Semantic Web Queries over Scientific Data
(extended abstract)

Andrej Andrejev
Department of Information Technology
Uppsala University
Sweden

The amount of scientific and engineering data has grown exponentially in the recent decades [29], and this growth includes the rapid increase in the amount of data sources publicly available on the web [16]. Complexity and diversity (structural, terminological, etc.) of this data is also expected to rise steadily in the coming decades, as novel data models emerge along with new and unforeseen applications. The efforts directed towards data integration and interoperability are becoming of vital importance [7], [14], [21].

One promising direction of these efforts is the search for *lingua franca* - a model general and flexible enough, so that the other, more specific data models can be mapped into it in a lossless way; and yet being meaningful and easy to understand and query. Semantic Web [8] and Linked Open Data [9] are conceived as a potential solution: all kinds of data and metadata can be represented as a graph with nodes and (classes of) edges identified by globally unique URIs. The original aim of this data model was to describe the resources available on the web - hence the name: Resource Description Framework (RDF).

For querying RDF datasets the graph-based pattern-matching query language SPARQL was proposed and recommended by W3C. In its current state, SPARQL 1.1 allows queries that retrieve data from an RDF graph, filter the potential query solutions, and postprocess them before emitting the results. SPARQL bridges the gap between the traditionally separated *data* and *metadata*, the latter being the semantic, structural, statistical, and other kinds of descriptions of the former. A potential to fully combine *data* and *metadata* search and conditions in one query, thus simplifying the process and eliminating extra round-trips to the remote data sources, is contained within the Semantic Web paradigm but is not fully realized.

The main problem is that although most kinds of other data models can be mapped to RDF, the efficiency and usefulness of such mappings might become unsatisfactory. For example, *numeric multidimensional arrays*, a data abstraction that is central in all natural sciences and constitutes the main

bulk of accumulated data, when mapped to RDF have to be transformed into graphs, thus making even the simplest array operations (e.g. *element access*) unfeasible to perform or even express in a general case.

So far RDF and SPARQL gained limited adoption within the scientific community, due to the lack of array support [20] and other important features – such as extensibility with user-defined functions, query modularity, integration with existing environments and workflows. Some users turn towards the 'more mature' relational database technology (e.g. [30]), eventually extending it with missing array functionality [11], [24], while others find the idea of relational schema design too restrictive, resorting to specialized file formats or hierarchical databases. In either case, array data is separated from metadata and the latter sometimes ends up encoded into eventually very complex file names, so that data retrieval and processing become a nontrivial task for a programmer. While many complications arise from the need of manual data/metadata re-integration, another challenging task is the adequate estimation of data quantities and distributions, in order to come up with an optimal order of data retrieval operations.

Automating the task of programming the data retrieval and processing is the essence of *query optimization*. The relational database management systems (RDBMS) were taking care of data statistics and evaluation cost models, producing optimal execution plans since 1970s [28], [10]. The modern RDF stores [13], [22], [23], [31], [32] employ similar techniques based on indexing, query rewriting and materialized views in order to address the challenges of web-scale query processing [1], [15], [27][17], [18], [25], [26], [27].

Addressing different data and metadata sources in a single query is possible within a data integration framework where machine-readable descriptions of the structure and semantics of the available data are present. RDF is specifically designed for publishing such descriptions by creating and referring to vocabularies of globally-scoped terms, and by defining the logical relationships within and across such vocabularies, using the RDF Schema and OWL formalisms.

In the presented work, the RDF data model has been extended, so that numeric multidimensional arrays of arbitrary shape and size (including those exceeding the main memory limit) can be attached as *values* in *subject-property-value* RDF triples. We call this model *RDF with Arrays*, and it is backwards-compatible with the basic RDF: arrays that are recognized within the imported RDF graphs are *consolidated*, i.e. their elements are co-located and the array shape is determined. Internal array storage facilities are used in

that case, and such structured data becomes available to the queries using array-oriented features.

The important research questions answered in this work are:

1. How can RDF and SPARQL be extended to be suitable for scientific and engineering numeric data representation and analysis tasks, in particular, those which combine data and metadata?
2. How can extended SPARQL query processing be implemented on the basis of a database management system? In particular, what extensions to the underlying algebra operators are needed for efficient processing of SPARQL queries?
3. How can existing state-of-art data persistence approaches (RDBMS, specialized file formats, array databases) be utilized for scalable storage of RDF data with arrays?
4. How can query functionality of extended SPARQL be integrated into existing environments and workflows for scientific and engineering data analysis?
5. How do we measure the impact of data storage decisions and retrieval strategies on the overall query performance?

