On Web-scale Reasoning

Urbani, J.

2013

document version
Publisher's PDF, also known as Version of record

Summary

On Web-scale Reasoning

The Semantic Web is an extension of the current World Wide Web, where the meaning of information can be interpreted by machines. In the Semantic Web, the information is stored as a set of subject-predicate-object statements.

Currently, there are billions of such statements publicly available on the Web. These statements describe information on a very wide range of domains, from biomedical to government information. URIs are often used to identify concepts to ensure unambiguity and to favor reuse in a distributed setting like the Web.

One of the advantages of storing the information using Semantic Web technologies is that machines can "reason" using the data and infer new information. This process, which is broadly referred as "reasoning", is becoming more and more challenging due to the exponential growth of the availability of information on the Web. While at the beginning of 2009 the Semantic Web was estimated to contain about 4.4 billion of such statements, one year later, the size of the Web had tripled to 13 billions and the current trend indicates that this growth rate has not changed.

The research question that is tackled in this thesis is: "How can we perform reasoning to enrich query results over a very large amount of data (i.e. on a web-scale) using a parallel and distributed system?"

The thesis is divided in two parts: in the first reasoning is performed by applying a set of rules over the entire input, and the purpose is to derive every possible conclusions that we can obtain using the input data. In the second part, we change the purpose of our task to derive only conclusions that might be relevant for the user queries.

In the first part we exploit the MapReduce programming model to perform this task on a very large scale and consequently achieve high performance. In the first chapter we present a series of MapReduce-based reasoning algorithms. In the second chapter, we use the same programming model to compress the input a more compact form so that it could be processed more efficiently. In the third chapter, we use Pig, which is a language built on top of MapReduce, to encode large SPARQL queries execution. In this way, we provide for a complete inference and querying system using the same system architecture and programming model.

In the second part we swift our focus to reasoning that is invoked when the user is querying the knowledge base. In this use case, we cannot rely on MapReduce because of the high latency required to launch a MapReduce job. Therefore we introduce a new technique of hybrid reasoning where only a small part of the entire derivation is computed beforehand. Then, this small derivation is used at query time to reduce the computation of reasoning at query time. In the fifth chapter, we analyze our technique of hybrid reasoning from a theoretical perspective and verify whether the approach is sound and complete. After this, in Chapter 6, we describe a distributed and parallel prototype implementation of this technique and analyze the performance

on the DAS-4 cluster using a standard benchmark tool.

In the last chapter of this thesis, we extract from the technical contribution presented in the previous chapters a number of principles, which we call "laws", that are arguably holding on current data and that responsible for the performance that we achieved in our experiments. These laws can be used to better understand the properties of current web-scale reasoning and be used to drive further research on this topic.