ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 1, January 2016

XML Tree Pattern Matching Algorithms: Survey

Gajanan Patle[1], Pragati Patil[2]

Student, Dept. of CSE., Abha Gaikwad-Patil College of Engineering, Nagpur, India[1]

Asst. Professor, Dept. of CSE., Abha Gaikwad-Patil College of Engineering, Nagpur, India[2]

ABSTRACT: Due to the business collaborations and for the purpose of portability enterprises are storing data in XML format. This has become a common practice as XML is portable and irrespective of platforms in which applications were developed, they can share information through XML file format. Such XML files are also validated using DTD or Schema. XML parsers are available in all languages that facilitate the usage of XML programmatically.

XML has become a defacto standard to store, share and exchange business data across homogenous and heterogeneous platforms. The interoperability is possible though XML. As enterprises are generating huge amount of data in XML format, there is a need for processing XML tree pattern queries. The existing holistic algorithms for XML tree pattern matching queries exhibit sub-optimality problem as they consider intermediate results before taking final results. This causes suboptimal performance. This sub-optimality is overcome by using TreeMatch algorithm. This paper implements a prototype application that makes use of dewey labelling scheme to overcome sub-optimality. The experimental results revealed that the proposed algorithm is better than the existing algorithms.

KEYWORDS: XML tree, holistic algorithm, Dewey labelling, DTD and Sub-optimality.

I. INTRODUCTION

The eXtensible Markup Language (XML) has emerged as a standard for data representation and exchange over the Internet, as many (mostly scientific, but not only) communities adopted it for various purposes, e.g., mathematics with MathML, chemistry with CML, geography with GML , and e-learning with SCORM, just to name a few. As XML became ubiquitous, efficiently querying XML documents quickly appeared primordial and standard XML query languages were developed, namely XPath and XQuery. Research initiatives also complemented these standards, to help fulfil user needs for XML interrogation, e.g., XML algebras such as Tree Algebra for XML (TAX) and XML information retrieval [1].

Efficiently evaluating path expressions in a tree-structured data model such as XML's is crucial for the overall performance of any query engine. Initial efforts that mapped XML documents into relational databases queried with SQL induced costly table joins. [1] Thus, algebraic approaches based on tree-shaped patterns became popular for evaluating XML processing natively instead. Tree algebras indeed provide a formal framework for query expression and optimization, in a way similar to relational algebra with respect to the SQL language.
In this context, a tree pattern (TP), also called pattern tree or tree pattern query (TPQ) in the literature, models a user query over a data tree. Simply put, a tree pattern is a graphic representation that provides an easy and intuitive way of specifying the interesting parts from an input data tree that must appear in query output. More formally, a TP is matched against a tree-structured database to answer a query.

XML tree pattern queries are to be processed efficiently as that is the core operation of XML data. Recently many researchers developed various methods or algorithms for processing XML tree queries. [2] A stack based algorithm was proposed by Khalifa et al. that matches relationships such as A-D, and P-C. TwigStack is another algorithm proposed by Bruno et al. for the purpose of XML tree pattern queries. However, the drawback of these algorithms is that they take unnecessary intermediary nodes while processing the query thus causing more time to process. To overcome the drawbacks indexing concept is used by algorithms provided in and. Some other algorithms use labelling schemes. In industrial and academic applications these algorithms have proven to be highly promising.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 1, January 2016

II. WHAT IS XML

Extensible Markup Language (XML) is markup language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable. XML is a text-based markup language that is fast becoming the standard for data interchange on the web. As with HTML, you identify data using *tags* (identifiers enclosed in angle brackets: <...>). Collectively, the tags are known as markup. The design goals of XML emphasize simplicity, generality, and usability over the Internet. It is a textual data format with strong support via Unicode for different human languages. Although the design of XML focuses on documents, it is widely used for the representation of arbitrary data structures [3].

III. USE OF XML

There are several basic ways to use XML:

- Traditional data processing, where XML encodes the data for a program to process
- Document-driven programming, where XML documents are containers that build interfaces and applications from existing components
- Archiving--the foundation for document-driven programming--where the customized version of a component is saved (archived) so that it can be used later
- Binding, where the DTD or schema that defines an XML data structure is used to automatically generate a significant portion of the application that will eventually process that data

IV. XML TREE

Due to the business collaborations and for the purpose of portability enterprises are storing data in XML format. This has become a common practice as XML is portable and irrespective of platforms in which applications were developed, they can share through XML file format. Such XML files are also validated using DTD or Schema. XML parsers are available in all languages that facilitate the usage of XML programmatically. Moreover XML is tree based and it is convenient to manipulate easily using DOM (Document Object Model) API. XML tree pattern queries are to be processed efficiently as that is the core operation of XML data. Recently many researchers developed various methods or algorithms [2], [3], [4], [5], [6], [7] for processing. 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. XML documents have a hierarchical structure and can conceptually be interpreted as a tree structure, called an XML tree. XML documents must contain a root element (one that is the parent of all other elements). [4][5]All elements in an XML document can contain sub elements, text and attributes. The tree represented by an XML document starts at the root element and branches to the lowest level of elements.

