Query optimization Pattern matching guide index table

Hamdollahghamgin

Instructor, Department of Engineering,Payame Noor University, po Box 19395-3697, IRAN.
Email: hamghamgin@yahoo.com

Abstract: Nowadays, XML is one of the most important models for storing and communicating data. The flexibility of XML leads to be used commonly In order to manage these documents, an expressive system is needed because traditional file systems can't manage and accountability such a huge volume of data. With the increasing number and size of documents that need to accelerate the implementation of database queries can be felt. And researchers trying to improve these methods is high.. So far, many methods have been proposed to solve this problem in the XML world, but each method only a small batch of queries and to respond to large query that does a lot of time spent on query processing are not appropriate. The result is still a certain way as a standard, traditional relational databases like SQL in there. In this thesis we are looking for a way to handle the query in the form of a large tree and will respond in less time and less number of nodes will reach. Direction indicators have a method that combines with the middle and lower data volume does. Way that does not pay directly to the query document. Since many articles are written in XML, Just as content is often focused on a particular subject and the angle is quite specific and limited time to review the cases and rarely large family of XML technology and XML standards are mentioned. This study describes the key elements associated with the XML technology and their relation to each of them to implement decisive and location information suppliers and planners are described. The performance of the proposed methods in this field, we, in any precise way to look at the strengths and weaknesses of each method are described. The third chapter presents the proposed method, and in his fourth season with the use of several tests on the performance pledge to prove it. We hope that this research bright aperture open to researchers.
[Hamdollahghamgin. **Query optimization Pattern matching guide index table.** *Academ Arena* 2013;5(12):62-68] (ISSN 1553-992X). http://www.sciencepub.net/academia. 8

Key words: Pattern matching guide index table. Query optimization

1. Introduction.

In the xml world since. We don't reach to standard methods such as sal in the relational database there are many methods in this case that we have shown their major mistakes. In bellow:
1- product intermediate data.
Increase answering time with increase length of query.
3- involving all the nods in the query to response.
4-application for a small group of query and pperators.
5-inconsistent with the methods used to document index.
Many researchers have tried to apply relationship traditional methods to manage documents also related to the xml(Jiang,2008).
But structure of a document in the frame of a tree is different than old and relationship structure. For example in the data relationship mode there is a table in a level and relationship between the tables also as bond easily to implement but in a document there are another relationship such as father. Child. grandfather. Generation (Kaushik,2002 Chien,2005).
So predicated complexity of users queries. And also changed the range of answers. on other side simple performance such as not also is not poSSible as simple as past.

In this article with changes that we represent in the index table method. Represent new model in the answering to types of queries this method proceSSing the nods as optimal.

So that in the ideal situation from proceSSing each node to reach a part of answer and don't unduly proceSSing nods.

- The works that were present in this article. Summery as follow:
- ability to answer to queries in leSS time.
- The same performance for all the queries in the xml expreSSed.
- New development of Dewey encoding use of this encoding method and use of route indexes makes it easier to reach the final answer.
- Compatible with all existing indexes.
- A method of optimizing the parameters of the schema document we describe the.

2.background (an overview of work).

Here with an example we describe the works. Suppose the Q1 query:

Q1.STUDENT // BOOK [TITLE='XML']

And now we must in document too reach TITLE so that, these nods be book's child and own nods of books be generation of STUDENT node. There are many methods for answering to this conformity such as STRUCTURAL join, HOLSTICTWIG JOIN for example in answer to Q1 with structural join this query charged to the numbers of binary link. (STUDENTBOOK), (BOOK/TITLE) and with decompose the query to father and child relationship or generation product more volumes of middle data(Hilippe, 2003).

