Pushing Sensor Observations for RDF Stream Processing - Short Paper

Alejandro Llaves, Javier D. Fernández, and Oscar Corcho

Ontology Engineering Group, Universidad Politécnica de Madrid, Madrid, Spain
{allaves,jdfernandez,ocorcho}@fi.upm.es

Abstract. This paper describes ongoing research on the development of a scalable RDF streaming engine. Preparing a dataset for evaluating a stream processing engine is sometimes an underestimated time-consuming task. We explain here the different steps for cleaning, modelling, and transforming a historical dataset about earthquake observations in Spain into a dynamic RDF stream. At the end, we propose open questions related to the efficient processing of RDF streams from the Sensor Web.

1 Introduction

Our current research deals with the design and implementation of an engine that allows efficient processing of complex queries over heterogeneous data streams in near real-time. Data streams may come from multiple sources, being Sensor Web sources the most common ones. During the last years, Sensor Web technologies were established as the technological foundations to allow publishing, updating, and accessing sensor data streams [3]. Sensors may range from environmental ones, normally with slow rate frequencies, to sensors monitoring industrial processes, producing millions of measurements per second.

There is a growing number of applications that depend on the usage of real-time environmental data, and which need to complement the usual three levels of decision making (strategic, tactical, and operational) with real-time decision making. One example would be oil prospecting monitoring, where decisions on the maintenance of a extraction plant may be made on short time slots based on the combination of a set of spatio-temporal data streams coming from different providers, e.g. seismic activity or weather information. Getting information from these streams is complex because of the heterogeneity of the data, the rate of data generation, and the data volume. To tap these data sources accordingly and get relevant information, scalable processing infrastructures are required, as well as efficient approaches to enable data integration and fusion.

2 Extending morph-streams

This work is an extension of previous research on ontology-based access to data stream sources [4,6,5]. Morph-streams, see figure 1, is a system that allows query-

ing data streams using SPARQLStream [6]. Clients register queries in SPARQL-Stream, which extends SPARQL operators to perform continuous queries over streams. The Query Rewriting module translates SPARQLStream to SPARQL and sends the translated query to be processed. R2RML mappings are used to convert data streams to streams of RDF triples.[1] The Query Processing module pulls data from streaming engines, such as Esper or GSN. Then, query results in the form of tuples are sent to the Data Translation module that converts them to RDF triples using R2RML.

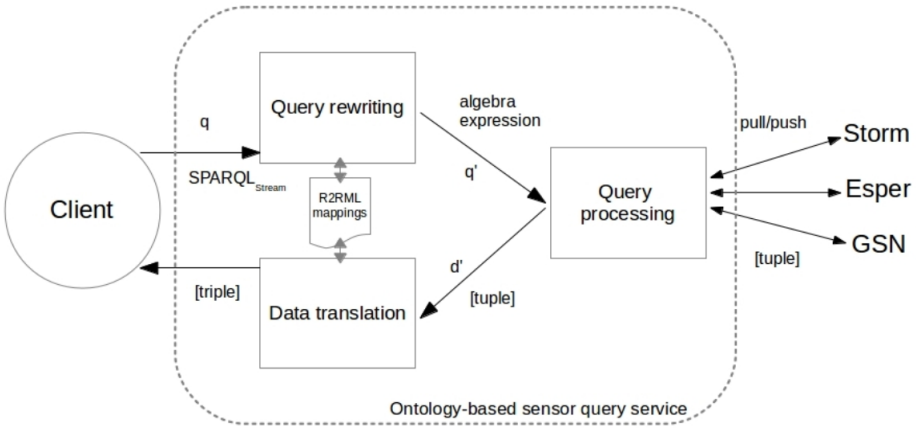

Fig. 1. Morph-streams information flow diagram (adapted from [6]).

In order to enhance the query processing part of morph-streams, features from existing solutions for large-scale stream processing, e.g. Storm,[2] Spark Streaming,[3] or CQELS Cloud [11],[4] are being considered for the extension of the system. At this moment, we use Storm topologies to manage the data stream polling/pushing and the query processing. Storm topologies are formed by spouts and bolts.[5] Spouts are stream sources, whereas bolts are stream processors.

