RDF On the Go: An RDF Storage and Query Processor for Mobile Devices

Danh Le-Phuoc, Josiane Xavier Parreira, Vinny Reynolds, and Manfred Hauswirth

Digital Enterprise Research Institute,
National University of Ireland, Galway
Galway, Ireland
{danh.lephuoc,josiane.parreira,vinny.reynolds,manfred.hauswirth}@deri.org

Abstract. We present RDF On the Go, a full-fledged RDF storage and SPARQL query processor for mobile devices. Implemented by adapting the widely used Jena and ARQ Semantic Web Toolkit and query engine, it uses Berkeley DB for storing the RDF data, R-Trees for indexing spatial data indexing and a query processor that supports both standard and spatial queries.
By storing and querying RDF data locally at the user's mobile device, RDF On the Go contributes to improving scalability, decreasing transmission costs, and controlling access to user's personal information. It also enables the development of a next generation of mobile applications. RDF On the Go is available for the Android platform and can be downloaded at `http://rdfonthego.googlecode.com/`.

Keywords: RDF storage, SPARQL query processor, mobile devices

1 Introduction

Mobile devices nowadays are equipped with powerful hardware and software, as well as data connectivity and sensing functionalities such as location and orientation. At the same time, the Semantic Web has been evolving and becoming the preferable choice for representing information, with its Resource Description Framework (RDF) for representing semantic data, and query languages such as SPARQL.

While currently most of the processing of semantic data is done in centralized workstations, there are many advantages in executing this processing on mobile devices: i) by distributing the computation among the large number of existing mobile devices, a great level of scalability can be achieved; ii) if the data generated locally on the mobile device such as the sensors, e-mails, calendar appointments can be queried remotely and only the final results need to be sent to another machine for further processing, the data transmission costs can be dramatically reduced; iii) the local processing of user generated data also contributes to the privacy matter, since raw data no longer need to be shared in order to be analysed.

Having such RDF processing capability on the mobile device also enables a new range of applications such as integrating personal information with the

Linked data cloud, Semantic Mobile Augmented Reality [1] and Mobile Semantic Context-based applications.

Although the processing power is on the rise in mobile devices, it is still far behind workstations, and current implementations of RDF storage and query processors for standard computers can not be directly ported to mobile devices, where these resources are still constrained.

While most of the available semantic based mobile applications have to ship their queries to a remote SPARQL Endpoint, some applications, such as micro-Jena[2], Androjena[3] and i-MoCo[4], do store and query RDF data locally. However, they are tailored to specific application scenarios and offer only limited features.

RDF On the Go is the first application to offer a full-fledged RDF storage and SPARQL query processor. It adapts the widely used Jena and ARQ Semantic Web Toolkit and query engine to the constrained mobile device's environment. The RDF data is stored in the B-Trees provided by the lightweight version of the Berkeley DB for mobile devices[5]. Indexes are created for faster data access, where R-trees are used for spatial data indexing, and our lightweight query processor supports standard and spatial SPARQL queries. RDF On the Go is available for the Android platform and can be downloaded at `http://rdfonthego.googlecode.com/`.

The paper demonstrates our RDF On the Go system. Section 2 describes the implementation in more detail and a short demonstration of the system in action is given in Section 3. Section 4 concludes the paper and provide some thoughts for future work.

2 Implementation Details

For storage of the data, RDF on The Go uses a lightweight version of the Berkeley DB that is suitable for mobile devices, which provides a B-Tree implementation for accessing the RDF graphs. For each RDF node, the system employs *dictionary encoding* [2, 3] where node values are mapped to integer identifiers. This reduces the space required to store each RDF node, since the encoded version of the nodes are considerably smaller than the original ones. Moreover, dictionary encoding also allows faster processing, since integer comparisons are cheaper. Fast lookups are achieved in a two-step approach: first, each triple node is stored in multiple ways with different orderings of the triple elements, similar to [1, 6]. Then indexes are built for every ordering of the triple pattern, as proposed in [4]. To support spatial data, we also use R-Trees indexes for storing URIs that have spatial properties. These indexes will output the bindings for spatial graph patterns which are pipelined to the query execution plan.

Currently we support all standard SPARQL operators except aggregation and sorting operators, and the following three spatial operators: "nearby", "within"

[1] `http://www.w3.org/2010/06/w3car/exploiting_lod_for_ar.pdf`
[2] `http://poseidon.ws.dei.polimi.it/ca/?page_id=59`
[3] `http://code.google.com/p/androjena/`
[4] `http://www.cs.vu.nl/~pmika/swc-2008/i-MoCo-Mobile%20Conference%20Guide-weissEtAl_challenge08.pdf`
[5] `http://www.oracle.com/technetwork/database/berkeleydb/overview`

and "intersect". We plan to extend our SPARQL query processor to support most of the patterns described in [5].

To encourage developers to use RDF On the Go to build their applications, we have adapted the core APIs of Jena [6] and ARQ[7] to the Android environment. This allows the developers to manipulate RDF graphs in the same way as they do with the desktop versions of Jena and ARQ . We also reuse some of the Jena and ARQ packages such as the RDF parser and the SPARQL query parser.

