

Fast Selection of Web Services with QoS using a Distributed Parallel Semantic Approach

Luis H. V. Nakamura*, Pedro F. do Prado*, Rafael Mira de O. Libardi*, Luiz H. Nunes*,
Julio C. Estrella*, Regina H. C. Santana*, Marcos J. Santana*, Stephan Reiff-Marganiec †

* *University of São Paulo (USP)*

Institute of Mathematics and Computer Science (ICMC), São Carlos-SP, Brazil

Email: {nakamura, pfprado, mira, lhnunes, jcezar, rcs, mjs}@icmc.usp.br

† *University of Leicester*

University Road, Leicester, LE1 7RH - UK

Email: srm13@le.ac.uk

Abstract—This paper presents a solution to performance issues in the quality of service aware selection of Web services using techniques of parallelism and mechanisms of inference provided by Semantic Web. The results point to a significant improvement in the speed of searching Web services and thus makes the use of semantic resources viable in distributed systems to provide better quality of service to the clients.

Keywords-Web Services; Semantic Web; QoS; Performance;

I. INTRODUCTION

The act of providing differential quality of services (QoS) in a globalized and competitive world may reflect an increase in corporate profits. One of the proposals adopted to classify and select Web services based on QoS attributes is through Semantic Web. The resources of the Semantic Web can provide information of a particular domain in an appropriate and expressive way since that in previous works [1], [2], [3] and [4], was noted that the inference process is costly when considering large amounts of information. Based on these motivations, the aim of this paper is to show a work in progress for a solution to the performance issues in the ontologies inference and selection (search) processes.

This paper is structured as follows: Related work is discussed in Section II. In Section III the ontology and module of this project are briefly described. Section IV presents the environment, planning and results analysis for the performance evaluation. Finally, Section V presents the conclusions and future work.

II. RELATED WORK

In the literature several studies propose optimizations in semantic Web services discovery. These works propose approaches seeking to increase the efficiency and response time. These methods were classified by [4] and are divided into two major categories; *reasoning* efficiency optimization and *matching* efficiency optimization.

In [5], the author proposes the combination of the inference engine *Pellet DL reasoner* with the production rules

engine named Jena. The purpose is to explore the reasoning capabilities of Pellet and also the production rules OWL 2 RL/RDF of Jena resulting in a faster implementation. Another way to improve the performance of these algorithms is proposed in [1] and it consists of parallelization of algorithmic reasoning. In that work, it was firstly proposed to partition the data set and also partition the rules set. It presents a parallel inference algorithm that uses the partition of rules and data to perform classification. This algorithm was more efficient and faster than sequential algorithms.

In this paper we proposed an approach to obtain a performance increase in the inference process through the use of a parallel programming model (“Problem Decomposition”). This approach does not need major changes in algorithms and ontologies already implemented. Moreover, the approach of this paper also involves the use of virtualization capabilities to ensure better performance and scalability.

III. DEVELOPMENT

The module DP-WSO_{nto} (Distributed and Parallel - Web Service Ontology) searches Web services with QoS in parallel. To validate this module an ontology [3] that includes the web service context was used.

At the moment that the module is started, the ontology inference process is executed with the assistance of the inference machine (Pellet reasoner). Pellet analyzes, verifies and makes inferences classifying the ontology elements. It is a complete inference engine and widely used by professionals. However, for those ontologies that use “*Equivalent Classes*” restrictions, it was noted that all the inference processing is performed in a single processing core. Observing the complex source code, the existence of several recursive calls was noted, making it difficult to parallelize. Thus, it was decided to maintain the same sequential code for the inference process. However, a parallel approach to the creation of threads was adopted. The number of threads should be equal to the number of providers, because each thread reads and infers the ontology of a specific provider. So the previous

ontology (within various services of several providers) was copied to other ontologies, each one containing only the services of a single provider. Furthermore, it was decided to use a single thread per core for performance reasons.

In this scenario considering the existence of several inferred ontologies into DP-WSOnto, it presents a new parallel approach for the process of searching for Web services with QoS. In this new approach, multiple instances of the search algorithms are started in parallel, each thread receives three arguments (the ontology URL, an *OntModel* object used to store the ontology already inferred and a *CyclicBarrier* object used by all threads to synchronize the process). Thus, each algorithm instance is reading one of the previously inferred ontologies and at the end of the execution of all threads, a unique and composed result is stored in memory to be used by the client. The proposal uses parallelism through decomposition of data (number of services), and several threads are created to read data from a distributed ontology.

