Very Large Scale OWL Reasoning through Distributed Computation*

Raghava Mutharaju

Kno.e.sis Center,
Wright State University, Dayton, Ohio
raghava@knoesis.org

Abstract. Due to recent developments in reasoning algorithms of the various OWL profiles, the classification time for an ontology has come down drastically. For all of the popular reasoners, in order to process an ontology, an implicit assumption is that the ontology should fit in primary memory. The memory requirements for a reasoner are already quite high, and considering the ever increasing size of the data to be processed and the goal of making reasoning Web scale, this assumption becomes overly restrictive. In our work, we study several distributed classification approaches for the description logic EL+ (a fragment of OWL 2 EL profile). We present the lessons learned from each approach, our current results, and plans for future work.

1 Introduction

Over the years, the efficiency of classification algorithms for the description logic $\mathcal{EL}^+$has constantly improved [3,10,11], so much so that, ELK reasoner [11] can classify SNOMED CT[1], one of the largest biomedical ontologies in 5 seconds. But the improvement has been only in runtime and not space. In a recent study on the performance of reasoners [8], it was noted that, in tableau-based reasoners, memory exhaustion is a known problem. So, in this scenario, performing in-memory computations on a single machine would be problematic for ontologies larger than SNOMED CT.

The amount of available data is always on the rise. We would not be off the mark in saying that there would be ontologies bigger than SNOMED CT very soon. In fact, there is a biomedical ontology named LinkBase, which is thrice the size of SNOMED CT [26,17]. There could be even more bigger ontologies, especially, ontologies with large ABoxes. Even if we consider that the RAM prices are cheap and that might solve the issue, in order to really perform OWL reasoning at Web scale, the current infrastructure that the reasoners are based on, is not sufficient. In this scenario, there is a good possibility of falling short on both memory and computation power. This is where our work on distributed OWL reasoning algorithms is expected to bridge the gap.

* Supervisor: Pascal Hitzler.

[1] Can be obtained from http://ihtsdo.org

P. Cudré-Mauroux et al. (Eds.): ISWC 2012, Part II, LNCS 7650, pp. 407–414, 2012.

Normal Form	Completion Rule
$A_1 \sqcap \cdots \sqcap A_n \sqsubseteq B$	**R1** If $A_1, \ldots, A_n \in S(X)$, $A_1 \sqcap \cdots \sqcap A_n \sqsubseteq B \in \mathcal{O}$, and $B \notin S(X)$ then $S(X) := S(X) \cup \{B\}$
$A \sqsubseteq \exists r.B$	**R2** If $A \in S(X)$, $A \sqsubseteq \exists r.B \in \mathcal{O}$, and $(X, B) \notin R(r)$ then $R(r) := R(r) \cup \{(X, B)\}$
$\exists r.A \sqsubseteq B$	**R3** If $(X, Y) \in R(r)$, $A \in S(Y)$, $\exists r.A \sqsubseteq B \in \mathcal{O}$, and $B \notin S(x)$ then $S(X) := S(X) \cup \{B\}$
$r \sqsubseteq s$	**R4** If $(X, Y) \in R(r)$, $r \sqsubseteq s \in \mathcal{O}$, and $(X, Y) \notin R(s)$ then $R(s) := R(s) \cup \{(X, Y)\}$
$r \circ s \sqsubseteq t$	**R5** If $(X, Y) \in R(r)$, $(Y, Z) \in R(s)$, $r \circ s \sqsubseteq t \in \mathcal{O}s$, $(x, Z) \notin R(t)$ then $R(t) := R(t) \cup \{(X, Z)\}$

Fig. 1. Normal forms and Completion rules in CEL

The remainder of the paper is organized as follows. Section 2 contains some preliminaries. In Section 3, we mention the previous work and how our work differs from it. In Section 4, we present our approaches that we have taken towards tackling this problem and in Section 5, we present our preliminary results followed by our planned work for the future.

2 Preliminaries

Concepts in description logic $\mathcal{EL}^+$are formed according to the grammer

$$C ::= A \mid \top \mid C \sqcap D \mid \exists r.C,$$

where A ranges over concept names, r over role names, and C, D over (possibly complex) concepts. Ontology in $\mathcal{EL}^+$is a finite set of *general concept inclusions* (*GCIs*) $C \sqsubseteq D$ and *role inclusions* (*RIs*) $r_1 \circ \cdots \circ r_n \sqsubseteq r$, where C, D are concepts, n is a positive integer and $r, r_1, \ldots, r_n$ are role names. For the semantics of $\mathcal{EL}^+$please refer [2].

