from the national map. *Semant. Web* **2012**, *3*, 371–384.
27. Cheatham, M.; Krisnadhi, A.; Amini, R.; Hitzler, P.; Janowicz, K.; Shepherd, A.; Narock, T.; Jones, M.; Ji, P. The GeoLink knowledge graph. *Big Earth Data* **2018**, *2*, 131–143. [[CrossRef](#)]
28. Stadler, C.; Lehmann, J.; Höffner, K.; Auer, S. Linkedgeodata: A core for a web of spatial open data. *Semant. Web* **2012**, *3*, 333–354.
29. Van den Brink, L.; Barnaghi, P.; Tandy, J.; Atemezeng, G.; Atkinson, R.; Cochrane, B.; Fathy, Y.; Castro, R.G.; Haller, A.; Harth, A. Best Practices for Publishing, Retrieving, and Using Spatial Data on the Web. *Semant. Web* **2019**, *10*, 95–114. [[CrossRef](#)]
30. Narock, T.; Shepherd, A. Semantics all the way down: The Semantic Web and open science in big earth data. *Big Earth Data* **2017**, *1*, 159–172. [[CrossRef](#)]
31. Hofer, B.; Mäs, S.; Brauner, J.; Bernard, L. Towards a knowledge base to support geoprocessing workflow development. *Int. J. Geogr. Inf. Sci.* **2016**, *31*, 694–716. [[CrossRef](#)]
32. Keßler, C.; Raubal, M.; Wosniok, C. Semantic rules for context-aware geographical information retrieval. In Proceedings of the European Conference on Smart Sensing and Context, Guildford, UK, 16–18 September 2009; pp. 77–92.
33. Wiemann, S.; Bernard, L. Spatial data fusion in spatial data infrastructures using linked data. *Int. J. Geogr. Inf. Sci.* **2016**, *30*, 613–636. [[CrossRef](#)]
34. Huang, W.; Mansourian, A.; Abdolmajidi, E.; Xu, H.; Harrie, L. Synchronising geometric representations for map mashups using relative positioning and Linked Data. *Int. J. Geogr. Inf. Sci.* **2018**, *32*, 1117–1137. [[CrossRef](#)]
35. Guo, Y.; Pan, Z.; Heflin, J. LUBM: A benchmark for OWL knowledge base systems. *Web Semant. Sci. Serv. Agents World Wide Web* **2005**, *3*, 158–182. [[CrossRef](#)]
36. Schmidt, M.; Hornung, T.; Lausen, G.; Pinkel, C. SP²Bench: A SPARQL performance benchmark. In Proceedings of the 2009 IEEE 25th International Conference on Data Engineering, Shanghai, China, 29 March–2 April 2009; pp. 222–233.
37. Bizer, C.; Schultz, A. The berlin sparql benchmark. *Int. J. Semant. Web Inf. Syst.* **2009**, *5*, 1–24. [[CrossRef](#)]
38. Morsey, M.; Lehmann, J.; Auer, S.; Ngomo, A.-C.N. DBpedia SPARQL benchmark–performance assessment with real queries on real data. In Proceedings of the International Semantic Web Conference, Bonn, Germany, 23–27 October 2011; pp. 454–469.
39. Duan, S.; Kementsietsidis, A.; Srinivas, K.; Udrea, O. Apples and oranges: A comparison of RDF benchmarks and real RDF datasets. In Proceedings of the 2011 ACM SIGMOD International Conference on Management of Data, Athens, Greece, 12–16 June 2011; pp. 145–156.
40. Perry, M.; Sheth, A.P.; Hakimpour, F.; Jain, P. Supporting complex thematic, spatial and temporal queries over semantic web data. In Proceedings of the International Conference on GeoSpatial Semantics, Mexico City, Mexico, 29–30 November 2007; pp. 228–246.
41. Papadimitriou, F. The algorithmic complexity of landscapes. *Landscape Res.* **2012**, *37*, 591–611. [[CrossRef](#)]

42. Kolas, D. A Benchmark for Spatial Semantic Web Systems. In Proceedings of the International Workshop on Scalable Semantic Web Knowledge Base Systems, Karlsruhe, Germany, 26–30 October 2008.
43. Battle, R.; Kolas, D. Enabling the geospatial semantic web with parliament and geosparql. *Semant. Web* **2012**, *3*, 355–370.
44. Athanasiou, S.; Bezati, L.; Giannopoulos, G.; Patroumpas, K.; Skoutas, D. GeoKnow– Making the Web an Exploratory for Geospatial Knowledge: Deliverable 2.1.1 Market and Research Overview. Available online: http://svn.aksw.org/projects/GeoKnow/Public/D2.1.1_Market_and_Research_Overview.pdf (accessed on 30 September 2018).
45. Bellini, P.; Nesi, P. Performance assessment of RDF graph databases for smart city services. *J. Vis. Lang. Comput.* **2018**, *45*, 24–38. [[CrossRef](#)]
46. FORCE 11. Guiding Principles for Findable, Accessible, Interoperable and Re-Usable Data Publishing Version B1.0. Available online: <https://www.force11.org/fairprinciples> (accessed on 5 March 2019).
47. Bechhofer, S.; Buchan, I.; De Roure, D.; Missier, P.; Ainsworth, J.; Bhagat, J.; Couch, P.; Cruickshank, D.; Delderfield, M.; Dunlop, I. Why linked data is not enough for scientists. *Future Gener. Comput. Syst.* **2013**, *29*, 599–611. [[CrossRef](#)]
48. Abam, M.A.; De Berg, M.; Hachenberger, P.; Zarei, A. Streaming algorithms for line simplification. *Discret. Comput. Geom.* **2010**, *43*, 497–515. [[CrossRef](#)]
49. W3C. PROV-O: The PROV Ontology. W3C Recommendation. Available online: <https://www.w3.org/TR/prov-o/> (accessed on 28 April 2019).
50. W3C. The Organization Ontology. W3C Recommendation. Available online: <https://www.w3.org/TR/vocab-org/> (accessed on 28 April 2019).
51. Mirzov, O.; Huang, W.; Raza, S.A. ICOS CP metadata used for RDF store benchmarking. *Res. Data.* **2019**. [[CrossRef](#)]
52. Herring, J. OpenGIS Implementation Standard for Geographic Information-Simple feature access-Part 1: Common architecture. *OGC Doc.* **2011**, *4*, 122–127.



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).