Big Data approach applied to Graph based Information Request

Claire Laudy and Christophe Gouguenheim

Thales Research and Technology, France
{christophe.gouguenheim,claire.laudy}@thalesgroup.com

Abstract. The automation of information fusion and request is a challenging issue. We previously developed a framework based on graph structures and algorithm for soft data management. The graph request algorithm relies on the search for matching between sub-graphs, which is a difficult problem, especially on large graphs. We present a partially parallelized version of the request algorithm, based on the MapReduce approach. We tested the parallel algorithm to request a large information network. The network is derived from the Global Terrorism Database and contains more than 20,000,000 nodes linked together.

1 Introduction[1]

Within their day to day work, intelligence analysts pile up a large amount of information. The different information items are linked through a big information network. One of the issue for using such an amount of information is to be able to access relevant parts of it efficiently. Information fusion and request functions are asked for by intelligence analysts.

We previously developed a framework based on graph structures, graph algorithm and similarity measures for information fusion [1]. This paper focuses on the specific issue of querying information on a large network. The query algorithm relies on a sub-graph isomorphism algorithm. Because of the computational complexity of the sub-graph isomorphism problem, managing very big graphs is a challenge. We present a version of the information query algorithm, based on the MapReduce principle, in which the node to node matching part of the process is parallelized.

The second section of this paper is dedicated to the presentation of the data query algorithm. The third section presents the parallelization of this algorithm, which enables the management of big data graphs. Section four is dedicated to our experimentations and in the last section, we conclude and present future work.

[1] Our research has received funding from the European Union Seventh Framework Program (FP7/2007-2013) iSAR+ FP7-SEC-2012-1-312850

2 Graph based Information Query

We use Basic **Conceptual Graphs** (**BG** [2], [3]) to represent information. BG are bipartite graphs containing concept and relation nodes. Figure 1 gives an example of a conceptual graph. The rectangular boxes represent concept nodes and the ovals represent relation nodes.

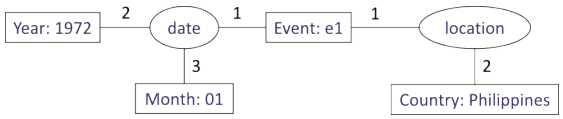

Fig. 1. Example of a conceptual graph

The **concepts** represent the "things" or entities that exist. For instance, in Figure 1 the concept `[Country:Philippines]` represents an instance of a `Country` object that has `Philippines` as value. The **relation nodes** indicate the relations that hold between the different concepts of a situation.

The types of concepts are organized into a hierarchy. Therefore, a specialization/generalization relation may be defined between several graphs ([4]). Answers to a query graph are sub-graphs of the data graph that are more specific than the query graph.

The query algorithm relies on a generic sub graph matching algorithm, which itself uses specific fusion strategies [1]. The graph matching component is in charge of the structural consistency between the query and information graphs.

The fusion strategy part is made of compatibility functions over elements of the graphs. They enable the customization of the generic algorithm according to the context in which it is used. Within the query function, we use a *subsomption* strategy and a whole-structure conservation mode.

The decision problem of whether two *graphs* match is well studied and is in NP [5]. The problem of finding matching *subgraphs*, known as the *subgraph isomorphism* problem, is known to be NP-complete and difficult to solve in parallel [6]. We propose hereafter an algorithm for the query function.

3 Query algorithm

The input of the query function are two graphs. The *data graph* is a big network, while the *query graph* is a relatively small one. Our approach is to split up the subgraph matching process. A first phase manages the node to node comparisons of the query and data graphs. This first step is parallelized. A second step is in charge of preserving the structure of the graphs.

3.1 Hadoop : a framework for processing “big data”

Hadoop [7] is an implementation of a distributed file system along with a model for performing distributed data processing on this file system. Hadoop's file system enables the partitioning of data across many nodes.

The data processing model is called MapReduce, and enables parallel processing of large data sets on multiple processing nodes. Users specify a map function that processes a key/value pair to generate a set of intermediate key/value pairs, and a reduce function that merges all intermediate values associated with the same intermediate key.

3.2 Parallel node to node comparisons

The two graphs (data and query) are cut into small pieces that are pairwise compared (see Figure 2). The graphs are cut around the relations nodes. Each subgraph contains a relation and the linked concepts.

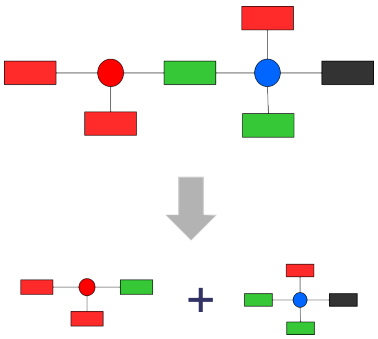

Fig. 2. Decomposition of a ‘query’ graph

The *map* step looks for all the candidate images for each subgraph of the query in the set of subgraphs of data (see Figure 3). The master node of the Hadoop cluster divides the data graph into subgraphs. It then distributes the compatibility test of the data relations against the query graph to the worker nodes of the cluster. The worker nodes process the compatibility test and produce, if the test is achieved, output records as pairs of compatible query and data subgraphs.

During the *reduce* step, the master node of the processing cluster collects the records produced during the map phase. It combines them so as to associate each subgraph of the query with the list of compatible subgraphs of the data.