Classification of an ontology is the computation of the complete subsumption hierarchy between all concept names occurring in the ontology. Classification is one of the standard reasoning tasks. Among others, CEL algorithm [4] performs classification of an $\mathcal{EL}^+$ontology. It uses the completion rules in Figure 1. It requires the ontology to be in normal form, where all the axioms should be in one of the forms shown in the left part of Figure 1.

3 Related Work

In order to make reasoning Web scale, algorithms should be scalable. To that extent, various parallel and distributed approaches for classification of OWL fragments and closure of RDFS have been explored. Harmelen et al. use MapReduce

and peer-to-peer network for large scale RDFS reasoning [18,24]. They extend their work to OWL Horst fragment in [7]. Many of their optimization techniques from their work on RDFS reasoning could not be carried over to OWL Horst due to the increased complexity of rules in OWL Horst. As the expressivity increases, the rules as well as the pre-conditions in the rules would be increasingly complex. An embarrassingly parallel algorithm is used in [27] for computing the RDFS closure. In [9], distributed hash tables were used for the computation of RDFS closure. Soma et al. [23] investigate two partitioning approaches for parallel inferencing in OWL Horst. In [25], backward chaining is used to scale up to a billion triples in the OWL Horst fragment. Distributed reasoning of fuzzy OWL Horst has also been investigated in [15].

Stuckenschmidt et al. have used resolution techniques in distributed settings to achieve scalability of various OWL fragments such as $\mathcal{ALC}$ [20] and $\mathcal{ALCHIQ}$ [21]. There have been attempts at achieving distributed reasoning on $\mathcal{EL}^+$profile in [16] and [22], but they do not provide any experimental results. Distribution of OWL EL ontologies over a peer-to-peer network and algorithms based on distributed hash table have been attempted in [5], but they do not provide any evaluation results.

There have also been some successful attempts at making use of the multiple cores on a single machine in order to speed up the classification of ontologies. Haarslev et al. have worked on parallel tbox classification [1] and parallel tableau based description logic reasoner for $\mathcal{ALC}$ [28]. In [13], the authors parallelized the non-deterministic choices inherent in tableau algorithms. Parallelization of tableau algorithm, for $\mathcal{SHIQ}$ has also been attempted in [14], but they haven't provided any evaluation results. In [11], the authors use multi-threading and consequence-based procedure to achieve highly optimized classification runtime. The authors of [19] extend the approach in [11] to parallel ABox reasoning. They were able to compute all ABox entailments for an ontology having 1 million individuals in 3 minutes. But, for this, they require an unreasonably high memory of 60GB on an 8 core processor. With concurrent approaches, it would be possible to improve the efficiency of the classification algorithm, but it would not be possible to achieve scalability. For Web scale reasoning and for very large ontologies, these approaches would suffer from the same memory constraints that were highlighted in Section 1.

Compared to the above approaches, the authors of [6] take a different route. They focus on using secondary memory for classification of $\mathcal{ELH}$ ontologies and were able to classify SNOMED CT in 20 minutes and the RAM used for computations is only 32MB. But this approach lacks the parallelism demonstrated in other approaches. Please note that many of the fragments mentioned here are different from the one that we are interested in, which is $\mathcal{EL}^+$. But this section highlights some of the existing scalable reasoning approaches. For reasonably expressive OWL profiles, we wish to explore the distribution of axioms of the ontology across the cluster and perform parallel computations. We explain our approach further in the next section.

4 Research Problem and Approaches

4.1 Research Problem

Our research problem can be broken down into the following two questions

1. What are the approaches for distributed reasoning of OWL reasoning algorithms; specifically, for profiles $\mathcal{EL}^+$and higher?
2. Demonstrate the need and the validity of the approach for distributed reasoning on a real world use case.

4.2 Research Plan

Our research plan for the above two questions is as follows

Step 1. Start with a relatively less expressive description logic such as $\mathcal{EL}^+$. Explore distributed reasoning approaches for this profile.

Step 2. Choose the distributed reasoning approach which is most appropriate and extend it to more expressive profiles such as $\mathcal{EL}^{++}$[2] and $\mathcal{SROELV}_n(\sqcap, \times)$ [12]. Note that this step might not be a straightforward extension of step 1. It might require additional optimizations and further research.

