Knowledge-Based Systems 000 (2016) 1–10

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Short communication

A fine-grained load balancing technique for improving partition-parallel-based ontology matching approaches

Tiago Brasileiro Araújo[a,*], Carlos Eduardo Santos Pires[a], Thiago Pereira da Nóbrega[a,b], Dimas C. Nascimento[c]

[a] Department of Systems and Computer, Federal University of Campina Grande, Brazil
[b] Paraíba State University, Brazil
[c] Federal Rural University of Pernambuco, Brazil

ARTICLE INFO

Article history:
Received 22 March 2016
Revised 18 August 2016
Accepted 19 August 2016
Available online xxx

Keywords:
Ontology matching
Load balancing
Parallel computing
MapReduce
Large ontologies

ABSTRACT

Currently, the use of large ontologies in various areas of knowledge is increasing. Since these ontologies can present overlapping of content, the identification of correspondences between entities becomes necessary for different tasks, for example, data integration and data linkage. Matching large ontologies is a challenge since it involves an excessive number of comparisons between entities which leads to high execution times and requires a considerable amount of computing resources. This work proposes a fine-grained load balancing technique which can be applied to Partition-Parallel-based Ontology Matching (PPOM) approaches. A PPOM approach partitions the input ontologies into sub-ontologies and executes the comparisons between entities in parallel (for instance, using MapReduce). In this sense, the fine-grained load balancing technique aims to guide the even distribution of comparisons among the nodes of a cluster infrastructure. Experimental results indicate that the proposed load balancing technique is able to reduce the overall execution time of PPOM approaches.

1. Introduction

Currently, there is a vast amount of information on different areas of knowledge, e.g., health, weather, and science. One way to describe the knowledge of a particular area is through the use of (domain) ontologies [1–3]. In an ontology, the information is represented as a set of concepts, instances, and relations between concepts. A concept corresponds to a group of individuals sharing the same properties. Properties are binary relations used to define relationships between individuals or between individuals and data values. Instances are the specific representation of a concept. Relations specify in what sense a concept is related to other concepts in the ontology.

Since ontologies can present overlapping of content, the (semi-)automatic identification of similarities between entities (i.e., concepts, properties or individuals) becomes necessary for different tasks, e.g., data integration and data linkage. The process which determines the pairs of similar entities (correspondences) between ontologies is called Ontology Matching (OM) [1,4]. In traditional OM, two ontologies O_1 and O_2 (normally modelling the same or similar domains) are received as input, and a pairwise comparison of entities is performed following the Cartesian product, i.e., all entities of ontology O_1 are compared with all entities of O_2. Similarly to [1,3,5,6], in this work, we have restricted the correspondence identification to concepts.

To calculate the similarity between two concepts, multiple matching algorithms (matchers) are applied. The matchers exploit lexical, structural or semantic properties of concepts using different similarity/distance functions [1]. Therefore, matchers tend to present different computational complexities. In general, each matcher explores an specific property of the involved concepts, e.g., label and *is-a* relations. Thus, several comparisons are performed for each pair of concepts, not only due to the use of multiple matchers, but also because a single matcher can perform diverse comparisons. For instance, a matcher that compares two concepts based on their subconcepts needs to match the whole set of subconcepts of both concepts. In this work, we apply linguistic matchers (e.g., matchers that explore concept labels) to determine the similarity between concepts.

The similarity value, which is produced by each matcher, varies between 0 (no similarity) and 1 (complete similarity). For each pair

* Corresponding author. Tel.: 0558333337323.
E-mail addresses: tiagobrasileiro@copin.ufcg.edu.br, brasileiroaraujo@gmail.com (T.B. Araújo), cesp@dsc.ufcg.edu.br (C.E.S. Pires), thiagonobrega@uepb.edu.br (T.P. da Nóbrega), dimascnf@uag.ufrpe.br (D.C. Nascimento).

http://dx.doi.org/10.1016/j.knosys.2016.08.017
0950-7051/

of concepts, the similarity value is generated through aggregation measures (e.g., average and weighted average) of the partial similarity values produced by the matchers [1]. If the similarity value is greater than a similarity threshold, the pair of concepts is considered as a correspondence. The correspondences form the alignment between O_1 and O_2. Quality metrics (e.g., *F-measure*) are used to measure the effectiveness of the alignment.

In general, domain ontologies contain thousands of concepts, being considered large ontologies (10,000+ concepts) [4]. When dealing with large ontologies, the traditional OM process (Cartesian product) is time-consuming and memory-intensive [6,7]. Therefore, to optimize the OM process, approaches for reducing the search space (i.e., minimizing the amount of comparisons to be performed) and/or execute the OM process in parallel [8] can be applied. In Partition-based OM [9], to minimize the match search space, each input ontology (O_1 and O_2) is individually partitioned and the concepts are divided into the sub-ontologies (partitions), such that there is no overlap between the sub-ontologies. After that, the sub-ontologies of O_1 and O_2 with a certain degree of similarity are paired and the concepts inside each pair of sub-ontologies are matched following the Cartesian product [4]. In turn, parallel-based OM approaches [5,8,10] aim to reduce the overall execution time by distributing the pairs of concepts among the various resources (e.g., computers or virtual machines) of a computational infrastructure. In this work, we consider the Partition-Parallel-based Ontology Matching (PPOM) approach, which combines both techniques in order to improve even more the efficiency of the OM process in the large ontologies context.