IV. PERFORMANCE EVALUATION

A preliminary performance evaluation presents the difference between the Search Times (Figure 1) using the two approaches (sequential and parallel) with different amounts of services (300, 600 and 1200). The performance evaluation was executed using two search algorithms [3]: (*OntAlgorithmSPARQL* and *OntAlgorithmObject*). Twenty repetitions were made for the each possible combinations (twelve) and adopting 95% of confidence level. The Search Time (ST) results shows that the more services in the ontologies (higher load) the longer is the search times (on average). For the sequential approach, it is noted that using the *OntAlgorithmObject* algorithm the variation in the number of services does not have much influence, because the average response time are overlapped with the confidence intervals. The same applies to the search time for 600 and 1200 services. Moreover, even in the sequential approach, the ST has results statistically different when using the *OntAlgorithmSPARQL*.

When using the parallel approach, the ST results are approximate to each other, but they are statistically different. However, our focus is the comparison among the approaches. The only cases where parallel and sequential approaches are statistically equal were when the *OntAlgorithmObject* algorithm was executed for 300 services. This indicates that the creation of threads causes an *overhead* in the parallel approach execution that have not overcome the sequential execution with 300 services. In the other possible cases, the parallel approach was always better than the sequential.

V. CONCLUSION AND FUTURE WORK

This paper exposed the problem and proposes a solution to improve performance in the inference process of ontologies. Through a parallel and distributed approach,

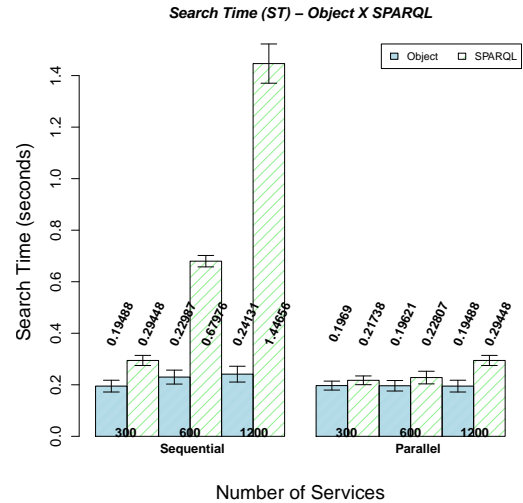


Figure 1. Search Time (ST) Results.

some algorithms were executed in parallel to improve search times for Web services with the appropriate QoS. The inference process had a great improvement in the time taken to provide ontologies inferred and ready to use. In future works, there are plans to use rules to establish the classification constraints. Besides, there are plans to perform new performance evaluations considering other ontologies with a larger number of individuals, and the use of other inference engines.

ACKNOWLEDGMENTS

The authors thank the FAPESP (Processes: 2011/09524-7 and 2011/12670-5) for the support it has given to this work.

REFERENCES

- [1] R. Soma and V. Prasanna, "Parallel inferencing for owl knowledge bases," in *Parallel Processing, 2008. ICPP '08. 37th International Conference on*, sept. 2008, pp. 75–82.
- [2] J. Zhang, X. Yu, P. Liu, and Z. Wang, "Research on improving performance of semantic search in uddi," in *Intelligent Systems, 2009. GCIS '09. WRI Global Congress on*, vol. 4, may 2009, pp. 572–576.
- [3] L. Nakamura, J. Estrella, M. Santana, and R. Santana, "Semantic web and ontology applied to web services discovery with qos," in *Sistemas Computacionais (WSCAD-SSC), 2011 Simpósio em*, oct. 2011, pp. 5–14.
- [4] C. Surianarayanan and G. Ganapathy, "A survey on optimization approaches to semantic service discovery towards an integrated solution," *ICTACT Journal on Soft Computing*, vol. 2, no. 4, pp. 377–383, 2012.
- [5] G. Meditskos and N. Bassiliades, "Dlejena: A practical forward-chaining owl 2 rl reasoner combining jena and pellet," *Web Semant.*, vol. 8, no. 1, pp. 89–94, Mar. 2010.