3 Demonstration

In this section we provide a short demonstration of our RDF On the Go system. The current prototype loads sample RDF data set in the N3 format[8] to the RDF store on the mobile device. For demonstration purposes, the system loads a dataset that contains all RDF triples from the LinkedGeoData collection[9] that are about places in Galway, Ireland. The screenshots in figure 1 show the prototype running on a HTC Desire mobile device[10]. Figure 1(a) demonstrates the support for spatial SPARQL queries. It displays a map overlay containing all URIs in the dataset that have geo data properties within a particular area. This constrain is represented by the spatial graph pattern *"?uri spatial:within(lat1 lon1 lat2 lon2)"*, where lat1, lon1 are the latitude and longitude of the left upper corner of the area, and lat2, lon2 are the latitude and longitude of the right lower corner. The spatial query pattern used can also be seen this this figure. Each URI is rendered as a marker on the map. By clicking on a marker details of the URI are shown in figure 1(b).

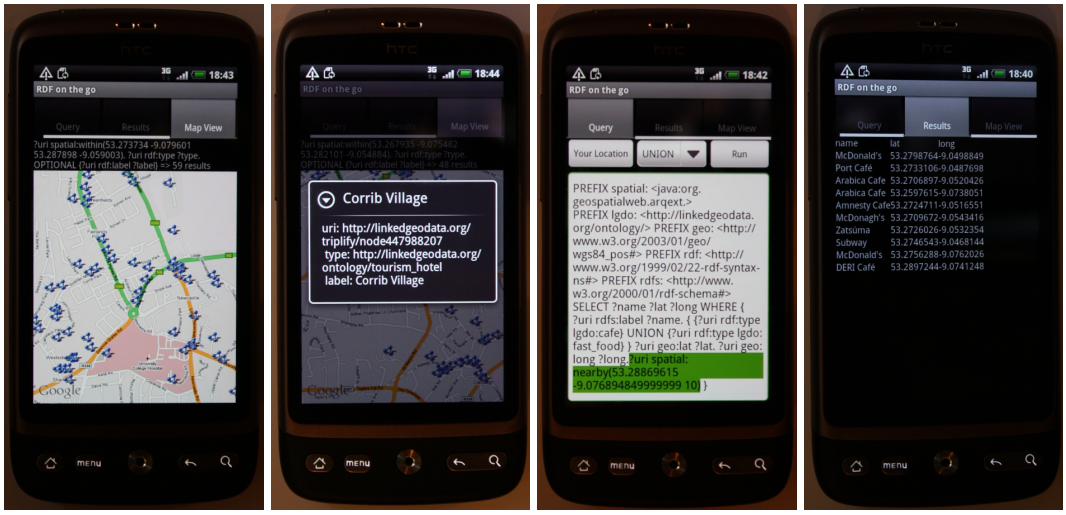
Figure 1(c) shows the interface for standard SPARQL queries. Here we are asking for the 10 nearest 'cafes' or 'fast food restaurants' to the current location. To facilitate entering the query, some common patterns such as OPTIONAL and UNION and the device's current location given by the device's GPS can be added by selecting the corresponding options from a dropdown menu. The results of this query are shown in figure 1(d).

RDF On the Go is available for the Android platform, version 2.1 or newer. The prototype, a video demonstrating the prototype in use, the data collection used in the demonstration and more information are available at `http://rdfonthego.googlecode.com/`.

4 Conclusion

This paper demonstrates the RDF On the Go system, a full-fledged RDF storage and SPARQL query processor for building semantic applications on mobile devices. It not only meets the demand of emerging semantic applications on these devices but also raises many interesting research problems in the areas of data storage and query processing in the context of mobile environments. In the next

[6] `http://jena.sourceforge.net/`
[7] `http://jena.sourceforge.net/ARQ/`
[8] `http://www.w3.org/DesignIssues/Notation3`
[9] `http://linkedgeodata.org`
[10] `http://www.htc.com/www/product/desire/overview.html`



(a) URIs shown on a map (b) Details of an URI (c) Example of a SPARQL query (d) Query results

Fig. 1. Screenshots of the RDF On the Go prototype.

steps, we will focus on efficiency issues, by building a mobile specific query optimizer and more efficient data storage machanism. In context of mobile devices for example, bandwidth and battery are a crucial elements that need to be taken into account in future work.

5 Acknowledgements

This work has been supported by the Science Foundation Ireland under Grant No. SFI/08/CE/I1380 (Lion-2), the European Union under Grant No. FP7-224342-ICT-2007-2 (PECES) and the Irish Research Council for Science, Engineering and Technology (IRCSET).

References

1. D. J. Abadi, A. Marcus, S. R. Madden, and K. Hollenbach. Scalable semantic web data management using vertical partitioning. In *VLDB*, pages 411–422, 2007.
2. J. B. et al. Sesame: An architecture for storing and querying rdf data and schema information. In *Spinning the Semantic Web*, 2003.
3. K. W. et al. Efficient RDF storage and retrieval in Jena2. In *EXPLOITING HYPERLINKS 349*, pages 35–43, 2003.
4. A. Harth and S. Decker. Optimized index structures for querying rdf from the web. In *LA-WEB*, page 71, 2005.
5. J. Pérez, M. Arenas, and C. Gutierrez. Semantics and complexity of SPARQL. *ACM Trans. Database Syst.*, 34(3):1–45, 2009.
6. C. Weiss, P. Karras, and A. Bernstein. Hexastore: sextuple indexing for semantic web data management. *VLDB*, 1(1):1008–1019, 2008.