- The terminology used in the XPath Data Model
- The terminology used in the XML Information Set.

XPath defines a syntax named *XPath expressions* that identifies one or more internal components (elements, attributes, etc.) of an XML document. XPath is widely used to accesses XML-encoded data. The XML Information Set, or XML infoset, describes an abstract data model for XML documents in terms of information items. It is often used in the specifications of XML languages, for its convenience in describing constraints on constructs those languages allow. An Example XML Document, XML documents use a self-describing and simple syntax:[6].

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 1, January 2016

```
<book>
<title> XML < =title>
<all authors>
        <author> jane < =author>
        <author> john < =author>
< =all authors>
<year> 2000 < =year>
<Chapter>
        <head> Origins < =head>
        <Section>
                <head> ...< =head>
                <section> ...< =section>
                < =section>
                <section> ...< =section>
                < =chapter>
                <chapter> ...< =chapter>
                        < =book>
```

Fig.1. Sample XML document

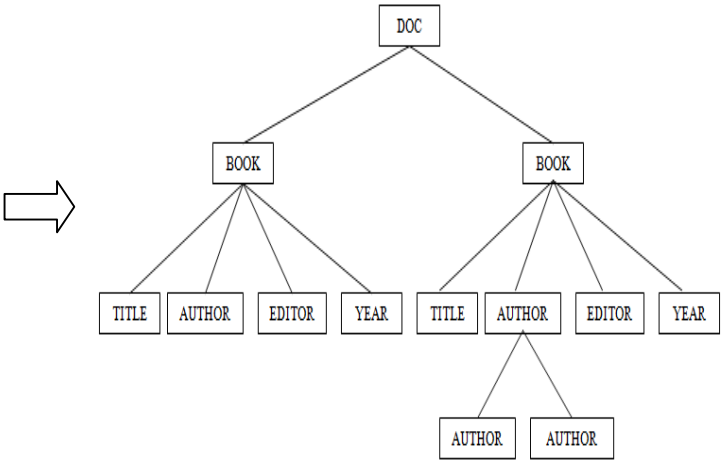

Fig. 2. Tree representations

From the sample XML document of Fig.1 and its tree representation is shown in Fig.2. Queries in XML query languages like XQuery, and XML-QL make fundamental use of node labelled tree patterns for matching related portions of data in the XML database. The query pattern labels which consists of element tags, attribute-value comparisons and string values, and the query pattern edges which include the parent-child edges. This query pattern would match the document in Figure1. In general, at each node in the query tree pattern, that specifies the node predicate on the attributes e.g., tag, content of the node [7][8].

V. LITERATURE SURVEY

1. Marouane Hachicha and Jerome Darmont, Member, IEEE Computer Society "A Survey of XML Tree Patterns", IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING VOL:25 NO:1 YEAR 2013.

This paper is a comprehensive survey of this topic, in which I outline and compare the various features of tree patterns. I also review and discuss the two main families of approaches for optimizing tree pattern matching, namely pattern tree minimization and holistic matching. We finally present actual tree pattern-based developments, to provide a global overview of this significant research topic.

2. Jiaheng Lu. "Benchmarking Holistic Approaches to XML TreePattern Query Processing"

This paper presents a proposed the problem of XML tree pattern matching and surveyed some recent works and algorithms. This comprehensive benchmarking compared five holistic algorithms and demonstrated their efficiency and scalability. There is no clear winner in all scenarios in our experiments. But TreeMatch has an overall good performance in terms of running time and the ability to process generalized tree patterns.

3. Jiaheng Lu, Tok Wang Ling, Senior Member, IEEE, ZhifengBao, and Chen Wang. "Extended XML Tree Pattern Matching: Theories and Algorithms". IEEE transactions on knowledge and data engineering, vol. 23, no. 3, march 2011.