Step 3. There is an ongoing work in our research center where the Semantic Web Journal website[2] is being upgraded to Drupal 7. The purpose of the upgrade is to have access to the Semantic Web extensions of Drupal 7. If not already present, we plan on developing an OWL reasoner module for Drupal and integrate the distributed reasoning work into it. The Semantic Web Journal website would be backed by an ontology and website content would be annotated appropriately. The website has a constant flow of submissions and by having a reasoner support, we plan on providing semantic search, semantic browsing and semantic content creation. Apart from the journal website content, the reasoner would also access appropriate datasets from Linked Open Data (LOD) cloud. The number of submissions for the journal website as well as the size of LOD cloud keep increasing. So we believe that this would be a very good application to demonstrate the need of having a distributed reasoner.

4.3 Approaches

All the approaches presented are for description logic $\mathcal{EL}^+$. Approaches can be categorised into distributed memory and shared memory.

[2] `http://www.semantic-web-journal.net`

4.3.1 Distributed Memory

MapReduce. Our first attempt was to use the popular distributed framework, MapReduce, for computing classification of $\mathcal{EL}^+$ontologies [16]. We revised the CEL algorithm [4] to suit the key-value format of the data required for MapReduce. In the Map phase, preconditions of the rules are checked and in the Reduce phase, conclusion of the rules are computed. Pros and cons of this approach are given below.

Pros

- Parallelization can be achieved easily.
- Fault tolerance is handled by the framework.

Cons

- In each iteration, duplicates are generated. This makes termination detection hard.
- MapReduce is not suitable if there are dependencies between the data chunks. In CEL completion rules, some of the rules are interdependent.
- It is difficult to filter data in subsequent iterations. For example, ideally, in the next iteration, the algorithm needs to run only on the newly generated data (compared to last iteration).

Distributed Queue. In MapReduce, nodes in the cluster cannot talk to each other. Since there are dependencies among the data chunks, there would be a need for the nodes to talk to each other. Due to this, we replaced map and reduce methods with our custom methods which can talk to other nodes, when required. We also replaced HDFS with a distributed key-value data store. CEL algorithm implementation makes use of a queue mechanism [4] to trigger rule execution. In distributed queue approach, the idea is to take this queue implementation and spread the load across the cluster. Axioms are distributed across the cluster and the queue implementation runs on each node of the cluster. So each node acts as a stand-alone reasoner, which talks to other nodes when required. This approach was not as efficient as we expected it to be due to the following reasons.

- There was a lot of cross communication among the nodes.
- Large ontologies like SNOMED CT generate many R(r)s which makes rule R3 in [4] very slow. This rule slows down the entire operation across the cluster.

Distributed Completion Rules. Instead of distributing the axioms and the queues randomly, we distributed the axioms based on their type. Based on the normal form type [4], each axiom in an ontology can be placed under one of the five types. Now, each node is dedicated to only one type of normal form and runs an appropriate rule on the axioms. Compared to the distributed queue approach, this approach has the advantage of isolating the slowest rule and not letting it affect the processing of other rules. Furthermore, we have split rule R3 into two rules, R3-1 and R3-2 as mentioned in [16]. These two rules run in parallel on separate nodes of the cluster. In order to reduce the

cross communication, we use fixpoint iteration instead of the queue algorithm to process the completion rules. This makes termination detection harder, because, we need to be able to detect that there is no new output across the cluster. This approach was efficient compared to our previous approaches. Some preliminary results are given in Section 5.

4.3.2 Shared Memory

Multi-threaded graph. Apart from the distributed approaches mentioned before, we have also tried shared memory approach. The idea here is to represent all the axioms as a graph and perform parallel traversals[3]. Concepts are represented as nodes and the relationship between concepts as edge. Unlabelled edges represent subclass relation and labelled edges represent role name. This work was done on Cray XMT[4], a massively parallel supercomputer with shared memory architecture.

After representing axioms as graphs, classification would be reduced to the problem of computing transitive closure in the graph with respect to the subclass relation. Cray XMT compiler generates parallelizable version of the code based on the hints that the programmer places in the code. Although Cray XMT provides huge computing power, if there are data dependencies in the code, it is difficult to parallelize that part of the code. We were unable to parallelize the compute intensive parts of the code due to these dependencies. Apart from this, issues like synchronization, deadlocks, hot spots need to be handled by the programmer. Overall, Cray XMT has a steep learning curve and resolving data dependencies is not straightforward. Due to this, the vast computing power of the supercomputer could not be utilized properly.

