Incremental Rule-based Reasoning over RDF Streams: An Expression of Interest

Rehab Albeladi, Kirk Martinez, Nicholas Gibbins

Electronics and Computer Science, University of Southampton, UK
{raab1g09, km, nmg}@ecs.soton.ac.uk

Abstract. This Expression of Interest describes our work in the RDF stream Processing area. We explain our approach to enable rule-based incremental reasoning based on the RETE algorithm.

Processing data streams has received an increasing interest in the semantic web community over the past few years, creating a new research area called RDF Stream Processing. Extensions to the RDF data model and to the SPARQL query language were proposed to enable processing of continuous flow of data, with engines that can provide real-time answers to registered queries [1][2][3][4]. However, little work has been done to enable background reasoning over streaming data [5], which can largely limit the expressiveness of these systems.

We have been working in the stream reasoning area since 2011. We mainly focus on the reasoning aspect, aiming to enable rule-based reasoning over dynamic RDF data as continuously running data flow networks. Our system provides native support for RDF streams using low-level operators directly applied to them. This approach – as opposed to the black-box approach used in many RSP systems- offers maximum optimization opportunities, which have a major impact on the performance of the system in terms of the response time, memory consumption, and completeness of results.

We employ an extension to the RETE algorithm [6] for pattern matching. In this processing model, rules are translated into RETE networks, where nodes represent pipelined, non-blocking operators, and edges represent the streaming data flow. As RETE algorithm materializes partial matches in alpha and beta memories, it provides natural support for incremental reasoning in a data-driven manner. However, unlike the original algorithm, we apply the concept of sliding windows over the alpha and beta memories to control the size of intermediate results. This approach enables stream reasoning on a fine-grained level. All operators are working on a stream-to-stream basis, i.e., they consume streams as input and produce streams as output, which should result in a highly reactive system.

We implemented a prototype stream reasoning engine that receives rules from the user, evaluates them on RDF streams and provides results continuously. Besides background reasoning, our system also supports domain-specific rules. We chose RIF [7] –with a simple extension to represent windows- to express those rules. In the RIF document, the user can specify if any type of background reasoning is required. Our system currently supports RDFS+ reasoning, with plans to support OWL 2 RL.

The system was examined using a real-world streaming data set from the Channel Coast Observatory[1] sensor network. We tested the functionality of the system against a number of rules that range from simple pattern matching to more complex use cases. To evaluate the efficiency of our approach and to emphasize the effects of dynamic reasoning, we compared our system to the static Jena[2] reasoner in terms of processing time. We observed that Jena performs slightly better with small datasets (10^4 triples), while our system significantly outperforms Jena with larger datasets. We are currently running experiments to compare our system to other state-of-the-art RSP systems that support background reasoning, namely Etalis [8] and Sparkwave [9]. We also examined the effects of varying window sizes on the completeness of results and the effects of operator sharing between rules on the memory consumption.

Our next step is to address the optimization problem. As rules and queries are long-running in a streaming environment, and as streaming conditions may change during the life of the running rule, the system needs to support adaptive optimization. To the best of our knowledge, CQUELS [2] is the only RSP engine that supports adaptive optimization. It uses the Eddy operator [10] that enables a fine-grained per-tuple optimization. We argue that this approach would be very expensive for the RDF model, in which an RDF graph –or stream- consists of a large number of small triples compared to the relational model where a smaller number of tuples are composed, usually of more than three attributes. Holding statistics and choosing a route order for every small triple would cause a big overhead. In our ongoing work, we opt for more coarse adaptivity at an intra-query level.

We believe that incremental reasoning is a demanded feature in the RSP area, as indicated in a recent review [11]. We also believe that participating in the RSP workshop and discussing our approach with the community group members will improve and refine our work.

References

1. Barbieri, D., Braga, D., Ceri, S., Della Valle, E. & Grossniklaus, M.: C-SPARQL: SPARQL for continuous querying. In: Proceedings of the 18th International Conference on World Wide Web. ACM, New York, USA (2009)
2. Le-Phuoc, D., Dao-Tran, M., Xavier Parreira, J. & Hauswirth, M.: A Native and Adaptive Approach for Unified Processing of Linked Streams and Linked Data. In: Aroyo, L., Welty, C., Alani, H., Taylor, J., Bernstein, A., Kagal, L., Noy, N., Blomqvist, E. (eds.) ISWC 2011, Part I. LNCS, vol. 7031, p370–388. Springer, Heidelberg (2011)
3. Anicic, D., Fodor, P., Rudolph, S. & Stojanovic, N.: EP-SPARQL: A unified language for event processing and stream reasoning. Proceedings of the 20th International Conference on World Wide Web. Hyderabad, India:ACM (2011)
4. Bolles, A., Grawunder, M. & Jacobi, J.: Streaming SPARQL: Extending SPARQL to process data streams. The Semantic Web: Research and Applications. In: Bechhofer, S., Hauswirth, M., Hoffmann, J. & Koubarakis, M. (eds.). Springer Berlin/Heidelberg (2008)
5. Zhang, Y., Duc, P., Corcho, O. & Calbimonte, J.: SRBench: a streaming RDF/SPARQL benchmark. In The Semantic Web–ISWC 2012, p641-657, Springer, Heidelberg (2012)

[1] http://www.channelcoast.org/

[2] http://jena.apache.org

6. Forgy, C.: Rete: A Fast Algorithm for the Many Pattern/Many Object Pattern Match Problem. Artificial Intelligence, 19, 17--37 (1982)
7. Boley, H., Hallmark, G., Kifer, M., Paschke, A., Polleres, A. & Reynolds, D.: RIF Core Dialect. W3C Recommendation. http://www.w3.org//TR/2010/REC-rif-core-20100622/ (2010)
8. Anicic, D., Fodor, P., Rudolph, S., Sthmer, R., Stojanovic, N. & Studer, R.: Etalis: Rule-based reasoning in event processing. In: Helmer, S., Poulovassilis, A., Xhafa, F. (Eds.), Reasoning in Event-Based Dis- tributed Systems. Vol. 347 of Studies in Computational Intelligence. Springer Berlin, Heidelberg (2011)
9. Komazec, S., Cerri, D. & Fensel, D.: Sparkwave: continuous schema-enhanced pattern matching over rdf data streams. In: Proceedings of the 6th ACM International Conference on Distributed Event-Based Systems. DEBS '12. ACM, New York, NY, USA (2012)
10. Avnur, R. & Hellerstein, J.: Eddies: continuously adaptive query processing. In proceedings of the ACM SIGMOD international conference on management of data, Dallas, Texas, United States (2000)
11. Margara, A., Urbani, J., van Harmelen, F., & Bal, H.: Streaming the web: Reasoning over dynamic data. Web Semantics: Science, Services and Agents on the World Wide Web, 25, 24-44. (2014)