This paper introduced a notion of matching cross to address the problem of the sub optimality in holistic XML tree pattern matching algorithms. In this identified a large optimal query classes for three kinds of queries, that is Q=;==;_, Q=;==;_;<, and Q=;==;_;<;:, respectively. And proposed a new holistic algorithm called TreeMatch to achieve such theoretical optimal query classes. Finally, extensive experiments demonstrate the advantage of our algorithms and verify the correctness of theoretical results.

4. Mirella M. Moro, Zografoula Vagena, Vassilis J. Tsotras, "Tree-Pattern Queries on a Lightweight XML Processor".

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 1, January 2016

This paper discusses proposed a classification of tree-pattern query processing algorithms considering important features such as data access and matching process, and adapted previous and successful XML query processing techniques for handling tree-pattern queries as well. Specifically, adjusted a DFA-based approach, and improved its performance by accessing nodes from a B+-tree instead of purely sequential scan. Such an improvement provided better results in comparison to the plain DFA. and also introduced a generalization that examines all left-deep plans.

5. Kamala Challa, E.Jhansi Rani "Algorithms for XML Tree Pattern Matching and Query Processing" Int.J. Computer Technology & Applications, Vol 3 (1), 447-451 JAN-FEB 2012.

This paper proposed the problem of XML tree pattern matching and surveyed some recent works and algorithms. Two algorithms TreeMatch and TJfast have introduced. TreeMatch has an overall good performance in terms of running time and the ability to process generalized tree patterns.

6. M.Muthukumaran1, R.Sudha2 "Efficiency of Tree Match Algorithm in XML Tree Pattern Matching" IOSR Journal of Computer Engineering (IOSRJCE) ISSN: 2278-0661 Volume 4, Issue 5 (Sep-Oct. 2012), PP 19-26.
This paper presents a wide analysis to identify the efficiency of XML tree pattern matching algorithms. TreeMatch has an overall good performance in terms of labeling schemes, optimality, query processing, output list and the ability to process extended XML tree patterns (twigs). In this TreeMatch to achieve such optimal query classes so, from this points that TreeMatch twig pattern matching algorithm can answer complicated queries and has good performance.

7. J. T. Yao M. Zhang "A Fast Tree Pattern Matching Algorithm for XML Query"

This paper proposed a TreeMatch algorithm to directly find all distinct matching of a query tree pattern in XML data sources. Unlike prior research for query tree pattern matching, the TreeMatch algorithm does not need to decompose the tree pattern into linear patterns and do not produce any intermediate results that are not part of the final results. The TreeMatch algorithm is applicable when the non-leaf pattern nodes do not have occurrences with self containment. The self-containment is seldom found in real XML documents and such a property can easily be identified. Therefore, the TreeMatch algorithm was more efficient than the existing methods under most cases.

8. N. Kannaiya Raja "A Novel XML Documents Using Clustering Tree Pattern Algorithms"

This paper identified a large optimal query classes namely that is Z\,\\,α; Z\,\\,α, β and Z\,\\,α, β ,γ respectively and also introduced a notion of matching cross to address the problem of the suboptimality in holistic XML clustering tree pattern matching algorithms. Based on results, planned a new holistic algorithm called TreeMatch to achieve such abstract optimal query classes. And, general experiments demonstrate the advantage of the algorithms and verify the accuracy of abstract results.

VI. COMPARATIVE STUDY

TP mining actually summarizes into discovering frequent sub trees in a collection of TPs. It is used, for instance, to cache the results of frequent patterns, which significantly improve query response time, produce data warehouse schemas of integrated XML documents from historical user queries, or help in website management by mining data streams. With XML becoming a ubiquitous language for data interoperability purposes in various domains, efficiently querying XML data is a critical issue [9][10]. Efficiently evaluating path expressions in a tree-structured data model such as XML's is crucial for the overall performance of any query engine.

Here recapitulate in Table 1 the characteristics of all TPs surveyed in this with respect to the comparison criteria.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 1, January 2016

Table 1: summary of algorithm analysis

Algorithms	Labeling Scheme	Optimality	Query	Output List
TwingStack	Containment	Optimal in terms of output sizes and not optimal for PC	Unordered	Many useless intermediate results when query contains p-c relationship
OrderdTJ	Containment	Not an optimal	Ordered	Much less intermediate results
TJFast	Extended Dewey	Not fully optimal	Unordered	One useless intermediate path and it is outputs the path solution for all nodes in query
TreeMatch	Extended Dewey And Bitvector	Fully optimal	Ordered restriction, Negation and Wildcard	No useless path

Based on previous detailed discussions, table 1 illustrates the comparative analysis of previous tree pattern matching algorithms with TreeMatch with the key factors of labeling schemes, optimality, and query and output list.

