Simulation Modelling Practice and Theory 64 (2016) 30–42

Contents lists available at ScienceDirect

Simulation Modelling Practice and Theory

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

Selectivity estimation of extended XML query tree patterns based on prime number labeling and synopsis modeling

Salahadin Mohammed[a,*], Ahmad F. Barradah[b], El-Sayed M. El-Alfy[a]

[a] College of Computer Sciences and Engineering, King Fahd University of Petroleum & Minerals, Dhahran 31261, Saudi Arabia
[b] Exploration Network Operations Department, Saudi ARAMCO, Dhahran 31311, Saudi Arabia

ARTICLE INFO

Article history:
Available online 19 February 2016

Keywords:
Selectivity estimation
XML query
Query optimization
XML synopsis
Prime number labeling
Extended query tree patterns

ABSTRACT

With the new era of big data and the proliferation of XML documents for representing and exchanging data over the web, selectivity estimation of XML query patterns has become a crucial component of database optimizers. It helps the optimizer choose the best possible plan for query evaluation. Existing selectivity estimators for XML queries can only support basic Query Tree Patterns (QTPs) with logical AND operator. In this paper, we propose a novel approach, called XQuest, for selectivity estimation that supports extended QTPs that may contain logical operators or wildcards. This approach is based on a modified implementation of prime number labeling to construct a structural summary model of the XML data. Subsequently, a simulator of an XML query evaluator runs on the resulting model from the previous stage and aggregates the estimate for each target QTP. We conducted several experiments to study the performance of the proposed approach on three XML benchmark datasets; in terms of synopsis generation time, storage requirements, and estimation accuracy. The results show that the proposed approach can have more accurate estimates with low memory and time requirements. For example, when compared to a Sampling algorithm with the same allocated memory budget, the error rate of the proposed approach never reached 5% whereas it reached 98.5% for the Sampling algorithm.

1. Introduction

In today's fast-paced cyber world, the eXtensible Markup Language (XML) is gaining popularity as a ubiquitous standard format for data representation and exchange among various applications and systems [1–3]. With the rapid growth of XML documents, query optimization is very essential for efficient processing of queries in XML Database Management Systems (XDBMS) [4,5]. XML queries are declarative expressions typically written in XPath or XQuery languages recommended by W3C Consortium [6,7]. They contain various document nodes whose relationship is represented, in general, by a tree known as Query Tree Pattern (QTP) [8,9].

There are different types of QTPs [10]. For example, the queries shown in Fig. 1(a), (e), and (f) are linear QTPs, but the queries shown in Fig. 1(b)–(d) are branching or twig QTPs. In a linear QTP, each node has at most one child node whereas in a twig QTPs at least one node has two or more children. According to the XPath notations, parent-child (PC) relationship is denoted by a forward slash (/) between two nodes whereas ancestor-descendent (AD) relationship is denoted by a double

* Corresponding author. Tel.:+96638601721.
E-mail addresses: adam@kfupm.edu.sa (S. Mohammed), barradaf@aramco.com (A.F. Barradah), alfy@kfupm.edu.sa (E.M. El-Alfy).

http://dx.doi.org/10.1016/j.simpat.2016.01.008
S1569-190X(16)00015-0/

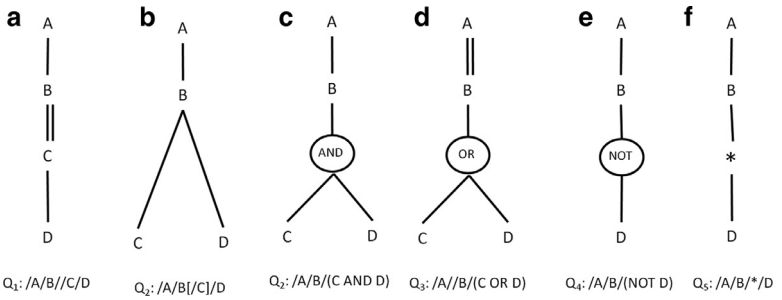

Fig. 1. Examples of various types of query-tree patterns.

