PSMQ: Path Based Storage and Metadata Guided Twig Query Evaluation

M. Archana, M. Lakshmi Narayana, and P. Sreenivasa Kumar

IIT Madras, Chennai,
India - 600036
archana.maram@gmail.com,lokesh512@gmail.com,psk@cs.iitm.ernet.in

Abstract. Efficient evaluation of queries on XML data is a major research issue. Structural join based techniques are well known for XPath evaluation. For the long path expressions, join techniques are not efficient as they increase the number of joins and disk I/O cost. Path based techniques try to reduce the number of joins. In this paper, we propose a metadata guided query evaluation technique which uses path based storage. We use interval encoding for the nodes. In addition, we use Strong DataGuide to assign integer path labels to distinct root-to-node label paths in the data tree. An *element list* is maintained for each distinct path consisting of nodes that can be reached by that path. The `Element-Map` gives the one-to-many mapping between element names (or tag names) to element lists with nodes having that tag-name. The `Path-Map` gives the root-to-leaf path for a given path label. Using these structures, we can combine top-down path matching and bottom-up path selections to efficiently evaluate linear path expressions. For twig queries, we perform structural joins at branch points. Through experimental evaluation on standard datasets, we show that our approach outperforms the existing path-index based approaches which in turn outperform structural join methods.

Keywords: DataGuide, XPath, structural summary, structural join.

1 Introduction

Many query languages such as XPath [4] and XQuery [9] have been proposed to query XML data. In this paper, we consider only the tree structured XML data i.e. data which does not include IDs and IDREFs and XPath queries are considered. We consider the queries that can be represented in the form of trees, called the *twig* queries.

Efficient processing of XPath expressions is one of the major recent research issues. Many techniques have been proposed to process path expressions efficiently. Some of the well known query evaluation techniques include join based algorithms [8,14], structural summary techniques [12] and path-ID based algorithms [17,5,10].

The structural join approaches [8,14] split the query into a set of binary structural join operations. The intermediate results of these joins are merged to get

R. Cooper and J. Kennedy (Eds.): BNCOD 2007, LNCS 4587, pp. 114–124, 2007.

the final result of the query. All the above mentioned structural join algorithms use interval encoding as the node identification scheme. To evaluate all the XPath axes, Staircase Join Algorithm [15] which uses pre-post node encoding scheme is proposed. In join approaches computation cost due to the style of one-join-per-location-step becomes unacceptably huge, especially when the path expression is long.

Unlike join approaches, structural summary based approaches restrict the search to only relevant portion of XML data. Examples of such approaches include DataGuide [12] and Index Fabric [6]. These techniques can efficiently process the absolute paths.

To reduce the number of joins and disk accesses, path based techniques such as BLAS [17] and MQEB [5] are proposed. These algorithms assign *pid*(path id also called P-Label) to each element and also to all the possible paths in the document. The *pid* of a node encodes the root-to-node path(also called as source path of the node) for that node. When compared to structural join techniques, these approaches reduce the number of joins and disk I/O cost. BLAS needs joins at each ancestor-descendant step and branching points but MQEB needs joins only at branching points. For non-branching path expressions, MQEB does not need any joins. But both of these approaches fail to assign *pid* values to elements in case of XML documents that are deep and have large number of distinct elements. As the P-Label calculation depends on the number of distinct tags information they need to reassign the P-Labels if a new tag-name is added.

The other path based technique XRel [10] is a relational system. XRel converts the given XML document into four tables: `Element, Attribute, Text` and `Path`. The schema of the tables is as follows:

```
Element(docID, pathID, start, end, index, reindex)
Attribute(docID, pathID, start, end, value)
Text(docID, pathID, start, end, value)
Path(pathID, pathexp)
```

This system also uses interval encoding to assign *start* and *end* values to all the elements, attributes and text in the document. Given XPath queries are converted to SQL and executed on relational tables.

BLAS and MQEB perform better than join approaches but have difficulties in assigning P-Labels. The XRel system has the additional overhead of query conversion. To overcome the problems with existing path based approaches, we propose a system called as PSMQ(**P**ath Based **S**torage and **M**etadata Guided **Q**uery Evaluation). We use interval encoding for node identification and strong DataGuide to assign P-Labels (or path labels) efficiently. We keep path summaries in the form of metadata to reduce the search effort while evaluating queries. Unlike BLAS and MQEB, we do not need to reassign the P-Labels, if a new tag-name gets added to the existing document.

Contributions of this paper are,

- Proposing a storage scheme which uses interval encoding as the node identification scheme and strong DataGuide approach to assign path labels.