In few words, the aim of this work is providing a viable solution (both conceptual and technical) opening the benefits of the Semantic Web approach to scientific data management, and making scientific data available and interoperable on the Semantic Web.

The Thesis answers the research questions by the following results:

- Question 1 is answered by the extension of the RDF data model with array data, and the design of Scientific SPARQL - a query language combining graph and array-based semantics. Scientific SPARQL is easily extensible with foreign functions, includes functional programming primitives, and is suitable for specifying complex modular queries, at once capturing the data retrieval, filtering, and post-processing needs, typical to scientific computing.
- Question 2 is answered by the design and implementation of Scientific SPARQL Database Manager (SSDM) - an extensible in-memory DBMS including SciSPARQL query processor.
- Questions 3 is answered by specifying a storage extensibility mechanism, which allows to query *RDF with Arrays* data stored by a wide range of interfaced storage systems, e.g. relational DBMSs, binary files, or array databases.

- Question 4 is addressed by presenting a tight integration of SciSPARQL queries into Matlab, along with the APIs for calling SciSPARQL queries from other algorithmic languages. It is shown how Semantic Web styled metadata can be used for annotation and, eventually, search for the numeric computation results, while essentially preserving the traditional workflows.
- Question 5 is answered by designing a mini-benchmark featuring typical array access patterns, and a systematic evaluation of different storage alternatives and array data retrieval strategies with respect to those access patterns. The presented benchmark results are complemented with an integrated evaluation of a real-life application from the field of computational biology.

In this Thesis we present the design, implementation and evaluation of *Scientific SPARQL* - a language for querying data and metadata represented using the RDF graph model extended with numeric multidimensional arrays as node values - *RDF with Arrays*. The techniques used to store *RDF with Arrays* in a scalable way and process Scientific SPARQL queries and updates are implemented in our prototype software - Scientific SPARQL Database Manager, and its integrations with data storage systems and computational frameworks.

In *RDF with Arrays*, arrays are used to model massive numeric data, typically ordered along a number of orthogonal axes. The rest of the RDF graph serves to represent different kinds of metadata, for example, a formalized description of an experiment, tools and methods used, parameter cases, provenance, etc. Scientific SPARQL allows combining metadata and numeric data conditions in one query, making it expressive and self-contained, eliminating the need for extra round trips to the server, and giving more freedom to the optimizer to build better execution plans.

The ability to process Scientific SPARQL queries involves scalable storage solutions for numeric multidimensional arrays, and efficient implementation of operations over such arrays. Whenever possible the SciSPARQL query processor accumulates such array operations and accesses the array content in a lazy fashion.

The array content can be physically stored in a variety of external storage systems, including files, relational databases, and specialized array data stores - SSDM offers a simple and flexible Array Storage Extensibility Interface. We have studied the different optimization strategies for the retrieval of array content under a variety of partitioning approaches and access patterns - the performance evaluation we present is based on our mini-benchmark for array queries.

Numeric computations are normally used for filtering or post-processing the retrieved data, and may typically be expressed in a functional way. Existing computational libraries (many of which became de-facto standards in scientific computing, and are often referred for reproducibility of results) can be interfaced and invoked from the query language as *foreign functions*. Cost estimates and alternative directions of evaluation can be additionally specified, in order to aid the construction of better execution plans.

As we expect complex tasks to be formulated as complex queries, good query modularity becomes as important for scalability as good data design and annotation. SciSPARQL allows expressing common query sub-tasks as *functional views*, i.e. SciSPARQL functions defined as parameterized queries. This flexibility is further strengthened by functional language abstractions such as *lexical closures* and *second-order functions*. When it comes to the array processing tasks, SciSPARQL offers *array constructors*, *mappers,* and *condensers* as second-order functions.

An integral real-life evaluation is presented, where the SciSPARQL queries addressing array data in an RDBMS back-end are compared to the equivalent manually written scripts run in pure Matlab - resulting in comparable performance in the general cases. Besides, the unification of array data and Semantic Web styled metadata makes the queries shorter and much easier to write than the equivalent procedural scripts.

SciSPARQL queries are easy to integrate into the common 'sequential' scientific and engineering workflows, involving generation, storage, retrieval, and post-processing of the numeric data, typically based on programs in Java, Python, or C, or scientific computing environments like Matlab. One important benefit is the communication saved, by pushing to the server all the costly processing (e.g. filtering and aggregation) that can be expressed in a query. We also demonstrate how such integration helps to supply and use the descriptive metadata, opening a way to interoperability and collaboration, while in all other aspects the users may keep doing their work the way they already do.