3.3 Graph structure preservation

Once the candidate answers to each subgraph of the query are processed, all the combinations of answer subgraphs are provided, preserving the original structure

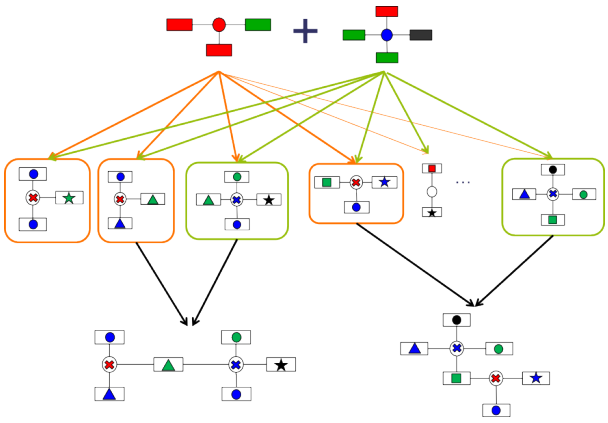

Fig. 3. Finding candidate images and reconstructing answer graphs

of the data graph (see Figure 3). The candidate answer subgraphs are assemble one with another, if and only if their association respects the structural constraints of the initial data graph. The connectivity between the relations through the concept nodes is checked.

4 Experimentations

The global terrorism database[2](GTD) is an open-source database containing over 113,000 descriptions of terrorist events that occured worldwide between 1970 and 2012. It includes information on 45 to more than 120 variables for each case. For each event, information is available on the date and location (region, country, state, city) of the incident, perpetrator group name, tactic used in attack, the weapons used and nature of the target, the number of casualties, etc.

Within our experimentations, we first imported the data contained in the GTD into an information network that contains more than 10,000,000 concept nodes linked through 13,560,000 relation nodes. We launched several queries on the network made of the GTD data. These queries are patterns of terrorists events containing some known and unknown values. The figure 4 depicts an example of such a query. It queries the network for information on years, types of weapons used and number of persons killed in all the terrorist events perpetrated by Tamils and that occurred in Sri Lanka. This query gave us 2935 answers under the form of graphs describing these terrorists events. This query gave us 2935 answers under the form of graphs describing these terrorists events.

In order to test the scalability of our implementation, we artificially increased the size of the information network by generating 7 terrorist events for each event described in the GTD. This generated a text input file of 6.64 Mo that describes the 1,560,000 events. The queries were sent on a single core CentOS 5.7 machine

[2] `http://www.start.umd.edu/gtd/`

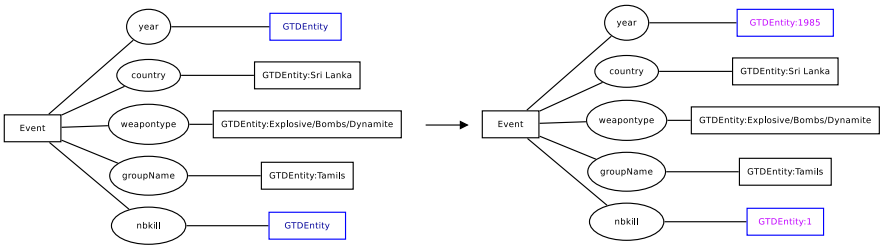

Fig. 4. Example of query and answer on the information network

with 3Gb RAM and 20Gb disk memory. Answers were obtained in less than 1 minute.

5 Conclusion and Future work

In this paper we present the implementation of a distributed algorithm for graph based information fusion. The algorithm relies on the Map/Reduce principle, which enables the distribution of part of the process on several processing nodes. The implementation uses the Hadoop framework.

Tests were conducted on an information network made of the data contained in the Global Terrorist Database. The tests consisted in querying specific patterns of terrorist events. The results obtained during this first experimentation phase are promising. The information network is still too small for the process to be distrubuted on several nodes. Future work will include use cases with real 'big data' information network.

References

1. C. Laudy, *Semantic Knowledge Representations for Soft Data Fusion — Efficient Decision Support Systems - Practice and Challenges From Current to Future.* Chiang Jao Publisher, 2011.
2. J. F. Sowa, *Conceptual Structures: Information Processing in Mind and Machine.* Reading: Addison-Wesley, 1984.
3. M. Chein and M.-L. Mugnier, *Graph-based Knowledge Representation: Computational Foundations of Conceptual Graphs.* Springer, 2008.
4. S. Fossier, C. Laudy, and F. Pichon, "Managing uncertainty in conceptual graph-based soft information fusion," in *Proceedings of the 16th International Conference on Information Fusion, FUSION 2013, Istanbul, Turkey, July 9-12*, 2013, pp. 930–937.
5. B. D. McKay and A. Piperno, "Practical graph isomorphism, II," *Journal of Symbolic Computation*, vol. 60, pp. 94–112, Jan. 2014. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S0747717113001193
6. T. Plantenga, "Inexact subgraph isomorphism in MapReduce," *Journal of Parallel and Distributed Computing*, vol. 73, no. 2, pp. 164–175, Feb. 2013. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S0743731512002559
7. T. White, *Hadoop: The Definitive Guide*, 1st ed. O'Reilly Media, Inc., 2009.