International Journal of Advanced Information Science and Technology (IJAIST) ISSN: 2319:268
Vol.2, No.9, September 2013 DOI:10.15693/ijaist/2013.v2i9.12-14

Dynamic XML Dissemination Supporting Twig Pattern Queries

S. Vidya
Final Year M.E/Department of CSE
Sri Sairam Engineering College, Chennai

R. Uma
Professor/ Department of CSE
Sri Sairam Engineering College, Chennai

***Abstract* - The main aim of this paper is to improve energy and latency efficiency of XML dissemination scheme for the mobile computing based on Lineage Encoding using a novel unit structure called G-node for streaming XML data in the wireless setting. It exploits the advantages of the structure indexing and attribute summarization that integrate relevant XML elements into a group and provides a way for selective access of their attribute values in a dynamic way that broadcasting can be done dynamically supporting Twig Pattern Queries.**

***Index Terms* - Twig pattern matching, dynamic xml, dissemination.**

I. INTRODUCTION

With the rapid development of wireless network technologies, wireless mobile computing has become fashionable. Users communicate within the wireless mobile surroundings using their mobile devices like good phones and laptops whereas they are moving. So, we define a novel unit structure called G-node for streaming XML data in the wireless environment. It exploits the advantages of the structure indexing and attributes summarisation which will integrate relevant XML components into a group. It provides some way for selective access of their attribute values and text content.

We propose Lineage Encoding, which is lightweight encoding scheme to support analysis of predicates and twig pattern queries over the stream. We want to think about energy conservation of mobile clients when disseminating data in the wireless mobile environment, because they use mobile devices with limited battery-power.The overall query processing time must also be minimized to provide fast response to the users. The goals of conventional query processing on streamed XML data are to reduce computation prices and filtering time.

In wireless XML broadcasting, the broadcast server retrieves XML information to be disseminated from the XML repository. Then, it parses and generates a wireless XML stream. The XML stream is incessantly disseminated via a broadcast channel. In the client-side, if a query is issued by the mobile client, the mobile client tunes in to the broadcast channel and selectively downloads the XML stream for query processing. The XML Broadcasting is completed expeditiously in such the simplest way the server will support dynamic dissemination of a G node with none interruption in Broadcasting.

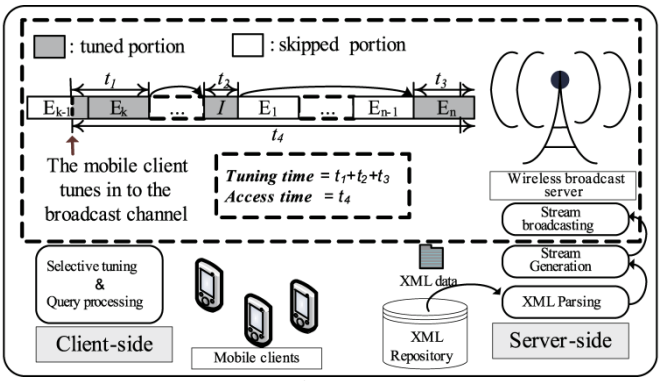

Fig. 1. Architecture of a wireless XML broadcasting system.

II. XML DATA MODEL AND XPATH QUERY PROCESSING

Client has to pass a request to server to get their appropriated Record. Conventional XML query processing methods are used in the native XML DBMS. On the other hand, the target of our work is the wireless mobile environment.

We evaluate the performance of our scheme by conducting extensive Experiments using both the real data set and the synthetic data set and its performance not much satisfied.

Systems were proposed so that the stream can support only static XML rendered from repositories. However their designs are not efficient for dynamic broadcasting of XML data over the stream. To prevent Structural overheads of the XML data methodologies like Structured Indexing have been already done so that the XML is ready for broadcasting with its attribute summarized values.[5][6][7][8].

III. DYNAMIC XML DISSEMINATION

Our work is different from the existing ways, we define a novel unit structure called G-node for streaming XML data in the wireless environment.[6][7][8][9].