3 Towards Efficient Processing of Queries over RDF Streams

Heterogeneous data streams are generated from different sources, at different rates, and include multiple domains. Our purpose is to build a distributed stream

[1] http://www.w3.org/TR/r2rml/
[2] http://storm.incubator.apache.org/
[3] http://spark.incubator.apache.org/streaming/
[4] https://code.google.com/p/cqels/wiki/CQELSCloud
[5] https://github.com/nathanmarz/storm/wiki/Tutorial

processing engine capable of adapting to changing conditions while serving complex continuous queries. Some sources already generate Linked Data streams [12,2]. Otherwise, we provide a layer serving an ontology-based access to non Linked Data stream sources. Adapters for various input formats, such as CSV or REST APIs, are used to convert heterogeneous streams to RDF. The steps described in this paper deal with the dataset preparation.

To reach an efficient query processing over data streams we will apply ad-hoc query execution planning. Traditional databases include a query optimizer that designs an execution plan based on the registered query and data statistics. In a distributed stream processing environment, there are several aspects to contemplate: changing rates of the input data, failure of processing nodes, and distribution of workload, among others. Adaptive Query Processing (AQP) [1,8] allows adjusting the query execution plan to varying conditions of the data input, the incoming queries, and the system. Additionally, it is used to correct query optimizer mistakes and cope with unknown statistics [1].

4 Earthquakes in Castellón in 2013

The Castor Project aims at increasing the gas storage capacity in Spain. With that purpose, a former oil deposit located 21 kilometres offshore Vinaroz (in Castellón, Spain) is being used as underground gas storage facility. In September 2013, the project activities were stopped because of several earthquake events reported by the Instituto Geográfico Nacional (IGN)[6], see figure 2, and the consequent protests of citizens. Earthquakes are not common in Castellón. Although some of the seismic movements were not perceivable in the coastal towns, there were some noteworthy earthquakes above 4 in the Richter scale that caused the final decision of stopping the gas injection tasks.

We investigate in this use case how to simulate past events using historical data in a streaming fashion. At long term, this will serve to design an adaptive stream processing approach that accounts for unexpected situations.

4.1 Historical Dataset

The original dataset contains earthquake records in Spain during 2013.[7] It was obtained as a text file from the IGN website.[8] IGN also provides an RSS feed with information of earthquakes in the last ten days in Spain. However, we preferred to create a streaming simulation with a higher data volume.

Each earthquake record is a row labeled with an event identifier. The observed properties include the depth of the earthquake in kilometres, its magnitude and the scale type, and its intensity (in the European Macroseismic Scale, EMS-98). Additionally, the dataset contains for each earthquake the date and time at which

[6] http://www.ign.es/

[7] https://www.dropbox.com/s/2y61thcrq85nke0/earthquakesSpain_2013.txt

[8] At http://www.ign.es/ign/layoutIn/sismoFormularioCatalogo.do?locale=en, all parameters left as default, except the date fields: from 2013/01/01 to 2013/31/12.

Fig. 2. Earthquakes in Castellón during September and October 2013.

the earthquake occurred, its geospatial coordinates (latitude and longitude), and a string referring to the location. The coordinate reference system used by official organizations in Spain is ETRS89.[9]

4.2 Cleaning and formatting the earthquakes dataset

We used LODRefine to clean and re-format the earthquakes dataset.[10] LODRefine is a powerful open source tool to deal with messy data. First, we renamed the columns to have more readable headers, e.g. magnitude instead of Mag.. Then, we removed the intensity column because many cells were empty. We also removed all the observation rows that did not include the depth value. The columns date and time were merged to a unique timestamp column and formatted following the W3C datetime guidelines. In the dataset, we noted that different types of magnitudes are used in the observations,[11] but they are consistent for small- and mid-scale earthquakes according to IGN. It is out of the scope of this paper to normalize the different types of magnitudes to a unique scale.

One of the most helpful tools of LODRefine is the application of transformations from a dataset to another one. We worked with a subset of the dataset for October and November 2013 and then, we extracted the transformations (in JSON format) and applied them to the complete dataset. The results are available online in CSV format.[12]

[9] Also known as EPSG:4258, see http://www.opengis.net/def/crs/EPSG/0/4258.
[10] http://code.zemanta.com/sparkica/index.html
[11] Different types of magnitudes explained (in Spanish) here http://www.ign.es/ign/head/sismoTipoMagnitud.do
[12] https://www.dropbox.com/s/wmsah6k6h537s0i/Earthquakes-Spain-2013.csv

4.3 Spatio-temporal modelling of observations

Each line of the CSV file represents an earthquake detected by IGN sensors. To model the earthquake depth and magnitude observations we used the Semantic Sensor Network ontology [7]. Figure 3 shows a concept map for depth observations. Namespaces are defined in the file resulting of the transformation to RDF/Turtle, linked in next subsection.

