ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)
Vol. 2 Issue Special 1 Jan-March 2014
International Journal of Advanced Research in Computer Science & Technology (IJARCST 2014)

An Efficient Processing of Twig Pattern Queries with Dynamic XML Dissemination

[I]Jetlin.C.P, [II]Mercy.W, [III]Dr. P.S.K.Patra
[I,II,III]PG Student, Dept. of CSE, Agni College of Technology, Anna University, Chennai, TamilNadu, India

Abstract
XML dissemination-where the streams of XML documents are arriving at a fast rate and the server is responsible for managing these documents and disseminate them to a pool of clients. The main aim of this project is to support an energy and latency efficient XML dissemination scheme for mobile computing. Hence we define a novel unit structure called G-Node supporting Twig pattern Queries based on Lineage Encoding. The Lineage Encoding scheme represents the parent-child relationships among XML elements as a sequence of bit-strings, called Lineage Code (V, H). In evaluating a given twig pattern query with predicates, suitable selection string function and operators are used in the lineage encoding over the stream. An XML automatic creation tool is proposed to customize the XML tree representation and thereby to support dynamic G-Node streaming. Thus, our scheme outperforms well than existing twig pattern algorithms and can support twig pattern query processing, while providing both energy and latency efficiencies

Keywords
XML Dissemination, Twig pattern queries, G-node, lineage encoding, dynamic XML dissemination

I. Introduction

XML is in fact emerged as a standard and most popular format for exchanging and representing data on the Internet. XML allows the encoding of Structural or hierarchical information and enables more sophisticated filtering mechanisms. XML describes the semantics of the document in prior to the data itself. This made that XML is the best way of data exchange in today's internet. A sample XML document is given in fig 1.1. A tree structure is a way to represent the hierarchical nature of XML data. XML provides great flexibility and wider acceptance throughout the world.

XML data is becoming an essential requirement for many applications in mobile wireless networks. Several indexing methods have been proposed to reduce the tuning time in processing the XML queries over the wireless XML stream. Tuning time is the sum of period of times which a mobile client stays in active mode in order to retrieve the

```
<Nation>
    <Country >
        <Name>Belgium</Name>
        <Province id=”id1” name=”prams”> </Province>
        <Location>north</location>
         <City id=”c1” population=”100000”> </City>
    </Country>
    <Country >
        <Name>Bulgaria</name>
        <province id=”id2” name=”Hani”> </Province>
        <Location>south</Location>
         <City id=”c2” population=”150124” > </City>
    </Country>
    <Country >
        <Name>Finland</name>
        <Province id=”id3” name=”eland”> </Province>
        <Location>west</location>
         <City id=”c3” population=”102546” > </City>
    </Country>
</Nation>
```

Fig. 1.1: Sample XML Document

required data over the wireless stream. Therefore, it is frequently used to estimate the energy consumption of a mobile client.

Figure 1.2 shows the architecture of the wireless XML broadcasting scheme supporting Twig pattern queries. With the tremendous development of internet and networking features, it made user possible to access large volume of data in a convenient way. At the server side, the XML data to be disseminated is parsed based on SAX interface and stream of g-node is generated. Wireless streaming have been proposed to reduce the structural overheads of the original XML document and attach indices containing time information to the XML data stream. Information Dissemination applications are gaining popularity in distributing data to the end users. In the client side when the query is issued by the client, query tree is generated, and tunes into the broadcast channel i.e., streaming and downloads selectively according to the client's interest. XML Automatic creation tool is used for the customized XML creation which enables the server to broadcast customized data without relying on the third party for XML files. Thus our implementation supports dynamic customized XML dissemination in wireless environment.