5 Results and Future Work

Except distributed completion rule approach, none of the other approaches work well on a large ontology like SNOMED CT. We were able to classify SNOMED CT in approximately 50 minutes using a 5 node cluster with the distributed completion rule approach. These are just preliminary results and they can be improved in a variety of ways like making more nodes work on the slowest rule, improving the termination algorithm etc. After further evaluation and optimizations, we plan to publish our results (with complete details) along with rest of the approaches. Then, we would be moving on to Steps 2 and 3 mentioned in section 4.2.

Acknowledgements. This work was supported by the National Science Foundation under award 1017225 "III: Small: TROn - Tractable Reasoning with Ontologies." Any opinions, ndings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reect the views of the National Science Foundation.

[3] Internship work at Clark & Parsia LLC.

[4] `http://www.cray.com/products/XMT.aspx`

References

1. Aslani, M., Haarslev, V.: Concurrent classification of owl ontologies - an empirical evaluation. In: Proceedings of the 2012 International Workshop on Description Logics, DL 2012, Rome, Italy, June 7-10. CEUR Workshop Proceedings, vol. 846, CEUR-WS.org (2012)
2. Baader, F., Brandt, S., Lutz, C.: Pushing the EL envelope. In: Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI 2005), Edinburgh, UK. Morgan-Kaufmann Publishers (2005)
3. Baader, F., Lutz, C., Suntisrivaraporn, B.: CEL — A Polynomial-Time Reasoner for Life Science Ontologies. In: Furbach, U., Shankar, N. (eds.) IJCAR 2006. LNCS (LNAI), vol. 4130, pp. 287–291. Springer, Heidelberg (2006)
4. Baader, F., Lutz, C., Suntisrivaraporn, B.: Efficient reasoning in $\mathcal{EL}^+$. In: Proceedings of the 2006 International Workshop on Description Logics (DL 2006). CEUR Workshop Proceedings, vol. 189 (2006)
5. De Leon Battista, A., Dumontier, M.: A platform for reasoning with owl-el knowledge bases in a peer-to-peer environment. In: Proceedings of the 5th International Workshop on OWL: Experiences and Directions (OWLED 2009), Chantilly, VA, United States, October 23-24. CEUR Workshop Proceedings, vol. 529. CEUR-WS.org (2009)
6. Delaitre, V., Kazakov, Y.: Classifying elh ontologies in sql databases. In: Proceedings of the 5th International Workshop on OWL: Experiences and Directions (OWLED 2009), Chantilly, VA, United States, October 23-24 (2009)
7. Urbani, J., Kotoulas, S., Maassen, J., van Harmelen, F., Bal, H.: OWL Reasoning with WebPIE: Calculating the Closure of 100 Billion Triples. In: Aroyo, L., Antoniou, G., Hyvönen, E., ten Teije, A., Stuckenschmidt, H., Cabral, L., Tudorache, T. (eds.) ESWC 2010, Part I. LNCS, vol. 6088, pp. 213–227. Springer, Heidelberg (2010)
8. Dentler, K., et al.: Comparison of reasoners for large ontologies in the owl 2 el profile. Semantic Web Journal 2(2), 71–87 (2011)
9. Kaoudi, Z., Miliaraki, I., Koubarakis, M.: RDFS Reasoning and Query Answering on Top of DHTs. In: Sheth, A.P., Staab, S., Dean, M., Paolucci, M., Maynard, D., Finin, T., Thirunarayan, K. (eds.) ISWC 2008. LNCS, vol. 5318, pp. 499–516. Springer, Heidelberg (2008)
10. Kazakov, Y.: Consequence-driven reasoning for horn SHIQ ontologies. In: Proceedings of the 21st International Conference on Artificial Intelligence (IJCAI 2009), July 11-17, pp. 2040–2045 (2009)
11. Kazakov, Y., Krötzsch, M., Simančík, F.: Concurrent Classification of $\mathcal{EL}$ Ontologies. 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, pp. 305–320. Springer, Heidelberg (2011)
12. Krötzsch, M., Maier, F., Krisnadhi, A., Hitzler, P.: A better uncle for owl: nominal schemas for integrating rules and ontologies. In: Proceedings of the 20th International Conference on World Wide Web, WWW 2011, Hyderabad, India, March 28 - April 1, pp. 645–654. ACM (2011)
13. Liebig, T., Müller, F.: Parallelizing Tableaux-Based Description Logic Reasoning. In: Meersman, R., Tari, Z. (eds.) OTM-WS 2007, Part II. LNCS, vol. 4806, pp. 1135–1144. Springer, Heidelberg (2007)
14. Liebig, T., Steigmiller, A., Noppens, O.: Scalability via parallelization of OWL reasoning. In: Proceedings of the 4th International Workshop on New Forms of Reasoning for the Semantic Web: Scalable and Dynamic (NeFoRS 2010) (2010)