Particularly, parallel-based OM approaches usually tackle two major problems: load imbalancing and high replication rate [10]. Load imbalancing occurs when some nodes execute comparisons for a long time while other nodes remain idle. This problem directly influences the efficiency of the whole OM process, since the slowest node dominates the overall execution time. One reason for load imbalancing is that partition-based OM approaches can produce pairs of sub-ontologies which generate different amounts of pairs of concepts to be compared. Thus, if the workload distribution among the nodes is based on the amount of pairs of sub-ontologies, a node will probably receive more pairs of concepts than others. In addition, since the amount of comparisons for each pair of concepts varies, a node may receive pairs of concepts which results in much more comparisons than the other nodes.

To parallel the partition-based OM approaches, for each pair of sub-ontologies, the concepts of the first sub-ontology must be spread among the nodes while the concepts of the other sub-ontology must be replicated and sent to the nodes which contain the concepts of the first sub-ontology. In the worst case, the concepts of a sub-ontology will be spread to all nodes, which forces all concepts of the other sub-ontology to be replicated to all nodes. Although the replication of concepts is needed, a high replication rate reduces the efficiency of parallel-based OM approaches since it increases the volume of data transmitted among the nodes. Hence, the replication rate should be minimized by replicating concepts as few as possible.

In this work, we propose a fine-grained load balancing technique which can be applied to PPOM approaches. To guide the even distribution of comparisons among the nodes, the fine-grained technique takes into account the amount of comparisons to be performed in each pair of concepts. Furthermore, to reduce the amount of replicated concepts for each pair of sub-ontologies, the proposed technique minimizes the amount of nodes to which the concepts are sent. The technique is applied to the PPOM approach proposed in [8] to be evaluated regarding efficiency and effectiveness using real and synthetic ontologies.

2. Parallel-based ontology matching

Usually, the OM process can be parallelized in two ways, according to the way matchers are executed [5]: inter-matcher and intra-matcher. In inter-matcher parallelization, each node (e.g., computer, virtual machine or computer core) of a distributed computing infrastructure (e.g., a cloud or a cluster of machines) executes a particular matcher. Each node receives all pairs of concepts, where the matcher (particular of the node) executes the comparisons. In intra-matcher parallelization, each node executes all matchers. The pairs of concepts are distributed among the available nodes and all matchers perform the comparisons sequentially.

A major disadvantage of inter-matcher parallelization is related to the different complexities of matchers. The complexity of a matcher refers to the computational cost to execute a comparison. For a homogeneous infrastructure (nodes with the same configuration), the node which hosts the most complex matcher will probably execute for a long time while the other nodes (hosting less complex matchers) will likely remain idle. This imbalance directly affects the overall execution time of the OM process. This problem is out of the scope of this current paper. In this work, we assume that all nodes execute the same set of matchers (intra-matcher), i.e., all nodes are similar regarding the complexity of the executed matchers. However, intra-matcher parallelization introduces two relevant problems: high replication rate (which is discussed in Section 1) and load imbalancing caused by data skewness [11].

For parallel-based OM approaches, which use MapReduce as a programming model [12], the load imbalancing problem can happen both in the map or reduce phases (*map-skew* and *reduce-skew*, respectively) [11]. However, since the comparisons between concepts are executed in the reduce phase, this phase is the most affected one. The problem can occur due to three reasons [11,13]: (i) the skew caused by an uneven distribution of pairs of concepts to tasks; (ii) the skew caused by some pairs of concepts taking much longer to be compared than others; and (iii) the computing infrastructure with heterogeneous nodes (different configurations). To reduce the *reduce-skew* problem, the load balancing technique proposed in this work provides a solution to minimize problems (i) and (ii).

3. Related work

Parallel-based ontology matching has been addressed in several works. In [5,6], the authors proposed their own parallel architecture which is divided into three modules: data (extracts the information from ontologies), workflow (forwards the comparisons to available resources) and match (compares the concepts and determines the correspondences). In [8,14], MapReduce was used as a programmable solution to parallel the comparisons between concepts. The approach proposed in [14] converts the input ontologies into virtual documents and cyclically analyzes the similarities between documents to determine the correspondences. In [8], the proposed approach combines partition and parallel techniques (PPOM). It employs the intra-matcher strategy to parallel the OM process. Input ontologies are partitioned and the resulting pairs of sub-ontologies are provided as input to the PPOM approach. However, the approach does not apply any load balancing technique to distribute evenly the comparisons.

Since MapReduce offers only a straightforward technique to distribute the workload among the nodes [15], we apply a fine-grained load balancing technique to guide an even distribution of the workload. Our technique analyzes the amount of comparisons to be performed in each pair of concepts, based on the information (amount of concepts and properties) provided by an ontology partitioning algorithm.

Please cite this article as: T.B. Araújo et al., A fine-grained load balancing technique for improving partition-parallel-based ontology matching approaches, Knowledge-Based Systems (2016), http://dx.doi.org/10.1016/j.knosys.2016.08.017