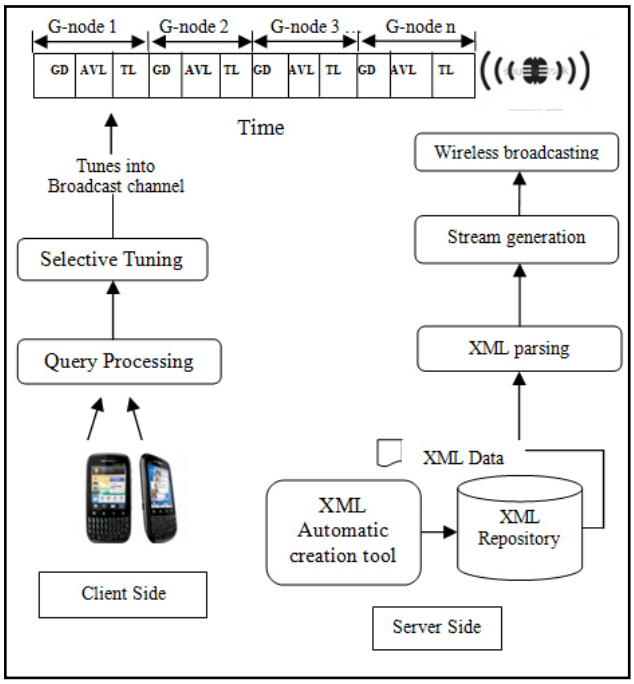

Fig 1.2: Architecture of XML Broadcasting Scheme

ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)

The remainder of our work organized as follows: backgrounds and problem statement are described in section II. In section III related works on twig pattern query algorithms are made. In section IV Wireless XML dissemination scheme and brief analysis of lineage encoding algorithm supporting twig pattern queries are made.

II. Background

Since the data objects in a variety of languages are typically trees, tree pattern matching (twig) is the central issue. Naturally queries in the XML query language specify the patterns of selected predicates on multiple elements which have a tree structured relationships. The complex query tree pattern is usually decomposed into set of basic parent-child and ancestor-descendant relationships. But finding all these basic structural relationships occurrences is a complex process in the XML query processing. There are various techniques provides wireless XML dissemination schemes but none of them supports twig pattern queries since they does not have parent child relationship. Normal index methods divides a query into several sub-queries, thereby join the results together to provide the final answer. Twig pattern search uses tree structures as the master unit of query to avoid expensive join operations.

A. XPath Expression

In this paper, we use XPath [11] as a query language. The results of an XPath query are selected by a location path. A location path consists of location steps. Processing each location step selects a set of nodes in the document tree that satisfy axis, node test and predicates. For example, a query that finds cities located in Belgium can be represented by the following XPath expression.
Q1://Country[@name=”Belgium”]/Province/City

B. Twig Pattern Query

Millions of people around the world have different favors or patterns in retrieving data. . Thus, the main challenge is to retrieve the information from the tremendous database of Internet based on user’s patterns. This is known as Twig Pattern Search. Index method which divides the query into several sub-queries, and then join the results together to provide the final answer. But twig pattern query uses tree structures as the extensive unit to avoid expensive join operations. The twig pattern query thus support any sort of complex queries. It involves two or more path expressions. Several researches have been made to efficiently process XML twig pattern queries. Generally the twig pattern query process as follows:

- Disintegrate the tree pattern into linear patterns which might be binary(parent-child or ancestor descendant) relationships between pairs of nodes or root- to-leaf paths
- Find all matches of each linear pattern use an index to facilitate the structural join process and do not require sorted input lists.
- Merge them to produce the result.

C. Wireless XML Streaming

Wireless streaming of the XML data supports energy-efficient processing of queries over the stream in mobile clients. In this the XML data are streamed in the wireless environment. With the recent development in the wireless technologies XML data’s are most often used in the information system. In wireless systems data broadcasting is widely used due to the bandwidth restrictions of wireless environment. The stream organization of the XML data which have different kinds of addresses for related data in the stream e.g., event-driven stream generation algorithms, search algorithms for simple XML path queries which leverage the access mechanisms incorporated in the stream.

C. Related Work

As XML queries are represented as twigs, the recent papers are proposed to efficiently process an XML twig pattern. S. Al-Khalifa et al [1] proposed Structural Joins for efficient XML query pattern matching. Though this algorithm supports relational query processor this algorithm does not focus on the pointer-based joins. And there are many issues yet to be explored. Zhang et al[2] proposed merge join algorithm, called Multi-Predicate merge join (MPMGJN) algorithm for finding all occurrences of the basic structured relationships i.e., containment query processing. A new holistic twig join algorithm is proposed by Nicolas Bruno et al [3] to optimal XML pattern matching. But they still need to merge various twig path patterns which result in high computational cost. Twig2Stack algorithm proposed by Songting et al. in [4] uses hierarchical stack encoding method. Twig2Stack leads many random accesses in memory, and makes to load the whole XML tree into memory in the worst case. Hence it does not handle the worst case memory issues. Twig List which makes the Twig Pattern Matching Fast is proposed by Lu Qin et al. in [5] where the space and time complexity of this algorithm are linear with respect to the number of occurrences of twig-patterns and the size of XML tree. Wei Wang et al [6] proposed XML Twig Queries with OR Predicates. The study shows that the previous work addressed only the simple path queries. And they are inefficient in twig pattern queries. But our lineage encoding algorithm is the only method which supports the twig pattern queries and predicate conditions efficiently