Holistic twig join. Trying to solve this problem and don't analys this query to answer a query proceSSing all the existing nods in the query and all the poSSible adaptations test the documents(Yang. B, 2004; Garofalakis.M,2000;Goldman,1997) amony all the query nods. On the other hand for the structure of a document we have guides such as xml SCHEM.DTD, DATA, GUIDE STRUCTURAL SUMMARV, with the use of these we can achieved the schema for query (Bruno, 2007), that this schema can act as a query guide. in addition we have a lot of direction indicators. Such as TOXIN, STRONG, Data guide fabric, APEX, index, F & BALK) That trying indexed the nodes that are in the same path. And with this work answer to path queries with more quick (Hilippe, 2003; Kaushik,2005; Nestorov,1997; oldman, 1997). All the path indexes methods TJFAST, and etc…That mentioned in the world of xml are known to containment join this group of methods use of an index a named index.The work of this name index is fast acceSS to elements with the tags of same name for example suppose the follow equerry:

Q1: STUDENT // BOOK [TITLE='XML']:

For this quickly all the title, book , student, nods in the document available to algorithm. For example all the book nodes and put them in a order and orderly proceSSed them. In the all above methods there is a basic problem this group of methods without direction to elements place.

Just think to find a path for improved compare of binary nodes and of these compares trying to reach directly to query answer while many of these compares don't reach to none of query answer none of answer methods that proposed in above don't use efficiently and optimally of these path indexes (Hilippe 2003). These guides can help use for adjustment in the production of schema. Because don't proceSSing of query in the document blindly (O'Neil, 2004).

- **We mentioned briefly the problems and chalevyes that there is in this methods:**
  - Most methods that have been expreSSed need to acceSS to all nods. This involved a large number of proceSSing nodes useleSS and waste of time of proceSSer.
  - Foliar ways to spend that much time to decode the code in many cases simply not cost- effective.
  - Containment join ways RegardleSS of the location of the only respondents to compare the two groups are compared and don't use of document schema as a proceSSingguide(Al-halifa, 2008).
  - On the other side path index methods only whit h attention to document schema only for small range of query performance are neceSSary and for a big group of query will be need to acceSS to real document data too.
  - Most methods have acceptable performance only for rang of queries and with specific conditions for examples most methods have problems for answering to queries with not operator.

The proposed method.

3.the plan

Given the shortcoming of existing methods we need to have a plan that can meet the following objectives to be confirmed(Rizzolo,2001).

- Volume of data interface or the middle witch are not the final answer and only use for product to final answer to minimize.
- volume of useleSS data- the data but not the final answer and for data that are proceSSed to minimize.
- Directly don't compare the nods and combines with the path index method (Cooper. B,2001; Kaushik)
- For types of AND, NOT, OR operater. Performance is required.

4.document number method.

We chose number for documents in its method. First it better we describe this number then express the reason of choose it.

The method of deway number to this way ……. Node. The number of Unode have V node code number to as prefix and n is continuation of this code. For example if deway code be V=<1/3/7> node and U node be fifth child the U= <1/3/7/5> code will be for example. Note to encoding in the below tree.

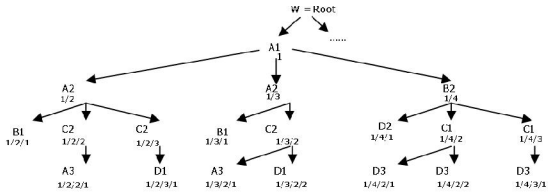


Figure (1-3) a sample of deway encoding.

The main reason for chosing deway is as a numbering method. The ability to see the nods 5.3.2.9.1 can easily understand that this node the path 5.3.8.9.5.3.8/5.3/5 has gone from the voot of tree.

So, with proceSSing each node we can get full list of it but another figure of this method is find the relationship between both comparision nods.

Using this numbering method can easily find another criterions such as difference between the two groups.

But this number method have a basic problem that at times it cant implement this problem is for saving. high volume of nods. for example suppose has been places that in this case we will need to 100 number of it for saving But we must note to two case of it(Cohen,2003):

- On the one hand. this problem occurs for big documents and other hands in big document such as BBLP grow the tree will be in width and no in depth.

So this length of nodes code in a tree is not depends on its volume.

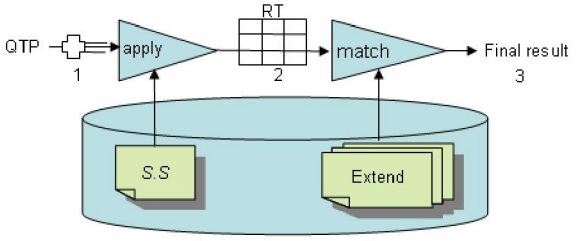
- In and offering a method for compreSS encoding Dewey that even in cases this method will be more efficient that other same one.

