International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2013): 4.438

Comparative Analysis of Algorithms for Query Processing in XML TREE

Vijaya D Kamble[1]

[1]Assistant Professor. Department of CSE, Zewari College of Engineering. Chandrapur
Email: kamblevijaya30@gmail.com

Abstract: ***Today is world of data processing from various Datasets. This data sets are used by business section and also by enterprises..XML has International format for data processing through domains with heterogeneous and homogenous platform. The XML has provided B2B integration .For query processing in XML file ,we need XML datasets. The XML file are checked for design and schema. An XML documents can be presented using tree structure. We present techniques which exactly matches pattern with XML tree .Whenever a query is inputted by user, it is matched with XML tree. The pattern matching algorithms is used for processing .Here we analysis the algorithms for query processing. The TwigStack and Tree Matching algorithms are used. The comparative study is done for XML query processing tree***

Keywords: XML, XQuery, TreeMatching, TwigStack,

1. Introduction

Data Mining is an analytical process designed to explore data in search of consistent pattern between variables and then to validate the findings by applying the detected patterns. Here we are using Data Mining techniques which work upon huge datasets of XML. We design Query to extract information from XML document. The Query language query not only content, but also the structure. The XML documents are presented as tree. XML document representation is done using DOM parser. The XML parser converts XML into XML DOM object. DOM is document object model. DOM parser is used to access and manipulate XML tree. The XML DOM contains method to transverse XML trees and access it. However to access XML document, it must be loaded into an XML Dom object. Here we use XQuery for querying XML. XQuery is a language for finding and extracting elements and attributes from XML documents. XQuery is supported by all major databases. The language use some complex symbols to perform query processing. XPath uses path expressions to navigate in XML documents. The reviews tell that we can follow various techniques for processing the document. We start with TwigStack algorithm [4]. The flaws that are found worked and reviewed in Tree Matching Algorithm[2].A Pattern Matching Algorithm Call TreeMatching[10] is used to overcome to suboptimality problem faced by existing system. This algorithm uses the Dewey labeling. As per labeling schemes, for the root node, children, and grand children etc. a number or label is associated. This predicte value of node. is now the important task is matching the tree structure with XML Querry. The processing of this can be seen is later part of the paper. The review of paper is organized as follows. Section 2. explain of TwigStack algorithm followed by Section 3. Explain the TreeMatching algorithm. Section 4 we do the comparative analysis and overview.

2. TwigStack Based

Traditionally, XML query language like XQuery and XPath were used for query processing . in an XML files. Starting tree representation of Query is shown in Figure 1

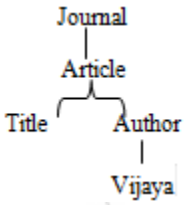

Figure 1: Tree representation of Query

XQuery and XPath are complicated system and they are not user friendly. A stack based algorithm[10] was presented by khalifa. It matches the parent and child and the accentor and decendent. The draw back this algorithm was, it produced useless intermediate steps. Then TwigStack is pattern matching algorithm was proposed by Bruno et al. The algorithm has some flaws. The query are worked on basis of Parent Child(P-C) and Ancestor Descendent (A-D) relationship. It uses the symbol for representation ie P-C edge is denoted by / and A-D is denoted by //. The TwigStack. is obtained by merging the intermediate result of query processing. Algorithm is working on divide and conquers technique. It uses the technique of decomposition matching and merging algorithm. The work process starts with query. Each is query is decomposed separately. The result of sub query is stored separately and executed separately. The final result is merged.. This algorithm works on the principal of decomposition .Each query and sub query is break down into sub query. Each query is having independent path of execution. The final result is obtained by grouping intermediated result. The TwigStack algorithm follows.

```
//Phase 1
1: While notEnd (q)
2: qact = getNext (q)
3: If (isNotRoot (qact)) then
4: cleanStack (parent (qact), nextL (qact))
5: end if
6: if (isRoot (qact) or isNotEmpty (Sparent (qact))) then
7: cleanStack (qact,, next(qact))
8: moveStreamToStack (Tqact, Sqact, pointerto top (Sparent (qact)))
9: if (is Leaf (qact)) then
```


Licensed Under Creative Commons Attribution CC BY

10: show Solutions WithBlocking (Sqact, 1)
11: pop (Sqact)
12: end if
13: else
14: advance (Tqact)
15: end if
16: end while
//Phase 2
17: merging AllPathSolutions ( )

Algorithm TwigStack operates in two phases. In the first phase liens (1-16) , are executed, some but not all solution to individual query root to leaf path are computed. In the second phase their solution are merge joined to compute to answer to query Twig pattern.

