



A Novel Encoding Technique of Broadcasting XML data using Twig patterns in wireless Environment

Mahadevi S. Namose*
ME Computer & BSCOER
Pune India

Rajesh Kulkarni
Assistant Professor & BSCOER
Pune India

Abstract— Objective of this paper is improving energy and latency efficiency of XML dissemination scheme for the mobile computing, which is based on Lineage Encoding, G-node and scheduling algorithm for streaming XML data in the wireless environment. It uses the advantages of the structure indexing and attribute summarization methods. The Lineage Encoding scheme represents the parent-child relationships among XML elements as a sequence of bit-strings, called Lineage Code(V, H), and provides basic operators and functions for effective twig pattern query processing at mobile clients. The organization of data on wireless channels, which aims to reduce the access time of mobile clients, is a key problem in data broadcasts. Many scheduling algorithms have been designed to organize data on air. We analyze the data broadcast scheduling problem of on-demand XML data broadcasts and define the efficiency of a data item. Based on the definition, a Least Efficient Last (LEL) scheduling algorithm is also devised to effectively organize XML data on wireless channels.

Keywords— Twig pattern matching, LEL, Attribute Summarization, Lineage Encoding, XML Dissemination.

I. INTRODUCTION

Extensible Markup Language (XML) is the World Wide Web Consortium recommendation for the exchange of structured information over the Internet. It is an open, non-proprietary, human and machine-readable metalanguage. XML was designed to enable businesses to create and exploit opportunities presented by the Internet as a medium for financial and information transactions. XML not only describes the data itself, but also the semantics of the document[10]. This enables users to organize information flexibly. That is the reason why it is used so widely in today's Internet. as many communities adopted it for various purposes, e.g., mathematics with MathML [5], chemistry with CML [6], geography with GML [7], and e-learning with SCORM [8]. Wireless broadcasting is an effective information dissemination approach in the wireless mobile environment because of the following benefits: 1) the server can support a massive number of mobile clients without additional costs, 2) the broadcast channel is shared by many clients (i.e., the effective utilization of bandwidth), and 3) the mobile clients can receive data without sending request messages that consumes much energy. There are **two typical** broadcast modes for data broadcast 1) Broadcasting Mode-data is periodically broadcast on the downlink channel. Clients only “listen” to that channel and down-load data they are interested in. 2) On-Demand Mode-the clients send their requests to the server through uplink channel and the server considers all pending requests to decide the contents of next broad-cast cycle[3]. We need to consider 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. To measure the energy-efficiency and latency-efficiency in wireless broadcasting, the tuning time and access time are used respectively. The tuning time is the sum of the elapsed times spent by a mobile client to download the required data. When a mobile client downloads the data it consumes more energy than when it waits for data. Thus, the tuning time is used as a performance measure for energy-efficiency. The access time is the time elapsed from when a mobile client tunes in to the broadcast channel to when the desired data is completely retrieved from the stream [1].

There is a one propose system exist Lineage Encoding, which is lightweight encoding scheme to support analysis of predicates and twig pattern queries over the stream[1].There is a efficient scheduling algorithm exist LEL for wireless broadcasting data [3], In this paper we propose a third system which combines above two existing system. So the overall query processing time must also be minimized to provide fast response to the users.

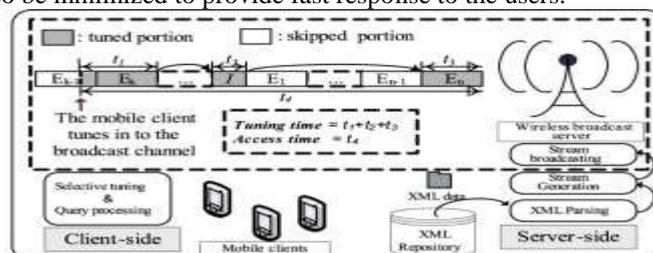


Fig 1 Architecture of a wireless XML broadcasting system

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. In Fig. 1 assuming that I is an index segment over the stream and E_n is the target data, the tuning time is the sum of t_1 , t_2 , and t_3 , whereas the access time is t_4 [1]. 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.

II. BACKGROUND

A. Definitions

- *XML Document*

XML is known to be a simple and very flexible text format. It is essentially employed to store and transfer text-type data. The content of an XML document is encapsulated within elements that are defined by tags. These elements can be seen as a hierarchy organized in a treelike structure

- *XPATH & XQUERY*

XPath 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 described in the step. XPath is a declarative language, and XQuery is an iterative language which uses XPath as a building block, providing path expressions as a searching condition.