VII. CONCLUSION

From the above literature survey and comparative study it is observe that lots of algorithms implemented on XMLTree optimization and it is also observe that Sub-Optimality is major issue in XMLTree pattern to increase searching performance.

There are lots of algorithm works to increase the performance but it has linear search approach and it does not work efficiently with large amount of data at that time with XPath query we need to use the algorithm which overcome the problem of linear search and implement the technology which increase searching pattern.

REFERENCES

1. D. Suresh Babu et al, B.Shiva kiran "*Extended XML Tree Pattern Matching Using TREEMATCH Algorithm*" (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 3 (5) , 2012,5210 – 5211.
2. Sravan Kumar K, Madhu P, Raghava Rao N "*Efficient Handling of XML Tree Pattern Matching Queries – A Holistic Approach*" International Journal of Advanced Research in Computer and *Communication* Engineering Vol. 1, Issue 8, October 2012.
3. M.Muthukumaran1, R.Sudha2 "*Efficiency of Tree Match Algorithm in XML Tree Pattern Matching*" IOSR Journal of Computer Engineering (IOSRJCE) ISSN: 2278-0661 Volume 4, Issue 5 (Sep-Oct. 2012), PP 19-26.
4. N.Kannaiya Raja, M.E., (P.hd),2Dr. K. Arulanandam, Prof and Head,3P. Umadevi, M.E., (A/P), 4A.Balakrishnan, M.E "*A Novel XML Documents Using Clustering Tree Pattern Algorithms*" International Journal of Computer Network and Security (IJCNS) Vol 4. No 1. Jan-Mar 2012 ISSN:0975-8283.
5. Kamala Challa, E.Jhansi Rani "Algorithms *for XML Tree Pattern Matching and Query Processing*" Int.J. Computer Technology & Applications, Vol 3 (1), 447-451 JAN-FEB 2012.
6. Jiaheng Lu, Tok Wang Ling, Senior Member, IEEE, ZhifengBao, and Chen Wang. "*Extended XML Tree Pattern Matching: Theories and Algorithms*". IEEE transactions on knowledge and data engineering, vol. 23, no. 3, march 2011.
7. M. Shalem and Z. Bar-Yossef, "*The Space Complexity of Processing XML Twig Queries over Indexed Documents*," Proc.24th Int'l Conf. Data Eng. (ICDE), 2008.
8. A. Trotman, N. Pharo, and M. Lehtonen, "*XML-IR Users and Use Cases*," Proc. Fifth Int'l Workshop of the Initiative for the Evaluation of XML Retrieval (INEX '06), pp. 400-412, 2006.
9. S. Chen, H.-G.Li, J. Tatemura, W.-P.Hsiung, D. Agrawal, and K.S.Candan, "*Twig2stack: Bottom-Up Processing of Generalized- Tree-Pattern Queries over XML Document*," Proc. Int'l Conf. Very LargeData Bases (VLDB), pp. 19-30, 2006.
10. H. Wang and X. Meng, "*On the Sequencing of Tree Structures for XML Indexing*," Proc. 21st Int'l Conf. Data Eng. (ICDE), pp. 372- 383, 2005.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 1, January 2016

11. J. Lu, T.W. Ling, C. Chany, and T. Chen, "*From Region Encoding to Extended Dewey: On Efficient Processing of XML Twig Pattern Matching*," Proc. Int'l Conf. Very Large Data Bases (VLDB), pp.193-204, 2005.
12. M. Moro, Z. Vagena, and V.J. Tsotras, "*Tree-Pattern Queries on a Lightweight XML Processor*," Proc. Int'l Conf. Very Large DataBases (VLDB), pp. 205-216, 2005.
13. P. ONeil, E. O'Neil, S. Pal, I. Cseri, G. Schaller, and N. Westbury, "*ORDPATHs: Insert-Friendly XML Node Labels*," Proc. ACMSIGMOD, pp. 903-908, 2004.
14. H. Jiang et al., "*Holistic Twig Joins on Indexed XML Documents*," Proc. Int'l Conf. Very Large Data Bases (VLDB), pp. 273-284, 2003.
15. B. Choi, M. Mahoui, and D. Wood, "*On the Optimality of the Holistic Twig Join Algorithms*," Proc. 21st Int'l Conf. Database and Expert Systems Applications (DEXA), pp. 28-37, 2003.