SciSPARQL is a proper superset of the W3C SPARQL 1.1 standard, and its query processor is implemented on the basis of Amos II - a functional object-oriented DBMS. The successful implementation of SPARQL constitutes an important part of this work, and proves the viability of such an approach in general, along with certain semantic mismatches discovered and extensions made. The SSDM system is tested, documented, and available on the project homepage: http://user.it.uu.se/~udbl/SciSPARQL.

The following papers were published in the course of this work:

- ***Scientific SPARQL: Semantic Web Queries over Scientific Data*** [2] introduces the query language, array data model, and in-memory implementation of array operations.
- ***Scientific Analysis by Queries in Extended SPARQL over a Scalable e-Science Data Store*** [3] puts SciSPARQL in the context of a real-world scientific computing application. In order to accommodate for massive numeric data involved, storage extensibility mechanisms and lazy array data retrieval are introduced.
- ***Scientific Data as RDF with Arrays: Tight Integration of SciSPARQL Queries into Matlab*** [4] presents the integration of SciSPARQL queries and updates, facilitating the Semantic Web way of handling metadata about scientific experiments into Matlab and typical computational workflows, demonstrating the benefits and the low cost of adoption of our approach.
- ***Spatio-Temporal Gridded Data Processing on the Semantic Web*** [5] positions Scientific SPARQL as a next unification step in handling geographic and other kinds of gridded coverage data on the web. As an example of a hybrid data store approach suggested, it features SSDM as a SciSPARQL front-end, and the Rasdaman [6] system for scalable storage of massive gridded datasets.

The author of this Thesis is the main contributing author in all research papers listed above.

[1] M.Acosta, M-E.Vidal, T.Lamp, J.Castillo, E.Rickhaus. ANAPSID: An Adaptive Query Processing Engine for SPARQL Endpoints. Proc. *10th International Semantic Web Conference (ISWC'11)*, Bonn, Germany, October 2011

[2] A.Andrejev and T.Risch. Scientific SPARQL: Semantic web queries over scientific data. Proc. *Third International Workshop on Data Engineering Meets the Semantic Web (DESWEB)*, Arlington VA, USA, April 2012

[3] A.Andrejev, S.Toor, A.Hellander, S.Holmgren, and T.Risch. Scientific Analysis by Queries in Extended SPARQL over a Scalable e-Science Data Store. Proc IEEE International Conference on e-Science, Beijing, China, October 2013

[4] A.Andrejev, X.He, T.Risch. Scientific data as RDF with Arrays: Tight integration of SciSPARQL queries into Matlab. Proc. *13th International Semantic Web Conference (ISWC'14)*, Riva del Garda, Italy, October 2014

[5] A.Andrejev, D.Misev, P.Baumann, and T.Risch. Spatio-Temporal Gridded Data Processing on the Semantic Web. *Proc. IEEE International Conference*

on Data Science and Data-Intensive Systems (DSDIS), Sydney, Australia, December 2015

[6] P.Baumann. On the Management of Multidimensional Discrete Data. *VLDB Journal 4 (3), Special Issue on Spatial Database Systems, pp. 401 - 444*, 1994

[7] G.Bell, T.Hey, and A.Szalay. Beyond the Data Deluge, *Science, 323, pp. 1297–1298*, March 2009

[8] T.Berners-Lee, J.Hendler, and O.Lassila. The Semantic Web - *Scientific American, 284 (5) pp. 34–43*, May 2001.

[9] C.Bizer, T.Heath, T.Berners-Lee. Linked Data - The Story So Far. *International Journal on Semantic Web and Information Systems, 5(3), pp. 1-22*, 2009

[10] S.Chaudhuri: An Overview of Query Optimization in Relational Systems. Proc. *18th ACM SIGACT-SIGMOD-SIGART symposium on Principles of Database Systems (PODS'98)*, Seattle WA, USA, June, 1998

[11] L.Dobos, A.Szalay, J.Blakeley, T.Budavári, I.Csabai, D.Tomic, M.Milovanovic, M.Tintor, and A.Jovanovic. Array Requirements for Scientific Applications and an Implementation for Microsoft SQL Server. Proc. *EDBT/ICDT Workshop on Array Databases*, Uppsala, Sweden, March 2011

[12] L.Galarraga, K.Hose, R.Schenkel. Partout: A Distributed Engine of Efficient RDF Processing. Proc. *23rd International Conference on World Wide Web,* Seoul, Korea, April 2014

