

Parallel Computation Techniques for Ontology Reasoning

Jürgen Bock

FZI Research Center for Information Technologies, Karlsruhe, Germany
bock@fzi.de

Abstract. As current reasoning techniques are not designed for massive parallelisation, usage of parallel computation techniques in reasoning establishes a major research problem. I will propose two possibilities of applying parallel computation techniques to ontology reasoning: parallel processing of independent ontological modules, and tailoring the reasoning algorithms to parallel architectures.

1 Motivation

Scalability is an issue that is subject in many semantic web research discussions. More and more researchers share the awareness that reasoning in its current form will not be able to bear the load of data it is supposed to handle in the near future. A polarising article was published by Fensel and van Harmelen [1], where the authors talk about “10.000 triples just to describe each human, which gives us 100 trillion.” Even though this guess may be intensionally provocative, it has in fact been proven several times in the past that even high estimations of growth were beaten in reality often long before they were predicted to eventuate.

An oberervation is, that available state-of-the-art reasoners do not exploit the benefits of parallel computation techniques, as these are not straightforwardly applied for reasoning calculi. Multithreading or other ways of distributed computation cannot easily be taken care of by the operating system. This is a major problem, since computational power at the level of integrated circuits is about to explore its physical limits by being “down to atoms” concerning conductor and transistor size. However, parallel computer architectures emerge, such as grid, peer-to-peer, or multi-core machines even in home-computing environments. This allows for the overall computational power to grow further, provided that software architecture and algorithms respect this computational paradigm shift. Thus parallel architectures do not inherently speed up all kind of computation, as the workload needs to be split into chunks of independent computations. Current reasoning algorithms do *not* naturally decompose into independent computational chunks.

In fact, it remains an open question whether algorithms currently used in reasoning adapt well to the paradigm shift in computer architecture. In particular it is unclear whether well established tableau algorithms, as widely used in state-of-the-art reasoners, can be parallelised.

One of the efforts towards making reasoning scalable while still guaranteeing soundness and completeness is the restriction to tractable fragments. While early research in description logics was striving to gain more expressivity, recent trends are going towards limiting that expressivity to those language features that preserve tractability¹. Since parallelisation in general does not change the computational complexity of reasoning in highly expressive language fragments, provided a fixed number of processors, this research will focus on parallel reasoning in tractable fragments, as the successful outcome will be highly profitable for semantic web applications.

2 State of the Art

Despite the major concerns about scalability, the research area of parallel reasoning in description logics has not received high attention. An early work is the FLEX system [2, 3] from 1995, which focuses on subsumption checking, classification, and propagation of rules for ABox reasoning. Due to the fine-grained structural algorithms used for TBox reasoning, certain parallel architectures are not well suited. Another work, and to the best of my knowledge, the only one addressing parallel tableau-based DL reasoning was conducted recently by Liebig and Müller at the University of Ulm [4]. Their method directly gears into the tableau calculus by parallelising nondeterministic branches in the tableau, which are mutually independent computational chunks, and thus can be processed in parallel. A longer history than parallel DL reasoning can be observed in the field of parallel processing of Prolog and Datalog. A summarising survey of many efforts in this direction has been published by Gupta *et al.* [5]. The article highlights Or-Parallelism and And-Parallelism in logic programming.

Another way of applying parallelism to ontology reasoning is the simultaneous processing of different *modules* of an ontology. The problem of ontology modularisation has recently attracted increasing interest. Relevant work in this area addresses the problem of identifying partitions of a large ontology, that are self-contained, *i.e.* one can refer to each partition while still preserving relevant context within this partition. The work of Stuckenschmidt and Klein [6] follows a structure-based approach considering the concept hierarchy of an ontology. A work of Cuenca Grau *et al.* [7] is tackling this shortcoming by introducing a modularisation strategy for OWL DL ontologies. The work, however, distances itself from modularisation for reasoning purposes, and explicitly stresses its contribution for modelling purposes. Similarly the recent work of Doran *et al.* [8] focuses on an improvement of ontology reuse by identifying self-contained modules of ontologies. Other work, which is close to the ontology reasoning problem in terms of ontology modularisation was conducted by Cuenca Grau *et al.* on \mathcal{E} -Connections [9] and MacCartney *et al.* [10]. The latter investigates a novel resolution strategy for a first-order theorem prover, by utilising partitioning and