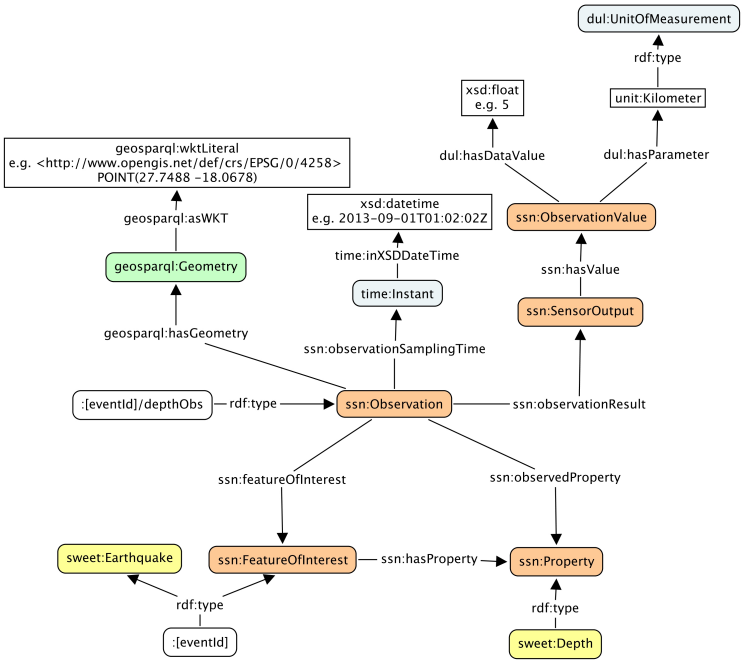

Fig. 3. Concept map for depth observations.

Magnitude observations (*ssn:Observation*) are result of a sensor output (*ssn:SensorOutput*) that produces a float value (*ssn:ObservationValue*). The property observed is a magnitude scale (*sweet:EarthquakeScale*) used to measure the severity of earthquakes (*ssn:FeatureOfInterest*). IGN provides an estimation of the instant (*time:Instant*) at which the earthquake happened (*ssn:observationSamplingTime*). The sensors used to measure seismic activity are not represented in the dataset. For this reason, we attached a geometry point (*geosparql:Geometry*) to each observation (which is also a *geosparql:feature*) that represents where the earthquake happened. Geometries are serialized as Well-Known Text (*geosparql:wktLiteral*). The model for magnitude observations is very similar, but it includes in the observation value the unit of measurement (*dul:UnitOfMeasurement*).

4.4 Transforming a static dataset into a RDF data stream

Morph-streams allows defining R2RML mappings to convert event objects into RDF observations. In this case, we created a R2RML mapping based on the observation model described above.[13] To test the RDF streaming we forked morph-web, a demonstrator of morph-streams, and added a new CSV data source.[14] As a result, earthquake observations are emitted periodically and can be queried using SPARQLStream. The dataset in Turtle format is also available online.[15]

To have a more accurate simulation, we will implement a CSV pusher that adapts the emission time for each row according to the earthquake timestamp and the total demo time. For instance, for a demo of 12 minutes using the 2013 earthquakes dataset, we will have during the ninth and tenth minutes (corresponding to September and October) a higher rate of earthquakes in Castellón.

5 Open Questions and Next Steps

RDF stream processing is a relevant and emerging area of interest that calls for a solution to i) serve continuous queries in near real-time and ii) deal with the complexities of Big Data, i.e. heterogeneity of sources, domains, and flow rates. We are active participants of the W3Cs RDF Stream Processing group, where advances in the field are discussed.[16] In this work, we show preliminary steps on the treatment of a static sensor dataset to be tested in a RDF stream processing engine.

Next steps will address the design of an adaptive approach to process RDF streams. These are some of the questions that we are working on to continue this research line:

- What is the best method to detect spatio-temporal clusters in a stream processing scenario? This information may be useful to design more efficient AQP algorithms.
- Are there implicit features of Linked Sensor Data that make its processing more parallelizable? Can this be used to implement optimized query operators for stream processing?
- Is RDF compression [9,10] a proper solution to alleviate data transmission rates in a Sensor Web scenario?

For future work, we will also apply the results of this research to other domains, such as public transport monitoring or credit card transactions. In these scenarios, the volume of data is higher and the data generation is more frequent.