- *Tree Pattern :*

Mostly there are two types of tree pattern[2]. 1) Tree Patterns in Algebraic Frameworks 2) Tree Patterns Used in Optimization Processes

- *Tree Patterns in Algebraic Frameworks :*

- TAX Tree Pattern
- Generalized Tree Pattern (GTP)
- Annotated Tree Pattern
- Ordered Annotated Pattern Tree

- *Tree Patterns Used in Optimization Processes*

- Global Query Pattern Tree (G-QPT)
- Twig Pattern
- C)Logical Operator Nodes
- D)Node Degree and Output Node Specification
- E)Extended Formula
- F)Extended Tree Pattern

- *Twig Pattern Query*

A twig pattern query consists of two or more path expressions, thus, it involves element selections satisfying complex patterns in tree-structured XML data. The twig pattern query is a core operation in XML query processing and popularly used as it can represent complex search conditions.

III. DYNAMIC XML DISSEMINATION

For providing energy-efficient query processing over XML data in wireless and mobile environments, several approaches are used. In mobile environments, the client request downloaded data requires more computation speed and having high latency to download the data. The mobile clients are to selectively download the data of their interests by using indices. However, they cannot process XML twig pattern queries efficiently since they do not contain branching information or parent-child relationships. To overcome this kind of problems in our proposed system use the Lineage Encoding scheme for processing the Twig pattern queries in wireless mobile environment. In our previous or existing system the data are streamed in wired environment. The messages are broadcast via the wired network over the mobile environment in our previous system. It requires more computation time and latency speed to done this process. To overcome this problem we use the wireless network for broadcast the messages via the internet. The multiple customers are handled and the efficiency for this approach is high. It requires less latency to broadcast the client requested messages. 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 components are highly accurate under various conditions [4].

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. 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. 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

V. A NOVEL ENCODING SCHEME: LINEAGE ENCODING

This novel encoding Scheme support twig Pattern matching. 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 [4].

5.1 Evaluating a twig pattern query

In evaluating a given twig pattern query with predicates, we should select a subset of elements of a particular type satisfying the given predicates. Then, for the selected elements, we should find their parent elements or child elements of a particular type [1].

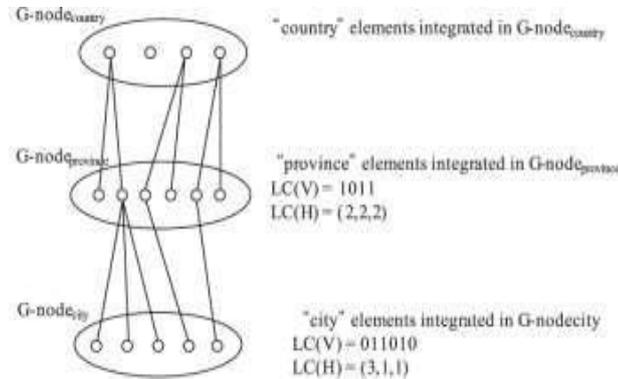


Fig.2. Lineage codes of G-nodcountry,G-nodeprovince and G-nodcity

VI. UNIT STRUCTURE OF THE STREAM: G-NODE

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.

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.

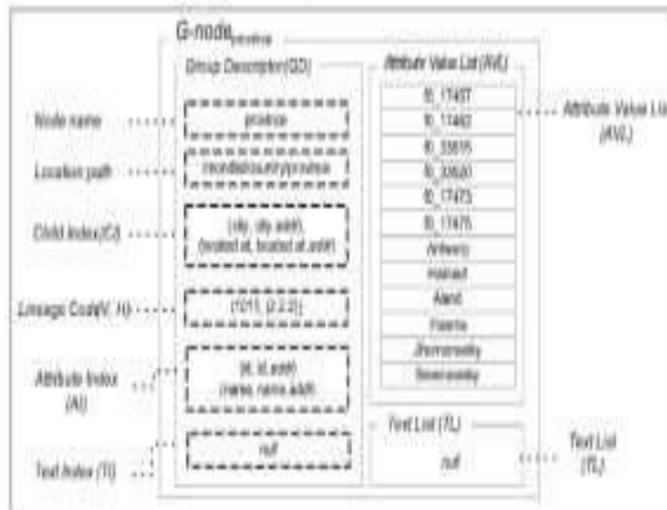


Fig. 3. Structure of an example G-node

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. 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[4].