IV. XML Dissemination Scheme

Our work focuses on to improve the energy and latency efficient XML dissemination for the mobile computing and to support twig pattern queries. In order to support this

- A novel unit structure called G-node is used for streaming the XML document. And this allows the selective tuning of data during the query processing, enables query processing time to be minimized.
- A light weight encoding scheme called lineage encoding is used, which converts the XML data into bits and thereby supports parent child relationship among G-nodes.
- An XML automatic creation tool is used to customize the XML and enables the server to broadcast the customized data without relying on the third party such as native XML DBMS for XML files. And thus to support dynamic XML dissemination for live updates.

A. XML Data & Manipulation

In an XML document the elements, attributes, and the texts are represented by nodes and the parent-child relationships are represented by edges. The broadcast server retrieves XML data to be disseminated from the XML Repository. Then the XML document is parsed using Simple API for XML. Structure Indexing approach integrate multiple elements of the same path into one node. Thus the size of data stream can be reduced by eliminating redundant tag names and to efficiently support twig queries.

B. Lineage Encoding & Attribute Summarization

A novel algorithm called lineage encoding proposed by J.P. park et

al. [7] is used here to support queries which are having predicates and twig patterns. This scheme uses two kinds of lineage code called vertical code denoted by Lineage Code (V) and horizontal code denoted by Lineage code (H). The Lineage Encoding scheme represents the parent-child relationships among XML elements as a sequence of bit-strings, called Lineage Code (V, H). This lineage code is lightweight and effective encoding scheme, which supports the twig queries and evaluation of predicates over the stream. It is also an efficient bit string representation. And also this scheme uses G-node method for the wireless dissemination of XML data.

Definition 1: for Lineage code (V), let G-node C is a child of the G-node P. from the parent element EP if there exist at least one child element in EC then mark it as 1 otherwise 0.

Definition 2: for Lineage code (H), it is the ordered list of positive integers for which the number of child elements in EC mapped to the same parent element EP in the XML document.

We also use relevant operators and functions that exploit bit-wise operations on the lineage codes. Attribute summarization is performed to reduce the size of the wireless XML stream by eliminating the redundant attribute names.

C. G-node & XML Dissemination

After performing lineage encoding, bi wise operation we define a streaming unit of a wireless XML stream, called G-node. The G-node structure eliminates structural overheads of XML documents, and enables mobile clients to skip downloading of irrelevant data during query processing.

The group descriptor is a collection of indices for selective access of a wireless XML stream. Node name is the tag name of integrated elements, and Location path is an XPath expression from the root node to the element node in the document tree. Child Index (CI) is a set of addresses that point to the starting positions of child G-nodes in the wireless XML stream. Attribute Index (AI) contains the pairs of attribute name and address to the starting position of the values of the attribute that are stored contiguously in Attribute Value List.

The components of the group descriptor are used to process XML queries in the mobile client efficiently. Specifically, Node name and Location path are used to identify G-nodes. Indices relating to time information such as CI, AI, and TI are used to selectively download the next G-nodes, attribute values, and text. Finally, Lineage Code (V, H) is used to support twig pattern queries in the user's query. Attribute Value List (AVL store attribute values of the elements represented by the G-node, respectively .All the G-Node data's are broadcasted with the help of a Wi-Fi device which can be received by any android devices in its coverage.

D. Query Tree Formation & Selective Tuning

When a query is issued by the mobile client, query tree is generated and we describe how a mobile client can retrieve the data of its interests. Two types of algorithms are used here:

- Simple path query processing
- Twig pattern query processing

1. Simple Path Query Processing

Given a query, the mobile client constructs the query tree. Then, it starts to find the relevant G-nodes over the Wireless XML stream. Now, the mobile client downloads the group descriptor of the G-node which corresponds to the query node. If the current node is the leaf node, the mobile client downloads all the attribute values and text in AVL and TL. Otherwise, the mobile client selectively downloads only the attributes values and /or texts involved in the predicate using AI and TI.