[13] https://github.com/allaves/morph-web/blob/master/sparqlstream-web/conf/mappings/earthquakes.ttl
[14] Source code available at https://github.com/allaves/morph-web.
[15] https://www.dropbox.com/s/cypbyj6hynf0d62/Earthquakes-Spain-2013.ttl
[16] http://www.w3.org/community/rsp/

References

1. Babu, S., Bizarro, P.: Adaptive Query Processing in the Looking Glass. In: Proceedings of the Second Biennial Conference on Innovative Data Systems Research (CIDR), Jan. 2005 (2005)
2. Balduini, M., Della Valle, E., DellAglio, D., Tsytsarau, M., Palpanas, T., Confalonieri, C.: Social Listening of City Scale Events Using the Streaming Linked Data Framework. In: Alani, H., Kagal, L., Fokoue, A., Groth, P., Biemann, C., Parreira, J.X., Aroyo, L., Noy, N., Welty, C., Janowicz, K. (eds.) The Semantic Web ISWC 2013, pp. 1–16. Springer Berlin Heidelberg (2013)
3. Broering, A., Echterhoff, J., Jirka, S., Simonis, I., Everding, T., Stasch, C., Liang, S., Lemmens, R.: New Generation Sensor Web Enablement. Sensors 11(3), 2652–2699 (Mar 2011)
4. Calbimonte, J.P., Corcho, O., Gray, A.J.G.: Enabling ontology-based access to streaming data sources. In: Proceedings of the 9th International Semantic Web Conference on The Semantic Web - Volume Part I, pp. 96–111. Springer Berlin Heidelberg, Shanghai, China (2010)
5. Calbimonte, J.P., Fernández-Carrera, A., Corcho, O.: Demo paper: Tablet-based visualization of transportation data in Madrid using SPARQLStream. In: Corcho, O., Henson, C., Payam, B. (eds.) Proceedings of the 6th International Workshop on Semantic Sensor Networks co-located with the 12th International Semantic Web Conference (ISWC 2013). pp. 67–70. CEUR-WS, Sydney, Australia (2013)
6. Calbimonte, J.P., Jeung, H., Corcho, O., Aberer, K.: Enabling Query Technologies for the Semantic Sensor Web. International Journal on Semantic Web and Information Systems 8(1), 43–63 (2012)
7. Compton, M., Barnaghi, P., Bermudez, L., Garcia-Castro, R., Corcho, O., Cox, S., Graybeal, J., Hauswirth, M., Henson, C., Herzog, A., Huang, V., Janowicz, K., Kelsey, W.D., Phuoc, D.L., Lefort, L., Leggieri, M., Neuhaus, H., Nikolov, A., Page, K., Passant, A., Sheth, A., Taylor, K.: The ssn ontology of the w3c semantic sensor network incubator group. Web Semantics: Science, Services and Agents on the World Wide Web 17(0) (2012)
8. Deshpande, A., Ives, Z., Raman, V.: Adaptive Query Processing. Foundations and Trends in Databases 1(1), 1–140 (Jan 2007)
9. Fernández, J.D., Martínez-Prieto, M.A., Gutiérrez, C., Polleres, A., Arias, M.: Binary {RDF} representation for publication and exchange (HDT). Web Semantics: Science, Services and Agents on the World Wide Web 19(0), 22–41 (2013)
10. Fernández García, N., Arias Fisteus, J., Sánchez Fernández, L., Fuentes-Lorenzo, D., Corcho, O.: RDSZ : An approach for lossless RDF stream compression. In: In 11th European Semantic Web Conference (ESWC 2014). pp. 1–16. Crete, Greece (2014)
11. Le-phuoc, D., Nguyen, H., Quoc, M., Van, C.L., Hauswirth, M.: Elastic and Scalable Processing of Linked Stream Data in the Cloud. In: Alani, H., Kagal, L., Fokoue, A., Groth, P., Biemann, C., Parreira, J.X., Aroyo, L., Noy, N., Welty, C., Janowicz, K. (eds.) The Semantic Web ISWC 2013, vol. 287305, pp. 280–297. Springer Berlin Heidelberg (2013)
12. Le-phuoc, D., Parreira, J.X., Hauswirth, M.: Linked Stream Data Processing. In: Reasoning Web. Semantic Technologies for Advanced Query Answering, pp. 245–289. Springer Berlin Heidelberg (2012)