Alony side this problem Benefits such as the ability to update. quick acceSS to nods and has mixed performance of operators and wide spread use of this method.

5.The three stages of the index table schema.

Index table method is the mixed of path index join.

As the figure shows. this method has there stages but we expreSSe the summery of there stages.



THE three stages of the index table schema(2-2)

Step1: in this stage such as path index methods first query run on the SS. But here query initially and don't run with its complex shapes But query to some query of few queries that easily is broken in all path indexes methods answer and then all this single queries speratly run on the SS. the main reason of this stage is decrease of nods search

Range in this stage of SS use as a pattern adjust guide.

Step2.all single queries run separatly on SS. and SS. And each one turn a group of nodes as an answer. Note that this nodes are on the SS not reul document nonds.

Combination of results of this run cause to reach a run project named index table. the goal of this step is it product it as a proceSSing guide. comp ration paper nods in document and compare them and how to reach them to response shows severalty. use of it there isn't any neSSesary to deencodoing of nodes. it with use of nods deway code number can show the way of reach to answer.

Step3. the document is numbering according to Dewey encoding as in the previous section com the nodes related to each node order according to Dewey in Extend. now query step3 act similar to containment join. here due to it query paper nods that there is extend compare and product the final answer this staye scince have the need for acceSS for acceSS to real nods of documents hamed as a most time – consuming(Rizzolo,2004). **the goal of this stage is reach to final exam in attention to II guide**s.

6.Query guide.

Basically query guide is similar to a document schem. SS has a larg dependence with DTD and or xml schema of a document (Kim,2004). scheman of a document shows the structure and total relationship among elements and have little dependence with volume and larg of document datas the structure and size of is generally fix or is with little changes But in the xml world this SS to different methods Be made. for example in the SS (Jiang,2009; Kaushik,2007) product with different methods and each one have capabilities and features we further shoe that the its

index table can be made by all these methods. Because the work of this path indexes. is the performance sing Branch query on the SS(Chien,2005). this kind of queries is simply run in the above methods. and none of them don't need acceSS to original datas of documents. But since the first stage of our method is to run query on the SS. we here need to research of size SS and the time required (Fontoura,2005; Jiang,2009; Lu. J,2006).

- ✓ **for this stage fortunately in previous years two following criteria in the recent years been proved**.
- The SS volume is very little than real document datas volum(Braumandl,2001).

For example known datast Tree Bank, xmark, DBLP that order have 130, 897, 532 sizes. the volum of their schems are 2.8, 4.2/3 KB.

- SS of a document has little dependence with datas size in document in (29) that its SS with strong Data guide method. that have rather high volume than to other similar methods. tested for two Banks, sports and synthetic. to these Bank respectively aSSes 30695,375449. But only added 2012 to their SS according to above 2 criteria can be found that volume and size schem will be very little and in many cases independent of the size of the document as a result we can easily suppose the time for first beat of a little and fix time.

7. the choose of path index

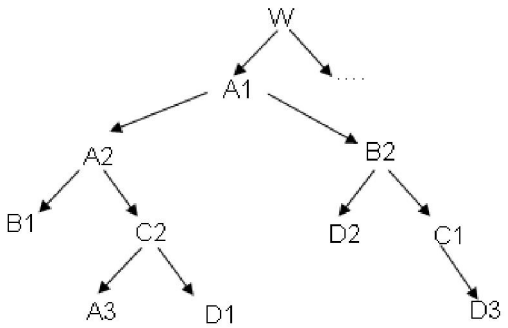
In the xml world SS Be made in different methods for example in SS product to various methods. and each one have specific features and capability. so there are many path index for election. But evey one of them have tried answer to compler queries alone. so for many of queries have need to reach to original data. and so don't result in efficient while the path index that we choose for our schem. is only for research on SS and for answer to single branch queries (Milo T,1999). so must only have been two features that is in below.