2. Twig Pattern Query Processing

In this method the twig pattern query processing is done by using three phases namely Tree traversal phase, Sub path traversal phase and Main path traversal phase. In tree traversal phase the search is done in a depth first manner. In the sub path traversal phase the search is done in the post order depth first traversal where it is starting from the highest branching node from the query tree. At last in the main path traversal phase the traverse is done from the main path from the query tree.

E. XML Automatic Creation Tool

The XML Automation tool is used for customized XML creation enables the server to Broadcast the customized data's when needed without relying on the third party for XML files. It also allows user to create various XML tree representation according to the client's data format thus to support live updates immediately and efficiently.

V. Conclusion

Twig pattern queries which containing complex conditions are popular and critical in XML query processing. In this paper, an efficient XML dissemination scheme is used for supporting twig pattern queries. The mobile client can retrieve the required data satisfying the given twig pattern by performing bit-wise operations on the Lineage Codes in the relevant G-nodes. Thus, our scheme can support twig pattern query processing while providing both energy and latency efficiencies. The lineage encoding algorithm is the only method which supports the twig pattern queries and predicate conditions efficiently. Our proposed approach XML automation creation tool support to dynamic customized XML is a major advantage of the wireless streaming in mobile environment. Dynamic addition of G-Node ensures the credibility of the Broadcast system and handles live updates immediately is the efficiency of our approach.

References

[1] *S. Al-Khalifa, H.V. Jagadish, N. Koudas, J.M. Patel, D. Srivastava, and Y. Wu, "Structural Joins: A Primitive for Efficient XML Query Pattern Matching," Proc. Int'l Conf. Data Eng. (ICDE), pp. 141-152, Feb. 2002.*

[2] *C. Zhang, J.F. Naughton, D.J. DeWitt, Q. Luo, and G.M. Lohman,"On Supporting Containment Queries in Relational Database Management Systems," Proc. ACM SIGMOD Int'l Conf. Management of Data Conf., pp. 425-436, 2001.*

[3] *N. Bruno, D. Srivastava, and N. Koudas, "Holistic Twig Joins: Optimal XML Pattern Matching," Proc. ACM SIGMOD Int'l Conf. Management of Data Conf., pp. 310-321, 2002.*

[4] *Songting Chen, Hua Gang Li, Junichi Tatemura, WangPin Hsiung, Divyakant Agrawal, K. Selc¸uk Candan "Twig2Stack:Bottomup Processing of Generalized Tree Pattern Queries over XML Documents" proc. ACM VLDB 06, September 1215, 2006.*

[5] *Lu Qin, Jeffrey Xu Yu, and Bolin Ding, "TwigList: Make Twig Pattern Matching Fast"*

[6] *H. Jiang, H. Lu, and W. Wang, "Efficient Processing of XML Twig Queries with OR-Predicates," Proc. ACM SIGMOD Int'l Management of Data Conf., pp. 59-70, June 2004.*

ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)

[7] J.P. Park, C.-S. Park, and Y.D. Chung, "Lineage Encoding: An Efficient Wireless XML Streaming Supporting Twig Pattern Queries," IEEE transactions on knowledge and data engineering, vol. 25, no. 7, july 2013

[8] J.P. Park, C.-S. Park, and Y.D. Chung, "Energy and Latency Efficient Access of Wireless XML Stream," J. Database Management, vol. 21, no. 1, pp. 58-79, 2010.

[9] M. Altinel and M. Franklin, "Efficient Filtering of XML Documents for Selective Dissemination of Information," Proc. Int'l Conf. Very Large Data Bases (VLDB), pp. 53-64, 2000.

[10] I. Tatarinov, S. Viglas, K. Beyer, J. Shanmugasundaram, E. Shekita, and C. Zhang, "Storing and Querying Ordered XML Using a Relational Database System," Proc. ACM SIGMOD Conf., pp. 204-215, 2002.

[11] A. Berglund, S. Boag, D. Chamberlin, M.F. Fernandez, M. Kay, J.Robie, and J. Simeon, "XML Path Language (XPath) 2.0," Technical Report W3C, 2002.