forward slashes(//) between the two nodes. Fig. 1(a) and (b) are basic QTP whereas the rest are extended QTPs. An extended QTP is a more general form of QTP which may contain one or more logical operators or wildcards. The queries in Fig. 1(c)–(f) are denoted as AND-QTP, OR-QTP, NOT-QTP, and $*$-QTP, respectively. The AND-QTP can be easily converted to an equivalent basic QTP, e.g. Fig. 1(b) and (c) are equivalent. But the other extended queries have multiple basic QTPs that satisfy the conditions of the given query. Twig queries are further classified as existential when branches are only treated as existential structural constraints, but the number of the branches is ignored; otherwise, they are called regular twig queries [11].

The role of a query evaluator is to find and retrieve all matches of a QTP in a given XML document, which can be conducted in a variety of ways known as query execution plans. Choosing the best possible plan for a query is the job of a database query optimizer, which often relies on a selectivity estimator to determine the rough count of intermediate results generated by each plan [12]. Desirable characters of a selectivity estimator include capability to process different types of queries, capability to handle larger datasets with different structural characteristics, and efficient in terms of accuracy, CPU cost, and memory space consumption. This problem has attracted the attention of many researchers for more than a decade and various approaches have been proposed in the literature, e.g. [12–17]. Since it is not efficient to directly use the original XML document for estimation, a category of existing approaches depends on constructing a structural synopsis model to summarize the XML document in a compact form. Yet, they only focus on simple query expressions without logical operators or wildcards. Though there are a number of query evaluators that support extended QTPs with logical operators [18–30], there is a lack of selectivity estimators, as far as we know, that support them. A naive approach for converting extended QTPs into all possible combinations of basic QTPs will be inefficient.

The main contribution of this paper is to present a novel and efficient selectivity estimator, called XQuest, that supports extended XML queries which can include AND, OR, and NOT logical operators, or $*$ wildcard. Moreover, the proposed approach is empirically evaluated using three benchmark datasets. The first stage in the proposed approach constructs a synopsis model summarizing the original XML document. Then, for each given QTP, an estimate of the number of matches is computed from the generated synopsis. The first stage of our approach relies on our earlier work [31].

The rest of this paper is organized as follows. Section 2 reviews related work. Section 3 explains the proposed XQuest approach. Section 4 describes the experimental settings and empirical evaluation of the proposed approach. Section 5 concludes the paper.

2. Related work

Various methods have been proposed in the literature for selectivity estimation of XPath queries [11,14,17,32–40]. However, most methods focused on basic QTPs including linear-path and simple twig queries. None of the existing selectivity estimators supports extended QTPs.

Among the methods that handles linear-path queries are the two techniques proposed in [14]. The first technique captures the frequency of each node in the XML document on a summary tree called path-tree. If the path-tree does not fit the available memory, it is pruned by deleting nodes with the lowest frequencies. The second technique generates a statistical structure called Markov Table (MT), which is implemented as a hash table and contains the count of any distinct path of a length up to m, for various values of m. MT is further summarized by deleting low-frequency paths. The selectivity of any path of length less than or equal to m can be directly computed from the table. The selectivity of longer paths is generated by combining the paths of limited lengthes in the Markov table. The authors of [41] proposed a new data structure called SF-Tree that stores the counts of all single path expressions in an XML document. In SF-Tree, the path expressions are divided into groups according to their frequency. Thus, the selectivity of a linear path expressions can be computed by finding to which group it belongs. To increase the accuracy and the time and space efficiency of the SF-Tree, the groups are stored in signature files which are organized as tree.

In [42], a graph-synopsis model is proposed called XSketch which exploits localized graph stability to capture, in a limited space budget, important statistical correlations between data paths in an XML dataset. To compensate for the lack of detailed distribution information, the estimation relies on the assumption of uniformity and independence. It has been shown that