Its SS be small. and quickly answer to single branch queries for all the poSSible operators (*, ? /11) use for the single branch queries in amony all path indexes the best option that product only two above case is YAP that is low –cost and faster than single branch queries.

8.1 step 1.the run of single branch queries on the pattern adjustment guide

As was said we first must broke the query. this act of breaking done to this way that broken query to single query and than each single query run seperatly on the SS.

Fortunately in most methods of path index single branch queries simply answer on the SS and without reach to documents data in this stage we compare all the single queries with SS and scince encoding document due to Dewey. and have lower nodes Heterogeneous information of higher nodes only we will need result of this run will be import a list of points in SS for each single query. the addreSS of these points will be absolute



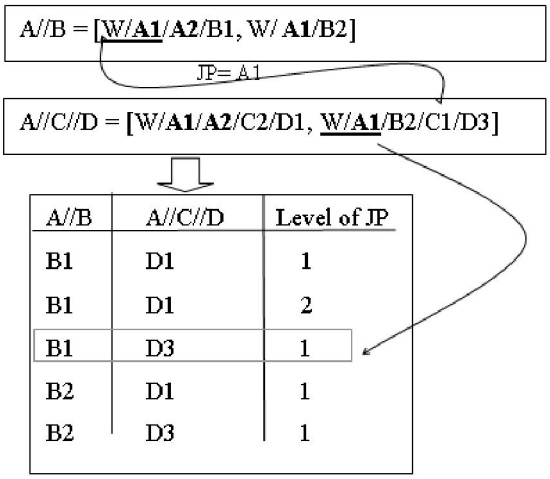
The figure of(3-3) for the hypothetical examples.

For example suppose the at we will run TPQ query on the below SS figure first broken the query to two single branch query A//C//D,A//B. Since the document encoding due to Dewey and low nods have erogenous in formation from own above nods-path form the root have in own and only to acceSS and maintenance queries paper will be for each branch. so for each branch A//B, B1, B2 nods and for A//C//D Branch D1, B3 nods are the answer single branch queries. don't need to say. that D2 nod scince don't have single branch. the answers are not brought. this points respectively are with absolute path w/A1/A2/B1W/A1/B2 FOR W/A1/B2/C1/B3, W/A1/A2/C2/D1/,All B Branch for A//C//B branch 8-2 staye2. product 8-8.2**Production index table.**

Define first index table:

IT is a column in table 3 that the first two columns of the paper nodes each query branch is in and third column of it point level connection is arroony two nods so each record of this table shows an act named connection act connection act when we compare two ore some node in the document to reach a part of answer after became query to sing branch queries and. obtained single branch queries paper leaf in SS. must obtained connection point among paper now result run of single branch queries on the SS find a list of nods for each single brunch queries now for obtain connection point among branches. from list of each branch leaf choose a node and compare their paths. if the path of chosen need were equal together.

add that two nods along with connection point level to it for example the result of run sing branch queries Q1 on the SS figure (1-3) figure to reach B1 node with W/A1/A2/B1for each branch A//B and obtained D1 node with W / A1/A2/B1 Pat and D3 node with w/a1/b2/c1/b3 path will be for A//C//B branch. this obtained path are absolutely. now we compare binary amony each branch if A,B be two members of comparies with during form a1/a2/… aj/…/ano aa and a1/a2/aj/…/am and be aj the connection point between two points andor in other words if the comparison be between two nodes. the path during root to query connection Doint here a was same. add a record along with connection point level for example two D3 B1 node to connection node have common point so add b1/d3/1> record to it. for example look at guide table in the figure (4-3).



A//B	A//C//D	Level of JP
B1	D1	1
B1	D1	2
B1	D3	1
B2	D1	1
B2	D3	1

Figure (4-3) a sample of index table the basic algoritm it produce

. for a two branch query. is below this is the initial pseudo code and only for two branches query in puti: aastopQ

Output: it asindex –table
1. let A and B the two leave sofa A , B are two leaf query
2. letyp = joint point between A and b
Node connection point show branch connection.
3. let Al = list of SS nodes match A branch.
4. let Bl = list of SS nodes match B branch.
AL, BL, Extend determined lists of A,B node. this node are the same leaf nodes are found **for each query branch.**
5. for each bn E AL do
6. for each bn E BL do
7. For each jp1 in an IP in bn do
8. if an prefix (jp1) = bn. prefix(jp2) then.
9. it add REC lan m bn, jp1. level
10. end if
11. end for
12. end for
13. end for

This lines performance the act of nodes two binary. note that each connection point may add a record to result table.