[13] F.Goasdoue, K.Karanasos, J.Leblay, I.Manolescu. View Selection in Semantic Web Databases. Proc. *38th International Conference on Very Large Data Bases (VLDB'12)*, Istanbul, Turkey, August 2012

[14] J.Gray, D.T.Liu, M.A.Nieto-Santisteban, A.S.Szalay, G.Heber, and D.DeWitt. Scientific Data Management in the Coming Decade. ACM SIGMOD Record, 34 (4), 2005

[15] A.Harth, K.Hose, M.Karnstedt, A.Polleres, K-U.Sattler, J.Umbrich: Data Summaries for On-Demand Queries over Linked Data. Proc. *19th International Conference on World Wide Web*, Raleigh NC, USA, April 2010

[16] T.Hey, S.Tansley, K.Tolle (eds). *The Fourth Paradigm: Data-Intensive Scientific Discovery* - Microsoft Research, 2009. ISBN 978-0-9825442-0-4

[17] Z.Kaoudi, M.Koubarakis, K.Kyzirakos, I.Miliaraki, M.Magiridou, A.Papadakis-Pesaresi. Atlas: Storing, Updating and Querying RDF(S) Data on top of DHTs. *Web Semantics: Science, Services and Agents on the World Wide Web 8 (4) pp. 271-277*, Elsevier, 2010

[18] S.Kotoulas, J.Urbani. SPARQL Query Answering on a Shared-nothing Architecture. Proc. *VLDB Workshop on Semantic Data Management (SemData)*, Singapore, September, 2010

[19] W.Le, A.Kementsietsidis, S.Duan, F.Li. Scalable Multi-Query Optimization for SPARQL. Proc. *IEEE International Conference on Data Engineering (ICDE'12)*, Arlington VA, USA, April 2012

[20] D.Maier, B.Vance. A Call to Order. Proc. *12th ACM SIGACT-SIGMOD-SIGART symposium on Principles of Database Systems (PODS'93)*, Washington DC, USA, May 1993

[21] Community Cleverness Required - *Nature, editorial, 455 (7209) p. 1*, 2008

[22] T.Neumann, G.Weikum. RDF-3X: a RISC-style Engine for RDF. Proc. *34th International Conference on Very Large Data Bases (VLDB'08)*, Auckland, New Zealand, August 2008

[23] T.Neumann, G.Moerkotte. Characteristic Sets: Accurate Cardinality Estimation for RDF Queries with Multiple Joins. Proc. *IEEE International Conference on Data Engineering (ICDE'11)*, Hannover, Germany, April 2011

[24] PostgreSQL. http://www.postgresql.org/

[25] F.Prasser, A.Kemper, K.A.Kuhn: Efficient Distributed Query Processing for Autonomous RDF Databases. Proc. *15th International Conference on Extending Database Technology (EDBT'12)*, Berlin, Germany, March 2012

[26] B.R.K.Reddy, P.S.Kumar. Optimizing SPARQL queries over the Web of Linked Data. Proc. *VLDB Workshop on Semantic Data Management (SemData)*, Singapore, September, 2010

[27] A.Schwarte, P.Haase, K.Hose, R.Schenkel, M.Schmidt. FedX: Optimization Techniques for Federated Query Processing on Linked Data. Proc. *10th International Semantic Web Conference (ISWC'11)*, Bonn, Germany, October 2011

[28] P.Selinger, M.Astrahan, D.Chamberlin, R.Lorie, T.Price. Access Path Selection in a Relational Database System. *Readings in Database Systems*. Morgan Kaufman, 1979

[29] A.S.Szalay and J.Gray. 2020 Computing: Science in an Exponential World. *Nature, 440 (7083), pp. 413–414*, 2006

[30] A.R.Thakar, A.S.Szalay, P.Z.Kunszt, and J.Gray. The Sloan Digital Sky Survey Science Archive: Migrating a Multi-Terabyte Astronomical Archive from Object to Relational DBMS. *Computer Science and Engineering, 5 (5), pp. 16–29,* September 2003.

[31] P.Tsialiamanis, L.Sigirougros, I.Fundulaki, V.Christophides, P.Boncz. Heuristic-based Query Optimization for SPARQL. Proc. *15th International Conference on Extending Database Technology (EDBT'12)*, Berlin, Germany, March 2012

[32] L.Zou, J.Mo, L.Chen, M.T.Öszu, D.Zhao. gStore:.Answering SPARQL Queries via Subgraph Matching. Proc. *37th International Conference on Very Large Data Bases (VLDB'11*), Seattle WA, USA, August 2011