VIII. LEL (LEAST EFFICIENT LAST) SCHEDULING ALGORITHM

In on-demand XML data broadcasts, the sizes of data items can vary in a very wide range. Thus data item size should be taken into consideration. First of all, we introduce some notations

- di : a data item (XML file) stored in the server
- D : the set of data items that will be broadcast.
 $D = d1; d2; : : : ; dm$
- Li : the length of data item di
- qi : a query issued by one or more mobile users
- Q : the query set $Q = q1; q1; : : : ; qn$ that were issued during the last broadcast cycle
- $QS(di)$: the query set in which all queries require data item di
- $freq(qi)$: the access frequency of qi
- σ : the broadcast schedule of a broadcast cycle
- $FQS(di)$: given a schedule σ , the query set that will be fully satisfied by item di

For a given schedule σ and a given query set Q , it is easier to identify what queries will be fully satisfied when a data item is broadcast if examining from the last item to the first item in schedule σ . For example, if dm is the last item of schedule σ and is required by $q1, q2$ and $q3$, then all these three queries will only be fully satisfied after they receive dm . Then after removing dm from σ and removing $q1, q2$ and $q3$ from Q , we perform the same check on the updated σ and Q . Taking similar steps, we can work out what queries (other than $q1, q2$ and $q3$) will be fully satisfied by the new last item of σ . By doing this repeatedly, the access time of all queries could be easily figured out. Moreover, since the last broadcast item produces the longest access time, it is reasonable that the last item should have the least access frequency. Furthermore, taking the lengths of data items into account, we propose to broadcast the least efficient data item as the last item in a schedule [3].

The efficiency of a data item di can be defined as follows:

$$Eff(di) = \frac{\sum_{q \in FQS(di)} freq(q)}{Li}$$

Based on this definition, when we schedule a new broadcast cycle, we first examine the efficiency of each data item and then select out the less efficient items to broadcast later and those items with higher efficiency will be broadcast earlier. The LEL (Least Efficient Last) scheduling algorithm can be described in the following.

Note that, in order to calculate $Eff(di)$ for the first scheduled data item di , we initially set all $FQS(d)$ the same as $QS(d)$ for every item d in D . Moreover, suppose the data item set D contains m items, then step 1 and step 4 both take $O(m)$ time and step 1 and step 4 will both repeat m times. Therefore, the computing complexity of LEL scheduling algorithm is $O(m^2)$. LEL Scheduling Algorithm:

1. select an item d from data item set D which has the smallest $Eff(d)$
2. place item d in the last vacant position of broadcast schedule σ
3. remove item d from D
4. update $FQS(d0)$ for every item $d0$ in D
5. repeat step 1 to step 4 until D becomes empty

IX. CONCLUSIONS

The previous work support only simple path queries. Thus, they are inefficient for twig pattern queries, because a twig pattern query consists of two or more path expressions. This combined scheme provides an energy and latency efficient way to evaluate predicates and twig pattern matching by using Lineage Encoding and scheduling algorithm. This scheme explores the benefits of the structure indexing and attribute summarization, so it reduces the size of the xml stream. 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. Therefore, by using LEL scheduling algorithm, the access efficiency can be improved significantly. We then devise a Least Efficient Last (LEL) scheduling algorithm to effectively organize XML data on wireless channels.

REFERENCES

- [1] Park, Jun Pyo, Park, Chang-Sup; Chung, Yon Dohn "Lineage Encoding: An Efficient Wireless XML Streaming Supporting Twig Pattern Queries" Knowledge and Data Engineering, IEEE Transactions on Volume: 25, Issue: 7 Publication Year: 2013, Page(s): 1559 - 157.
- [2] Marouane Hachicha and Jerome Darmont "A Survey of XML Tree Patterns", IEEE transactions on knowledge and data engineering, vol. 25, no. 1, January 2013

- [3] Yongrui Qin¹ Hua Wang¹ Jitian Xiao² "Effective Scheduling Algorithm for On-Demand XML Data Broadcasts in Wireless Environments" Australian Computer Society, Inc. 22nd Australasian Database Conference(ADC 2011), Perth, Australia, January 2011
- [4] S. Vidya, R. Uma "Dynamic XML Dissemination Supporting Twig Pattern Queries" International Journal of Advanced Information Science and Technology (IJAIST) ISSN: 2319:2682 Vol.17, No.17, September 2013.
- [5] D. Carlisle, P. Ion, and R. Miner, "Mathematical Markup Language (MathML) Version 3.0," World Wide Web Consortium(W3C), <http://www.w3.org/TR/MathML/>, 2010
- [6] P. Murray-Rust and H. Rzepa, "Chemical Markup Language -CML," <http://www.xml-cml.org/>, 1995.
- [7] R. Lake, D.S. Burggraf, M. Trninic, and L. Rae, Geography Mark-Up Language: Foundation for the Geo-Web. Wiley, 2004.
- [8] ADL, "SCORM 2004 Fourth Edition Version 1.1 Overview," Advanced Distributed Learning(ADL), <http://www.adlnet.gov/Technologies/scorm/>, 2004
- [9] 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
- [10] Scott, M. A., Comer, P. G." The Strategic Importance of XML Applications" School of Computing, Engineering and Technology University of Sunderland
- [11] 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.
- [12] 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
- [13] 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.