First, the server can support a massive number of mobile clients without additional costs (i.e., scalability).

Second, the broadcast channel is shared by many clients (i.e., - the effective utilization of bandwidth);

Third, the mobile clients can receive data without sending request messages that consumes much energy.

Dynamic XML dissemination can be done so that the mobile clients can have live updations of the data.No need to rely on third party repositories.

Finally our proposed framework and three components are highly accurate under various conditions.

```
<mondial>
    <country id="f0_162" name="Belgium" capital="f0_1477" population="10170241">
        <name>Belgium</name>
        <province id="f0_17457" name="Antwerp">1610695</province>
        <province id="f0_17462" name="Hainaut">
            <located at>West of Belgium</located at>
            <city id="f0_2335" country="f0_162" province="f0_17462">Charleroi</city>
                <population> 255634 </population>
            <city id="f0_2345" country="f0_162" province="f0_17462">Mons</city>
                <population> 188639 </population>
            <city id="f0_2345" country="f0_162" province="f0_17462">Liege</city>
        </province>
    </country>
    <country id="f0_174" name="Bulgaria" capital="f0_1487" population="8612757">BU</country>
    <country id="f0_208" name="Finland" capital="f0_1507" population="5105230">
        <province id="f0_33615" name="Aland">
            <city id="f0_35399" country="f0_208" province="f0_33615">Mariehamn</city>
                <population> 315630 </population>
        </province>
        <province id="f0_33620" name="Haeme">662000</province>
    </country>
    <country id="f0_184" name="Czech Republic" capital="f0_1493" population="10321120">
        <province id="f0_17473" name="Jihomoravsky">
            <located at>East Cost of Czech</located at>
            <city id="f0_2394" country="f0_184" province="f0_17473">Zlin</city>
        </province>
        <province id="f0_17475" name="Severocesky"></province>
    </country>
</mondial>
```

Fig.2. Sample XML document.

IV. XML DATA AND MANIPULATION

An XML document has root and it is ordered as well as it is labeled. Nodes represent the Elements, attributes, and texts and the edges represent the parent-child relationships. It shows a straightforward XML document which will be used as a running example within the paper.

A server retrieves an XML document to be disseminated from the XML repository and it generates wireless XML steam by using SAX (Simple API for XML), that is an event-driven API. SAX invokes content handlers throughout the parsing of an XML document. [1][2][3][4].

Structured 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 thereby enabling Twig Pattern Query Processing.

Q2: //country[name/text() ="Belgium"]/province/city

Q3: //country/province[located at /text()="Middle"]/city

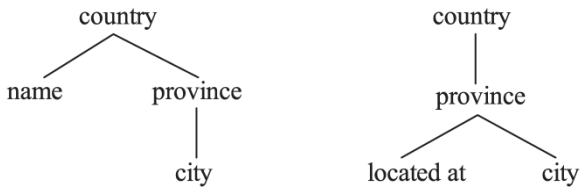

Fig.3. Example twig pattern queries and their tree representations

V. LINEAGE ENCODING & ATTRIBUTE SUMMARIZATION

The Novel algorithm is divided into two main phases:

There are two kinds of lineage codes namely, Lineage Code (V) which is vertical code and the Lineage Code (H) called the horizontal code. These codes are used to represent the parent-child relationships among XML elements in the two G-nodes. Relevant operators and functions that exploit bit-wise operations on the lineage codes are defined.

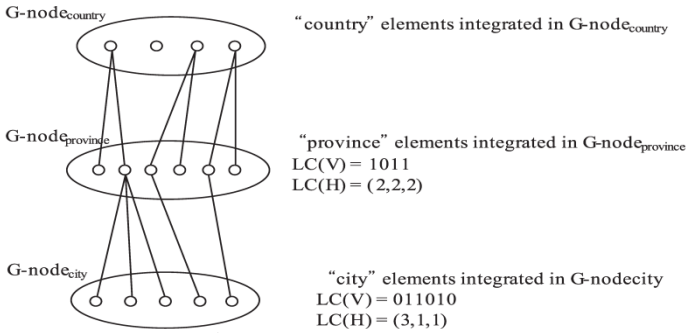

Fig.3. Lineage codes of G-nodecountry,G-nodeprovince and G-nodecity.