15. Liu, C., Qi, G., Wang, H., Yu, Y.: Large Scale Fuzzy *pD** Reasoning Using MapReduce. 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, pp. 405–420. Springer, Heidelberg (2011)
16. Mutharaju, R., Maier, F., Hitzler, P.: A mapreduce algorithm for el+. In: Proceedings of the 23rd International Workshop on Description Logics (DL 2010), Waterloo, Ontario, Canada, May 4-7 (2010)
17. Ongenae, F., De Backere, F., Steurbaut, K., Colpaert, K., Kerckhove, W., Decruyenaere, J., De Turck, F.: Appendix b: overview of the existing medical and natural language ontologies which can be used to support the translation process, `http://www.biomedcentral.com/content/supplementary/1472-6947-10-3-s2.pdf`
18. Oren, E., Kotoulas, S., Anadiotis, G., Siebes, R., ten Teije, A., van Harmelen, F.: Marvin: Distributed reasoning over large-scale Semantic Web data. Web Semantics: Science, Services and Agents on the World Wide Web 7(4), 305–316 (2009)
19. Ren, Y., Pan, J.Z., Lee, K.: Optimising parallel abox reasoning of el ontologies. In: Proceedings of the 2012 International Workshop on Description Logics, DL 2012, Rome, Italy, June 7-10. CEUR Workshop Proceedings, vol. 846. CEUR-WS.org (2012)
20. Schlicht, A., Stuckenschmidt, H.: Distributed resolution for alc. In: Proceedings of the 21st International Workshop on Description Logics (DL 2008), Dresden, Germany, May 13-16 (2008)
21. Schlicht, A., Stuckenschmidt, H.: Distributed Resolution for Expressive Ontology Networks. In: Polleres, A., Swift, T. (eds.) RR 2009. LNCS, vol. 5837, pp. 87–101. Springer, Heidelberg (2009)
22. Schlicht, A., Stuckenschmidt, H.: MapResolve. In: Rudolph, S., Gutierrez, C. (eds.) RR 2011. LNCS, vol. 6902, pp. 294–299. Springer, Heidelberg (2011)
23. Soma, R., Prasanna, V.K.: Parallel inferencing for OWL knowledge bases. In: 2008 International Conference on Parallel Processing, ICPP 2008, Portland, Oregon, USA, September 8-12 (2008)
24. Urbani, J., Kotoulas, S., Oren, E., van Harmelen, F.: Scalable Distributed Reasoning Using MapReduce. In: Bernstein, A., Karger, D.R., Heath, T., Feigenbaum, L., Maynard, D., Motta, E., Thirunarayan, K. (eds.) ISWC 2009. LNCS, vol. 5823, pp. 634–649. Springer, Heidelberg (2009)
25. Urbani, J., van Harmelen, F., Schlobach, S., Bal, H.: QueryPIE: Backward Reasoning for OWL Horst over Very Large Knowledge Bases. 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, pp. 730–745. Springer, Heidelberg (2011)
26. van Gurp, M., et al.: Linkbase, a philosophically-inspired ontology for nlp/nlu applications. In: KR-MED 2006, Formal Biomedical Knowledge Representation, Proceedings of the Second International Workshop on Formal Biomedical Knowledge Representation, Baltimore, Maryland, USA, November 8. CEUR Workshop Proceedings (2006)
27. Weaver, J., Hendler, J.A.: Parallel Materialization of the Finite RDFS Closure for Hundreds of Millions of Triples. In: Bernstein, A., Karger, D.R., Heath, T., Feigenbaum, L., Maynard, D., Motta, E., Thirunarayan, K. (eds.) ISWC 2009. LNCS, vol. 5823, pp. 682–697. Springer, Heidelberg (2009)
28. Wu, K., Haarslev, V.: A parallel reasoner for the description logic alc. In: Proceedings of the 2012 International Workshop on Description Logics, DL 2012, Rome, Italy, June 7-10 (2012)