¹ Tractability means computing a reasoning request in at most polynomial runtime, *i.e.* not exceeding the complexity class P.

a message-passing algorithm for reasoning on the first-order knowledge base. Recent work on distributed resolution for description logics has been conducted by Schlicht and Stuckenschmidt [11]. The authors delegate clauses to distributed resolution solvers which process them independently.

As mentioned in the previous section, applying parallel computation techniques to highly expressive description logics does not *per se* make them tractable. Hence the focus for large scale scenarios is on the use of tractable fragments, if soundness and completeness of reasoning requests have to be guaranteed. Tractable fragments² of OWL 2 that have been identified are DL-Lite, \mathcal{EL} , in particular with its more expressive extension \mathcal{EL}^{++} , and OWL-R, which is – in its DL version – based on DLP.

Research in description logic reasoning has led to a variety of available reasoners, resulting from the main research groups in the area. In particular these reasoners are FaCT and its successor FaCT++, Pellet, the Racer system, KAON2 and HermiT. While these systems support highly expressive description logics, there are few implementations, which focus on tractable fragments. These are CEL [12] for the fragment \mathcal{EL}^+ , and QuOnto [13] for answering conjunctive queries in DL-Lite. There is currently no effort to investigate established parallel infrastructures such as grid computation as a basis for reasoning algorithms.

3 Contribution

In this research I will strive for bringing the parallel computation paradigm closer to ontology reasoning techniques. I see two possible ways to benefit from parallel computation techniques. The first one is to modularise ontologies and/or queries to process different parts of the ontology (query) in parallel. The second possible way to benefit from parallel computation is a low-level redesign of the reasoning algorithm to allow for parallel processing.

3.1 Hypothesis 1 – Independent Ontology Modules

In general, ontologies are not structured as a collection of independent modules. Still it is not clear, whether this general statement is reflected in practice. This means that feasibility of this hypothesis depends on a detailed analysis of real-world ontologies.

The problem in reasoning on separate modules of an ontology or a collection of ontologies is, that in the general case ontologies do not behave modular, *i.e.* all parts of an ontology are connected with each other in some way. The challenge for reasoning on different parts of an ontology in parallel is to identify those modules that do not influence each other in terms of conclusions that can be derived by not considering other modules. While previous work on design issues does not heavily rely on complete independence of the modules considered, this does not hold for reasoning, where any interconnection between modules

² <http://www.w3.org/TR/owl2-profiles/>

can potentially require the reasoner to respect the connected modules in total. As this constraint makes modularisation of highly expressive description logics not sound very promising, it might be possible for tractable fragments due to language restrictions.

Respecting the previous work, modularisation in OWL [6, 7, 9], as well as related first-order theorem proving methods [10] and possible combinations of these two points of view will be taken into account.

3.2 Hypothesis 2 – A Parallel Reasoning Algorithm

Most state-of-the-art reasoning systems are based on tableau algorithms or resolution. Until now it is not clear, whether these methods can be parallelised, in particular since they are highly optimised [14, 15] and thus demand a high degree of communication between previously independent chunks of computation (*e.g.* branches in a tableau). However, modern multi-core, shared-memory machines could overcome this problem by providing computation results of one chunk to other processors via a common repository. Following up the state-of-the-art discussion, there are still unsolved problems in adapting reasoning algorithms to recent developments in multi-core architectures:

1. Parallel tableau algorithms only exploit concurrent processing of nondeterministic branches, which can be computed independently without any interaction [4]. This is insufficient in two main scenarios:
 - (a) Some tractable fragments do not allow disjunction or number restrictions and hence cannot benefit from this parallelisation.
 - (b) To increase the number of threads, that can be processed in parallel in order to be applicable to an extremely large number of processors, it will be necessary to consider other kinds of (deterministic) branches which need to communicate efficiently with each other.
2. To keep reasoning in tractable fragments in the theoretical complexity range, proprietary algorithms might be required which are not yet parallelised. There is *e.g.* no implementation for conjunctive query answering in \mathcal{EL}^{++} .
3. Extensively studied parallelisation methods for logic programming applied to description logic reasoning is unsatisfactory regarding the following:
 - (a) Applying logic programming techniques to description logic reasoning mainly improves ABox reasoning. It is not the preferred method for TBox reasoning.
 - (b) Nominals are important in some scenarios and are supported by the tractable fragment \mathcal{EL}^{++} . However, it is unclear how nominals can be dealt with in a datalog based description logic reasoner, such as KAON2.

4 Roadmap

Figure 1 shows the high-level roadmap of this research. The first step of benchmarking existing OWL reasoners has been completed and successfully published

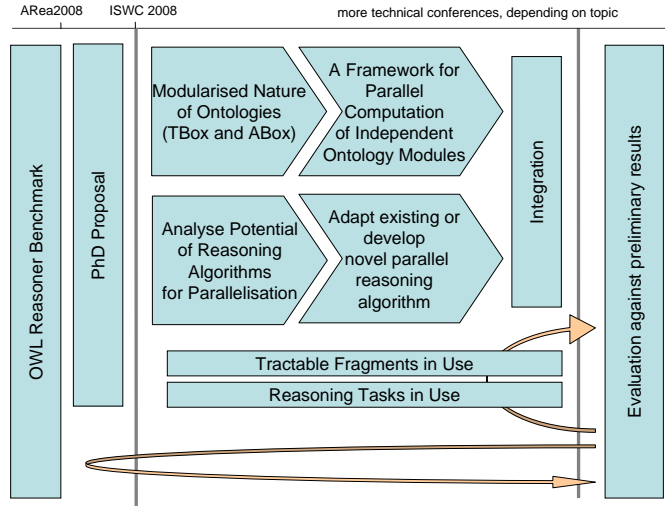


Fig. 1. Roadmap for this PhD research.

at the ARea2008 Workshop [16]. In this work we analysed the ontology landscape and identified major expressivity fragments, which are present in real world ontologies. Following this, we identified which reasoner performs best on which fragment. We distinguished between different reasoning tasks (TBox and ABox), and evaluated scalability w.r.t. ABox size.

The second step of narrowing down the topic has also been completed and resulted in the document at hand. While still providing space for two topics, as discussed in Sect. 3, there are two streams, which will be followed simultaneously. Respecting hypothesis 1, the next step will be to analyse real world ontologies in terms of how they naturally decompose into different partitions or modules. This can be built on our previous analysis [16]. In terms of hypothesis 2, the natural next step is to analyse existing work on parallelising tableau algorithms and conduct early experiments based on modified or newly implemented reasoner prototypes, which exploit parallel computation as discussed in Sect. 3.

In the case of hypothesis 1, following steps will include the setup and implementation of a framework for preprocessing, *i.e.* partitioning of ontologies and invoking the reasoner on these independent partitions. As for hypothesis 2, the outcome would be a modified existing, or entirely novel parallel reasoner, based on fundamental parallel tableaux. Finally, I will investigate the possibilities of integrating the two ideas, *e.g.* by checking whether a modularisation can be identified, which enhances the possibilities of parallelising the algorithm, or adapt the ideas of a parallel tableau algorithm to be applicable to distributed ontology modules. Parallel to the two streams, there will be a comprehensive analysis, of how reasoning tasks and tractable fragments are used in practice.

