Distributed Reasoning over Ontology Streams and Large Knowledge Base

Raghava Mutharaju
Computer Science and Engineering
Wright State University
3640 Colonel Glenn Hwy., Dayton, OH.
mutharaju.2@wright.edu
http://dase.cs.wright.edu/people/raghava-mutharaju

Keywords:
Distributed Reasoning
Semantic data
Ontology

Background

The velocity (streaming data from traffic and weather sensors) and volume (biomedical and social data) of data is increasing at an exponential rate in this age of Big Data. Making sense of this data efficiently and effectively is one of the most important problems faced by researchers and organizations. Knowledge representation and reasoning can help in this regard. Various efforts in this direction, such as schema.org [1], BBC ontologies [2] and Linked Open Data [3] demonstrate its usefulness.

In the context of Semantic Web, knowledge is represented in either RDF (Resource Description Framework) [4] or OWL (Web Ontology Language) [5], both of which are W3C standards. RDF describes resources and the relationship between them in the form of triples: subject, predicate and object; which in turn can be considered as a directed labeled graph. Large amount of RDF data is available on the Web. For example, there are around 90 billion triples [6] available as Linked Open Data.

In comparison to RDF, OWL can be used to represent more expressive and complex relationships. It is used to capture the knowledge of a particular domain in the form of an ontology. Higher is the expressivity, lower is the tractability of the operations that can be performed on ontologies. With the focus on tractability and scalability, the most recent version of OWL, referred to as OWL 2, provides three profiles or fragments, namely, OWL 2 EL, OWL 2 QL and OWL 2 RL. Although the expressivity of these profiles is limited, it is already sufficient to represent knowledge from diverse domains such as biomedicine and streaming traffic data. OWL 2 supports other intractable profiles.

Reasoning is one of the important operations that can be performed over OWL and RDF data. It is required in order to infer logical consequences and check the consistency of the knowledge base. Reasoning is memory and compute intensive. There are a wide variety of RDF stores (that also support reasoning) ranging from in-memory to on-disk, single machine multi-core to distributed RDF stores. However, this is not the case for OWL reasoners.

Problem Statement

All the existing OWL reasoners are in-memory reasoners and work only on a single machine. Although some of them are efficient, they cannot scale up with the volume and velocity of the data. Automated generation of facts from sensor data and text could lead to very large ontologies. In this case, current reasoners run out of memory and computational resources. A distributed approach to reasoning solves this problem by providing more memory and computational power.

The core research problem here is to find scalable approaches to reasoning algorithms of different expressivity. Following are some of the challenges that should be dealt with.

- Ontology is a set of facts represented in the form of axioms. A reasoning algorithm needs different types of axioms in order to complete the reasoning process. Due to the interdependency among the axioms, distributed reasoning is not an embarrassingly parallel problem.
- Generally, as data processing advances from one phase to the next, amount of data either remains the same or shrinks. But in the case of reasoning algorithms, the amount of data ***exponentially grows*** during processing.
- A scalable and efficient approach to a reasoning algorithm of one type of expressivity might not necessarily be carried over to a reasoning algorithm of different expressivity.

As a concrete plan, we started with the polynomial time reasoning algorithm of OWL 2 EL. The reasoning algorithm consists of repeatedly applying a set of rules on the given ontology until there is no new output. We explored three approaches with mixed results [7] (i) MapReduce, (ii) Random distribution of axioms, but rule triggering is based on axiom type and (iii) Distributed fixpoint iteration, where the distribution of axioms is based on their type and the (one) rule that corresponds to a particular axiom type is applied iteratively. Interestingly, the distributed fixpoint iteration approach, which is considered a rather naïve approach on a single machine gave the best results. DistEL [8], a distributed reasoner, uses this approach. As a next step, we plan to do the following.

1. Extend DistEL so as to cover OWL profiles with perhaps higher expressivity. This distributed reasoning framework should be able to take in any ruleset (EL being one of them), analyze the dependencies among the rules and come up with an efficient distribution strategy.
2. Performance modeling and analysis of the distributed reasoning framework in order to determine the speed-up and network overhead for each distribution strategy.
3. Cost modeling of the framework in order to determine the cluster size for optimum reasoning performance, given an ontology.

Evaluation: We already have traffic data from Dublin city, Miami, Bologna and Rio. In a day, there would be around 6 GB of traffic data, which is around 7 million axioms. After reasoning over this data, it would generate 21 million axioms which is approximately 20 GB in size. We conducted some initial experiments for a single day of data. Now, we will

check the feasibility of using our distributed reasoning framework in real-time traffic monitoring applications over a period of 3 months.

Broader Impacts

Results of this work can have an impact on ontology engineers, Semantic Web research community and users of Semantic Web technologies.

1. **Development of large ontologies.** Research on automated generation of axioms is still nascent in the sense that axioms with only simple relationships are generated. Although automatic generation of complex axioms that are consistent with rest of the knowledge base is hard, another reason could be the lack of reasoners that can handle such large ontologies. Having a distributed reasoner would encourage the ontology engineers to build very large ontologies.
2. **Development of reasoning applications.** There are several interesting streaming data applications such as [9] that use make use of domain knowledge in the form of ontologies. But lack of scalable reasoning approaches is forcing the application designers to look for alternate approaches that may trade accuracy (without domain knowledge) with performance. Successfully integrating our distributed reasoner into the streaming traffic data analysis framework would encourage the development of reasoning applications over streaming data including data from micro-blogs and other forms of streaming sensor data.
3. **Open source code.** Our distributed reasoner would be open-sourced. Approaches tried so far are open sourced and are available at [10,11]. This would encourage others to use and extend our distributed reasoner.
4. **Affect on research community.** We would disseminate our work through publications, talks and demos at conferences.

References

[1] schema.org (http://schema.org)
[2] BBC Ontologies (http://www.bbc.co.uk/ontologies)
[3] Linked Open Data (http://linkeddata.org)
[4] RDF Primer (http://www.w3.org/TR/rdf11-primer)
[5] OWL Primer (http://www.w3.org/TR/owl2-primer)
[6] LOD Stats (http://stats.lod2.eu)
[7] Raghava Mutharaju, Pascal Hitzler, Prabhaker Mateti. Distributed OWL EL Reasoning: The Story So Far. SSWS 2014.
[8] Raghava Mutharaju, Pascal Hitzler, Prabhaker Mateti, Freddy Lécué. Distributed and Scalable OWL EL Reasoning. In the 12th Extended Semantic Web Conference (ESWC 2015).
[9] Freddy Lécué, Simone Tallevi-Diotallevi, Jer Hayes, Robert Tucker, Veli Bicer, Marco Luca Sbodio, Pierpaolo Tommasi. Smart Traffic Analytics in the Semantic Web with STAR-CITY: Scenarios, System and Lessons Learned in Dublin City. Journal of Web Semantics, 2014.
[10] DistEL (https://github.com/raghavam/DistEL)
[11] DQuEL (https://github.com/raghavam/DQuEL)