3. Tree Matching

We have seen decomposition matching and merging process. The drawback of it we will see in the comparative part. The tree match system follows the keywords search technique. The data source matches the elements of non leafy pattern nodes that do not contain sub elements with the same tag. In XML query processing keyword search is followed. In the given system the exact pattern matching with XML tree is done. Figure 2 shows tree representation of XML document. It start with the root node with Dewey label for query matching.

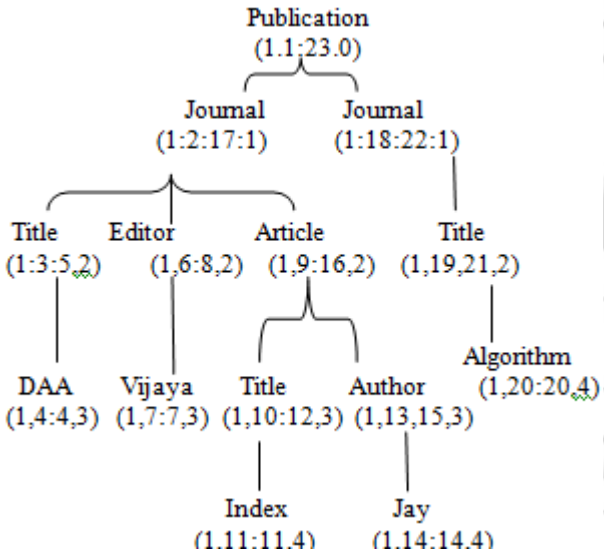

Figure 2: Tree representation of XML document

The concept of proposed system is implemented by following modules.

1) In MYSQL database. data transaction we insert data into the database in form of text image audio and video.
2) Create XML file. After storing the data into the database: we are printing those data into an XML format.
3) Views XML tree. We are using DOM parser to represent the XML file is the form of XML tree. Users can view the XML tree from the select XML file.
4) Searching using keyword. In this we are using keyword query to perform query answering in the XML tree. We create search engine. The search engine get input from user and perform query answering The input query is match with the XML tree and perform query processing easily.
5) Performance valuation is done by comparing the downloading time. The concept of tree matching algorithm is given as follows, query tree pattern is a tree Q = (Nq,Eq) where Nq is a set of labeled nodes and Eq is a set of edges. Each edge is represented by the pair or nodes. it connects. There are two kinds of edges Parents-Child edge Ancestor-Descendent edge. The basic idea of Tree matching algorithm is find all matching pattern recursively by calling function find (Q).

Tree Matching Algorithm
1: locateMatchLabel (Q);
2: while(endroot))do
3: f_{act} = getNext (topBranching Node);
4: if(f_{act} is return node)
5: addToOutputList (NAB (f_{act}, cur(Tf_{act}));
6: advance (Tf_{act});
7: updateSet (f_{act});
8: locateMatchLabel (Q);
9: emptyAllSets (root);

Explanation :
Trace the first element whose path match to the individual root leaf path pattern. After each iteration the leaf node is selected by each iteration. Add the matching element to the output list. Read the next element is tree and update the set an encoding. Locate the next element with matching path . Finally when all data is processed empty all the sets. In the given algorithm, the procedure addToOutputList(q, ,e_{qi}) we add the potential query answer ,e_{qi} to the set of S_{eq} where q is the nearest ancestor branching node of qi(ie.NAB(qi)=q) Procedure updates do three tasks. First it cleans the sets according to the current scanned elements. Second add element e into set and recursively update ancestor set of e.. The getNext function is core function in tree matching algorithm. It do two tasks, first task is to identify the next processed node. The second task is that before an element eb is inserted to the set S_b, we ensure that e_b is an ancestor of each other element $e_{bi,}$, so to match the node b in the corresponding solution path if there is more than one element to match the branching node b,e_{bi} is defined as their deepest element.

