Harnessing a World of Orderings: Streaming Algorithms for the Semantic Web

Emanuele Della Valle[1] and Stefan Schlobach[2]

[1] Politecnico di Milano, Dipartimento di Elettronica e Informazione, Milano - Italy
[2] Vrije Universiteit Amsterdam, Amsterdam - The Netherlands
emanuele.dellavalle@polimi.it, k.s.schlobach@vu.nl

Abstract. Streaming algorithms are able to rapidly perform useful computation with limited memory, if the data are streamed through them in the proper order. They master complexity by adding more information, i.e., the orderings.
Imagine the Web of Data in a new light: it is a sortable entity, where orderings can be enforced easily and logically. Imagine streaming algorithms that scale up to the Web of Data when we enforce orderings first, and then reasoning relies on these orderings. Well, you have just imagined to give birth to a new generation of faster and more scalable Semantic Web technologies.

In 2009, a IEEE Intelligent System column [1] brought to the attention of the Semantic Web community that we leave in a "streaming world!" and, thus, it is worth investigating *Stream Reasoning*, i.e., reasoning upon rapidly changing information. In the last three years, several independent groups have been investigating Stream Reasoning ([2,3,4,5,6,7] to cite, but a few, journal papers that directly referenced the IEEE column) applied to sensor networks, healthcare, financial fraud detection and social media analysis.

Those works share two ideas: *a*) reasoning "on the fly" on data streams while they pass by, and *b*) exploiting the temporal order of the data stream to optimize the computation (e.g., complex event detection [8]). Even if the application of these two ideas to reasoning tasks is new, they are the basis for a well-known class of algorithms: the **streaming algorithms**[3].

This class of algorithms **completely avoids random access to data**. It uses one pass (or a small number of passes) on the data and requires a workspace that is smaller than the size of the data. It includes many algorithms able to perform useful computations by splitting the problem in **sort data first, then compute results**. Some of them show a space complexity upper bound polylog in the size of the input (i.e., $O(polylog(n))$, where n is the size of the input).

[3] In 1998, "Computing on Data Streams" [9] was the first publication formalizing the streaming algorithms, but early works on this class of algorithms root back to the late '70s of the last century, when, for instance, in [10] the first query optimization technique based on estimation of order statistics in the data was presented.

Moreover, even if the streaming algorithms are often not bound on the computation time, the time required for the large majority of the used streaming algorithms is small.

Take, for instance, the streaming algorithms that compute graph statistics, matching in a graph, and random walks [11]: they cope with massive graphs (larger than the current Web of Data) that can only be stored in high capacity storage devices where random access is extremely slow (if compared to the in-memory random access). Notably, to use streaming algorithms, data is not required to be *naturally ordered* (e.g., by recency as in [1]), it has do be sortable by some criteria required by the streaming algorithm that will read it.

Imagine the Web of Data as a sortable entity, where we can enforce orderings easily and logically. Orderings can be omnipresent on the Web of Data. Sometimes resources are directly, and explicitly, described by triples containing **sortable literals**, providing information about the number of inhabitants of cities, ratings of restaurants, longitude and latitude or dates of birth etc. However, most data on the Web of Data also comes with a variety of **implicit ordering measures**, as resources are not just ordered by size or age, but also by meta-properties **such as popularity**[4]. **But not only are resources sortable, so are triples**. There is plenty of literature on how to extend triple with meta-data and annotations, and most of those ideas induce (at least partial) orderings on triples: based on uncertainty, provenance, trust to name but a few [12,13].

Here is the outrageous idea: **if we enforce orderings on data a priori, we can reason faster and we can scale reasoning to the Web of Data!**

Don't you trust us? Let us, for instance, consider the class of streaming algorithm that compute the top-k join of two relations according to a user defined scoring function [14]. If the scoring function can be split into two parts and the two parts can be pushed under the join to sort the two relations, this class of streaming algorithms assure to be able to extract the top-k joins reading only a small fraction of the two *sorted* relations (for an attempt to apply these techniques to SPARQL see [15]).

What if we came up with the notion of *top-k closure of a an ontology w.r.t. a query and a scoring function* (for an attempt see [16])? What if we were able to find streaming algorithm that compute the top-k closure using variants of the top-k join algorithms? Even more, what if we found an approach for computing the top-k closure incrementally for growing values of k (i.e., any-k closure)? **Let us be really outrageous**, what if we conceived an *any-k ontological query answering technique with error probability* ϵ, i.e. **a streaming algorithm for any-time approximate reasoning that computes the right top-k answers to an ontological query with probability at least** $1-\epsilon$? What if we could guarantee that the error probability ϵ decreases

[4] It is well-known that one of the prime criteria for ranking search results in Information Retrieval is based on the trust-worthiness of results, which is calculated using PageRank as a proxy, i.e. the number and importance of web-sites linking to the search result.

with the number of pass on the data? What if the pass on the data could be parallelized and distributed?

It is not difficult to pick-up open challenges that arose in this attempt to study streaming algorithms for the Semantic Web:

- Notion of soundness and completeness for top-k (or any-k) reasoning,
- Knowledge representation techniques for modelling ordered resources or for resources/triples annotated with sortable meta-data,
- Algorithms for enforcing orderings easily and logically,
- Streaming algorithms for reasoning on ontologies expressed in the various OWL2 profiles,
- Streaming algorithms for top-k (or any-k) ontological query answering for the OWL2 profiles (e.g., order-aware materialization for OWL2-RL or order-aware query rewriting for OWL2-QL),
- Streaming algorithms for top-k (or any-k) SPARQL query processing, and
- Parallelization and distribution of the above algorithms.