The Attribute Value List (AVL) generated in Attribute Summarization with lineage encoded data is the key to process the Twig Pattern Queries in Selective tuning approach in the mobile end.[9][10].

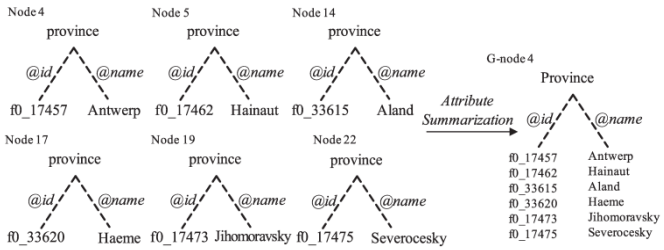

Fig.5. Attribute Summarization

VI. G-NODE & XML DISSEMINATION

G-node is a streaming unit of a wireless XML stream. This structure eliminates structural overheads of XML documents, and allows mobile clients to skip downloading of unnecessary data during query processing.

The group descriptor is a collection of indices and used for selective access of a wireless XML stream. Node name is that the tag name of integrated components. Location path is an XPath expression of integrated components from the root node to the element node within the document tree. Child Index (CI) is a set of addresses of the starting positions of the 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. [2][3][4][5]

The parts of the group descriptor are used to process XML queries in the mobile client efficiently. G-nodes are identified by the Node name and the Location path. To selectively download the next G-nodes, attribute values, and text, indices concerning time information such as CI, AI, and TI are used. Lineage Code (V, H) is employed to handle axis and predicate conditions within the user's query. All the G-Node data's are Broadcasted with the help of a Wifi device which can be received by any android devices in its coverage.

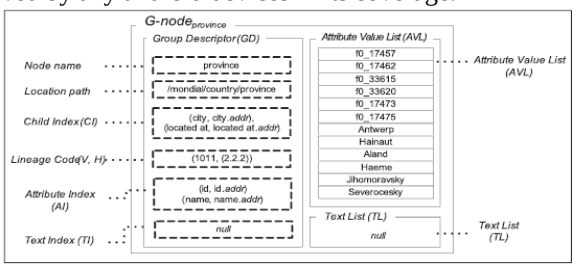

Fig.6. Structure of an example G-node.

VII. Query Tree Formation & Selective Tuning

In this section, we describe how a mobile client can retrieve the data of its interests. Here the assumption is that there is no descendant axis in the user query.

Simple Path Query Processing: This shows the easy path query processing over the wireless XML stream. The mobile client constructs a query tree for the given query. Then, it starts to seek out relevant G-nodes over the wireless XML stream. Group descriptor of the G-node is downloaded by the mobile client which corresponds to the query node. If this node is that the leaf node, the mobile client downloads.

Twig Pattern Query Processing: In the Tree traversal phase, the mobile client first constructs a query tree. Then the query tree is traversed in a depth-first manner. Group descriptors of the relevant G-nodes are selectively downloaded. [1][2][3][4]. Our Selective tuning approach is dynamic and it eases the client to minimize the tuning time and thereby reducing access time also. It dynamically chooses between the Twig Pattern Query and Normal Query and process to render the data.

Tuning is optimized with the help of the XPath Query pattern which holds the predicates.

VIII. ENHANCEMENT

Our XML Automation tool is used for customized XML creation enables the server to Broadcast the customized data's as and when needed without relying on the third party for XMl files. Our Implementation support to dynamic customized XML is a major advantage of the wireless streaming in mobile environment. The results of an XPath query are selected by a location path. A GNode the novel attribute of our system can be added dynamically to the broadcast channel without interrupting the streaming of XML data. This feature enables to dynamically add events in the existing channel.

