Chapter 92
Distributed Top-K Join Queries Optimizing for RDF Datasets

Jinguang Gu
College of Computer Science and Technology, Wuhan University of Science and Technology, Wuhan, China & Hubei Province Key Laboratory of Intelligent Information Processing and Real-Time Industrial System, Wuhan, China

Hao Dong
College of Computer Science and Technology, Wuhan University of Science and Technology, Wuhan, China & Hubei Province Key Laboratory of Intelligent Information Processing and Real-Time Industrial System, Wuhan, China

Zhao Liu
College of Computer Science and Technology, Wuhan University of Science and Technology, Wuhan, China & Hubei Province Key Laboratory of Intelligent Information Processing and Real-Time Industrial System, Wuhan, China

Fangfang Xu
College of Computer Science and Technology, Wuhan University of Science and Technology, Wuhan, China & Hubei Province Key Laboratory of Intelligent Information Processing and Real-Time Industrial System, Wuhan, China

ABSTRACT

In recent years, the scale of RDF datasets is increasing rapidly, the query research on RDF datasets in the transitional centralized environment is unable to meet the increasing demand of data query field, especially the top-k query. Based on the Spark distributed computing system and the HBase distributed storage system, a novel method is proposed for top-k query. A top–k query plan STA (Spark Threshold Algorithm) is proposed to reduce the connection operation of RDF data. Furthermore, a better algorithm SSJA (Spark Simple Join Algorithm) is presented to reduce the sorting related operations for the intermediate data. A cache mechanism is also proposed to speed up the SSJA algorithm. The experimental results show that the SSJA algorithm performs better than the STA algorithm in term of the cost and applicability, and it can significantly improve the SSJA's performance by introducing the cache mechanism.

DOI: 10.4018/978-1-7998-5339-8.ch092

INTRODUCTION

RDF (Resource Description Framework) (Decker, 2000) is a data model for data interchange on the web, plenty of research fields begin to use RDF to represent data for knowledge sharing. SPARQL (Simple Protocol and RDF Query Language) (Guha, 2003) is a standard RDF query language. In many cases, users are more interested in the most valuable results in the huge datasets, which is named as top-k query (Ilyas, 2008). Top-k join query is a kind of top-k query which involves multiple tables or several datasets and the results are sorted by the aggregation score. In the actual query, ORDER BY and LIMIT phrases can easily complete the extraction of top-k results. However, they are often used as a result modifier and completed at the last stage of the query in the SPARQL algebra expression. This kind of sorting mechanism is always insufficient.

In recent years, there are increasing research interests in the query optimization (Ci, 2014). For example, ite can be used in service selection (Ngoko, 2015) and service recommendation (Zhang, 2016). Since the query process involves many complex connections and sort operations, an accurate top-k join query is a time-consuming work. With the rapid development of semantic web technology, the amount of RDF datasets continues growing, thus brings a huge challenge to query performance. Classic methods of top-k join query on RDF datasets are designed on single machine, which have bottlenecks in memory space and computing performance that may affect the further development and application of semantic web query technology.

Cloud computing has become one of the most popular research fields due to its high performance and easy extension of mass data storage and computing power (Armbrust, 2010). SPARK, developed by AMP lab of Berkeley, is chosen for this research as an ideal new generation of distributed processing framework for big data process (Zaharia, 2010).

SPARQL top-k query optimization is a relatively comprehensive research area. The existing methods are mainly focus on the optimization of top-k join query algorithms (Hwang, 2007; Liu, 2006; Martinenghi, 2012) and relational algebra (Bozzon, 2012; Pérez, 2006; Schmidt, 2010;). Within that, optimization of top-k join algorithm is crucial. The typical example is HRJN (Ilyas, 2004) algorithm, which can only support accessing tuples sequentially thus it needs additional hash table to store the input tuples and causing the calculation of threshold is quite complex. RSEQ (Rank Sequence Operator) algorithm (Magliacane, 2012) optimized of the HRJN algorithm which uses the single ordered set to support random access to minimize sequential access thus enhance the efficiency. Wagner introduced the PBRJ (Wagner, 2012) algorithm which includes boundary pattern B and tuple access strategy P. P is used to select which set should be chose to read data and B is used to calculate the threshold. However, all above algorithms can only be used in a single machine hence cannot be directly applied to large-scale RDF dataset query in the distributed environment. How to optimize the traditional SPARQL top-k query algorithm and make it adapt to the distributed computing framework of cloud computing has become the key problem to be solved in this paper.

Recently, researchers have proposed some methods to calculate top-k queries in distributed environment for different types of large scale datasets. TPUT (Three-Phase Uniform-Threshold algorithm) (Cao, 2004) is a method which used in the distributed environment to calculate top-k queries. TPUT reduces network bandwidth consumption by pruning away ineligible objects and terminates in three round-trips regardless of data input. Paper (Lu, 2013) introduced a rank calculation method of uncertain data for parameterized ranking functions (PRF) and by analyzing that it puts forward a method to compute the