Investigating streaming algorithms for the Semantic Web, **we can give birth to a new generation of faster and more scalable Semantic Web technologies**. Notably, they can have an high impact: no matter how much memory the computers of the future will have got, they will be able to reason on datasets that cannot fit into it.

Let us close with a real example, whose data size goes much beyond the one of the current Web of Data: Space Situational Awareness, i.e., the identification of objects in space. For many different purposes it is important to know what flies above, e.g. for environmental studies using lasers, but also for commercial companies running satellites. There are many different types of objects flying around space, and many ways of measuring whether there are objects in a particular segment of the sky (think of debris, satellites etc). The data is time and space-bound, complex, massive, and comes from different sources. The applications require any-time approaches, and often almost immediate answers. We can scale up reasoning to address this problem by enforce ordering on time (temporal proximity to the moment of the experiment), space (spatial proximity to the place we want to fire the laser), quality of the measurements, size of the expects objects, etc. and then applying streaming algorithms relying on this orders.

Space Situational Awareness is just started to being addressed; it calls for new research driven, fundamental, approaches. Could the streaming algorithms for the Semantic Web be the solution?

Acknowledgements. We thank Alessandro Bozzon, Stefano Ceri, Ian Horrock, Tony Lee, Andreas Schreiber and Frank van Harmelen for the fruitful discuss that (hopefully) allowed us to propose an *outrageous* idea.

References

1. Della Valle, E., Ceri, S., van Harmelen, F., Fensel, D.: It's a Streaming World! Reasoning upon Rapidly Changing Information. IEEE Intelligent Systems **24**(6) (2009) 83–89

2. Barbieri, D.F., Braga, D., Ceri, S., Valle, E.D., Grossniklaus, M.: Querying rdf streams with c-sparql. SIGMOD Record **39**(1) (2010) 20–26
3. Ruta, M., Colucci, S., Scioscia, F., Sciascio, E.D., Donini, F.M.: Finding commonalities in rfid semantic streams. Procedia CS **5** (2011) 857–864
4. Do, T., Loke, S., Liu, F.: Answer set programming for stream reasoning. In Butz, C., Lingras, P., eds.: Advances in Artificial Intelligence. Volume 6657 of Lecture Notes in Computer Science. Springer Berlin / Heidelberg (2011) 104–109 10.1007/978-3-642-21043-3_13.
5. Anicic, D., Rudolph, S., Fodor, P., Stojanovic, N.: Retractable complex event processing and stream reasoning. In Bassiliades, N., Governatori, G., Paschke, A., eds.: Rule-Based Reasoning, Programming, and Applications. Volume 6826 of Lecture Notes in Computer Science. Springer Berlin / Heidelberg (2011) 122–137 10.1007/978-3-642-22546-8_11.
6. Fergus, P., Haggerty, J., Taylor, M., Bracegirdle, L.: Towards a whole body sensing platform for healthcare applications. In England, D., ed.: Whole Body Interaction. Human-Computer Interaction Series. Springer London (2011) 135–149 10.1007/978-0-85729-433-3_11.
7. Park, J., Shin, Y., Kim, K., Chung, B.S.: Searching social media streams on the web. IEEE Intelligent Systems **25**(6) (2010) 24–31
8. Anicic, D., Fodor, P., Rudolph, S., Stojanovic, N.: Ep-sparql: a unified language for event processing and stream reasoning. In Srinivasan, S., Ramamritham, K., Kumar, A., Ravindra, M.P., Bertino, E., Kumar, R., eds.: WWW, ACM (2011) 635–644
9. Henzinger, M.R., Raghavan, P., Rajagopalan, S.: Computing on data streams (1998) http://www.hpl.hp.com/techreports/Compaq-DEC/SRC-TN-1998-011.pdf.
10. Selinger, P.G., Astrahan, M.M., Chamberlin, D.D., Lorie, R.A., Price, T.G.: Access path selection in a relational database management system. In Bernstein, P.A., ed.: SIGMOD Conference, ACM (1979) 23–34
11. Zhang, J.: A survey on streaming algorithms for massive graphs. In Aggarwal, C.C., Wang, H., Elmagarmid, A.K., eds.: Managing and Mining Graph Data. Volume 40 of The Kluwer International Series on Advances in Database Systems. Springer US (2010) 393–420 10.1007/978-1-4419-6045-0_13.
12. Lopes, N., Polleres, A., Straccia, U., Zimmermann, A.: Anql: Sparqling up annotated rdfs. In: Proceedings of the 9th international semantic web conference on The semantic web - Volume Part I. ISWC'10, Berlin, Heidelberg, Springer-Verlag (2010) 518–533
13. Bonatti, P.A., Hogan, A., Polleres, A., Sauro, L.: Robust and scalable linked data reasoning incorporating provenance and trust annotations. J. Web Sem. **9**(2) (2011) 165–201
14. Ilyas, I.F., Beskales, G., Soliman, M.A.: A survey of top- query processing techniques in relational database systems. ACM Comput. Surv. **40**(4) (2008)
15. Bozzon, A., Della Valle, E., Magliacane, S.: Towards an efficient sparql top-k query execution in virtual rdf stores. In: Proceedings Fifth International Workshop on Ranking in Databases (2011) in Conjunction with VLDB 2011, August 29, 2011, Seattle, WA, USA
16. Schlobach, S.: Top-k reasoning for the Semantic Web: a research manifesto. In: Proc. 1st Ordering and Reasoning workshop (OrdRing 2011) co-located with ISWC 2011. (2011) 52–55