Dynamic addition of GNode ensures the credibility of the Broadcast system efficiently proposed by our approach. AVL tree and Structured Indexing process will be handled that will probably affect the XML document in temporary buffer.

Dynamic modification of Attribute value enables to change any data on the broadcast stream whenever needed and is achieved by the Attribute summarization mechanisms and the Structured Indexing of XML data handled in our system.

IX. CONCLUSIONS AND FUTURE WORK

Twig pattern queries containing complicated conditions are popular and significant in wireless XML streaming method which supports twig pattern queries is proposed. w Lineage Encoding and twig pattern matching is proposed. Relevant operators and functions to efficiently process twig pattern matching are defined. The mobile client will retrieve the desired information by satisfying the given twig pattern and by performing the bit-wise operations on the Lineage Codes in the relevant G-nodes. Thus, our scheme supports the twig pattern query processing whereas providing each energy and latency efficiencies. We demonstrated our scheme is effective and efficient. We also showed that the typical XML query processing strategies are inefficient within the wireless mobile environment owing to their immense indices. In future, we plan to analyze the following issues: First, depth-first traversal of components increases the access time for specific queries. Second, as the communication isn't stable in the wireless broadcasting environment, the indexing mechanism should consider network failures like tail drops and packet losses.

References

[1] S. Acharya, R. Alonso, M. Franklin, and S. Zdonik, "Broadcast Disks: Data Management for Asymmetric Communication Environments," Proc. ACM SIGMOD Int'l Conf. Management of Data Conf., pp. 199-210, Mar. 1995.

[2] 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.

[3] 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.

[4] S. Amer-Yahia, S. Cho, L.V.S. Lakshmanan, and D. Srivastava, "Minimization of Tree Pattern Queries," Proc. ACM SIGMOD Int'l Conf. Management of Data Conf., pp. 497-508, 2001.

[5] 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.

[6] 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.

[7] S. Chen, H. Li, J. Tatemura, W. Hsiung, D. Agrawal, and K.S.Candan, "Scalable Filtering of Multiple Generalizaed-Tree-Pattern Queries over XML Streams," IEEE Trans. Knowledge and Data Eng.,vol. 20, no. 12, pp. 1627-1640, Dec. 2008.

[8] C. Chung, J. Min, and K. Shim, "APEX: An Adaptive Path Index for XML Data," Proc. ACM SIGMOD Int'l Conf. Management of Data Conf., June 2002.

[9] Y.D. Chung, S. Yoo, and M.H. Kim, "Energy- and Latency-Efficient Processing of Full- Text Searches on a Wireless Broadcast Stream," IEEE Trans. Knowledge and Data Eng., vol. 22, no. 2,pp. 207-218, Feb. 2010.

[10] B. Cooper, N. Sample, M.J. Franklin, G.R. Hjaltason, and M.Shadmon, "A Fast Index for Semistructured Data," Proc. Int'l Conf.Very Large Data Bases (VLDB), Jan. 2001.

Authors Profile

S. Vidya received the **B.E.** degree in Computer Science and Engineering from the Arulmigu Kalasalingam College of Engineering, Krishnankoil, Anna University, Chennai, India, in 2010.Currently doing **M.E.** in Computer Science and Engineering in Sri Sairam Engineering College, Anna University, Chennai, India. Her research interest includes Data mining and Data warehousing, Data structures, Computer Networks, Crptography and Network security.

R. Uma received the **B.E.** degree in Electrical and Electronics Engineering from the Madras University, Chennai, India, in 1996. She has received the **M.E.** degree in Computer Science and Engineering from the Anna University, Chennai, 2002. Currently doing **Ph.d.** in Anna University,Chennai, India. Her research interest includes Data Mining.