4. Comparative Analysis of Systems

XQuery and XPath system are complicated to understand by non database user. XQuery and XPath are not user friendly. The query analysis become query analysis become very complicated in this system. . Next is TwigStack algorithm. It produced large useless intermediate result for query having Parent-Child relationship and it reduced the size of intermediate result for Ancestor Descendent relationship. The twig pattern matching algorithm requires bounded main memory for small queries.

4.1 Proposed System

The Tree Matching algorithm is very much useful in query processing. It does not require any complex query languages like XPath and XQuery. It uses extend Dewey Label for query matching. In tree matching algorithm, as matching of pattern is against the data source. We need not decompose the query tree pattern. So it does not produce intermediated

Licensed Under Creative Commons Attribution CC BY

results and does not need merging. The final results are compactly encoded in stacks and explicit representation of the result is either a tree or relation with each tuple representing one matching, can be generated efficiently. Processing time of Tree matching algorithm is less compared to the decomposition matching and merging algorithm It does not produced useless intermediate result. It has less processing time comparative to other algorithm. As shown in figure 3. It also solves the sub optimality problem.

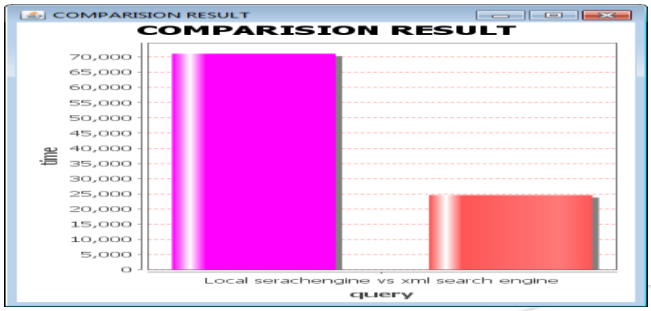

Figure 3: The downloading time of audio file in XML search engine is compared in local search engine.

5. Conclusion

In this paper we have analysis TwigStack algorithm and Tree matching algorithm. Tree Matching algorithm has overall good performance in terms of labeling schemes ,optimality ,and query processing

References

[1] Mirjann Mazuran,Elisa Quintarelli, and Letizia Tanca.(2012), "Data Mining for XML Query Answering Support." *IEEE Transactions on Knowledge and Data Engineering.*

[2] Jiaheng Lu, Tok Wang Ling, Zhifeng Bao and Chen Wang.(2011), "Extended XML Tree Pattern Matching Theories and Algorithm". *IEEE Transactions on Knowledge and Data Engineering.*

[3] Jiaheng Lu.(2013), "Orderd and Generlized XML Tree Pattern Processing." Springer Journal,pp. 120-145.

[4] M. Hachicha.(2013), a Survey on XML Tree Pattern. *IEEE Transactions on Knowledge and Data Engineering. IEEE Transactions on Knowledge and Data Engineering.*

[5] Choi. B, Mahoui M. and Wood.(2003), On the Optimality of the Holistic Twig Join Algorithm. Proc. Of DEXA. Page 28-37.

[6] Ling.T.W., Chan.C. and Chen.T.(2003), On efficient processing of CML Twig Pattern Matching. In VLDB. Pages 193-204.

[7] Chen Wang.j., Naughton. (2006), On supporting Containment Queries in Relational Data Base Management Systems. Proc. Of SIGMOD Conference, pages. 274-285

[8] Chien S., Vagena Z., Zhang D., and Tsotras (2002), EfficientStructural Joins on Indexed XML Documents. Proc. Of VLDB, pages. 263-274.

[9] S. Al-Khalifa, H. V.Jagadish, J.M.Patel, Y.Wu,N. Koudas, and D. Srivastava. *Structural Joins: A Primitive for Efficient XML Query Pattern Matching. In ICDE*, pages 141-152, February2002.

[10][10] D Bujji Babu ,Dr R.Shiva Rama Prasad and

[11] M.Santhosh ,"Twig Pattern Matching Algorithms for XML," .In IJARCSSE Vol-2 ,May 2012

Author Profile

Vijaya Kamble has done M.Tech. in Computer Science and Engineering from RCERT ,Chandrapur .She is Teaching professional WITH Chartered Engineer status with excellent track record spanning 14 years, with in-depth knowledge of Academics**.** She is working as a Asst. Professor is Zevari College of Engineering.

Licensed Under Creative Commons Attribution CC BY