The final step will be an evaluation of the newly developed solutions w.r.t. the initial reasoner benchmark, as well as to the long-term studies on reasoning tasks and tractable fragments in use, to ensure, that the contribution reflects real world needs in terms of optimised performance on important reasoning tasks and frequent language fragments.

References

1. Fensel, D., van Harmelen, F.: Unifying Reasoning and Search to Web Scale. *IEEE Internet Computing* **11**(2) (March 2007) 94–96
2. Bergmann, F.W., Quantz, J.J.: Parallelizing Description Logics. In: *KI-95: Advances in Artificial Intelligence*. Volume 981 of LNCS., Berlin, Springer (1995)
3. Quantz, J.J., Dunker, G., Bergmann, F., Keller, I.: The FLEX System. Technical Report 124, Technische Universität Berlin, Berlin (December 1995)
4. Liebig, T., Müller, F.: Parallelizing Tableaux-Based Description Logic Reasoning. In: *OTM 2007 Workshops*. Volume 4806 of LNCS., Berlin, Springer (2007)
5. Gupta, G., Pontelli, E., Ali, K.A., Carlsson, M., Hermenegildo, M.V.: Parallel Execution of Prolog Programs: A Survey. *ACM Trans. Program. Lang. Syst.* **23**(4) (July 2001) 472–602
6. Stuckenschmidt, H., Klein, M.C.A.: Structure-Based Partitioning of Large Concept Hierarchies. In: *Proceedings of the 3rd ISWC*. Volume 3298 of LNCS., Berlin, Springer (2004)
7. Cuenca Grau, B., Parsia, B., Sirin, E., Kalyanpur, A.: Modularizing OWL Ontologies. In: *Proceedings of the KCAP-2005 Workshop on Ontology Management*, New York, ACM (October 2005)
8. Doran, P., Tamma, V.A.M., Iannone, L.: Ontology Module Extraction for Ontology Reuse: An Ontology Engineering Perspective. In: *Proceedings of the 16th CIKM*, New York, ACM (November 2007)
9. Cuenca Grau, B., Parsia, B., Sirin, E.: Tableau algorithms for \mathcal{E} -connections of Description Logics. Technical report, University of Maryland Institute for Advanced Computer Studies (UMIACS) (2004)
10. MacCartney, B., McIlraith, S.A., Amir, E., Uribe, T.E.: Practical Partition-Based Theorem Proving for Large Knowledge Bases. In: *Proceedings of the 18th IJCAI*, San Francisco, Morgan Kaufmann (August 2003)
11. Schlicht, A., Stuckenschmidt, H.: Distributed Resolution for ALC. In: *Proceedings of DL2008*. Volume 353 of CEUR Workshop Proceedings. (May 2008)
12. Baader, F., Lutz, C., Suntisrivaraporn, B.: CEL—A Polynomial-time Reasoner for Life Science Ontologies. In: *Proceedings of the 3rd IJCAR*. Volume 4130 of LNAI., Berlin, Springer (August 2006)
13. Acciarri, A., Calvanese, D., Giacomo, G.D., Lembo, D., Lenzerini, M., Palmieri, M., Rosati, R.: QuOnto: Querying Ontologies. In: *Proceedings of the 20th AAAI*, Menlo Park, California, United States, AAAI Press (July 2005)
14. Horrocks, I.: Optimising Tableaux Decision Procedures for Description Logics. PhD thesis, University of Manchester (1997)
15. Motik, B.: Reasoning in Description Logics using Resolution and Deductive Databases. PhD thesis, Universität Karlsruhe (TH), Institut AIFB, Karlsruhe, Germany (2006)
16. Bock, J., Haase, P., Ji, Q., Volz, R.: Benchmarking OWL Reasoners. In: *Proceedings of the ARea2008 Workshop*. Volume 350 of CEUR Workshop Proceedings. (June 2008)