If any two comp rational have been same prefix to connection point level add are cord that is include a record that is two nodes and connection point level among them.

8.3 third step. the final result is generated according to results table

Each record of this table guide query proceSSer because to reach a piece of the answer. the community of this pieces product the final answer. so final results are same return results community by each record.

Method step three each record of index table have 3 fild. two fist column of two node is in SS and third column is connection point level among two node. as seen in figure (4-3) the real node document according to own Dewey code are in Extend respectively now Extend lists must compare with each other in SS. if the prefix two comparetion node was equal to connection point level they are only answer. this proceSS will continue until End –to reach one of two lists. this act named to adjustment process act. when we compare two or some nodes in document to reach to a part of answer.

Conclusion

We stated a modern method called three – step approach to improve earlier methods this method have 3 steps:
1. stage 1: simplification of query and. reduction of range searching.
2. production index table as a guide for query proceSSing.
3. proceSSing document nodes in according to production index table in previous step.
This three step method could its performance to prove in all previous methods.

We have developed this method so that performance for multi – land complex branches. and then we try expreSS the method as algebraic expreSSions. and with use of this expreSSions and statements we improve the methods. and then we try improve this method for another group of queries with weak operator.

In order to optimize procedure we reached to useleSS nods useleSS nods. are the nods that not production an answer for user. or return a duplicate. with use of Bt tree index we could jump to form on

this nods and reduction the number of node proceSSing to increase the performance of an inder called the level index and with use of it we could jump form on more nods. we hope to reach to this cases with more work represent new way that for answer to single queries. just proceSSing one branch. The present a new method for that in the time of run query we can make query guide and use of query guide can to reach guide table.

Represent full optimization method on the pattern adjustment guide.

Represent Dewey code to numeric or range if the adjustment pattern guide represent to a list of number range.

The use of this table avoided of compare the nodes blindry and try all the nods that proceSSing to produce an answer of problem for us. or in other hand be equal the number of proceSSing nods with answer nods. but we have interval to full equality.

The methods that we paid in this paper is know to optimization method of queries in terms of physical. we paid in this paper only optimization nods sue to physical location. but another background of queries recently considered by the researchers now day. few investigation are on the find the concept of between focused document elements and with this work whelp the user for find and reach to desired result.

Correspondence to:
Instructor, Department of Engineering, Payame Noor University, po Box 19395-3697, IRAN.
Telephone:0452-4254908
Email: hamghamgin@yahoo.com

References:

1.Al-Khalifa.S., Jagadish. H.V., Koudas.N., Patel. J.M., Srivastava. D., Wu. Y. Structural Joins: A Primitive for Efficient XML Query Pattern Matching. In Proc. ICDE: 141-152(2008)
2. Chien. Et. Efficient Structural Joins on Indexed XML, In Proc. VLDB Conference(2009)
3. Jiang, H., Wang, W., Lu, H., and Xu Yu, J, an Chin. B, XR-Tree: Indexing XML Data for Efficient Structural Joins. In Proc. ICDE Conference:253—264(2008)
4. Mathis. C., Härder. T, Haustein. M. Locking-Aware Structural Join Operators for XML Query ProceSSing, In Proc SIGMOD Conference: 467 - 478 (2008)
5. Wu. Y., Patel. M. J., and Jagadish. H. V. Structural join order selection for XML query optimization, In Proc. VLDB Conference,(2008)
6. Bruno. N., Koudas. N, Srivastava.D. Holistic Twig Joins: Optimal XML Pattern Matching, In Proc. SIGMOD Conference: 310–321(2007)
7. Chen. T., Lu. J., Ling. T. On boosting holism in XML twig pattern matching using structural indexing techniques, In Proc. SIGMOD Conference: 455 – 466(2005)
8. Fontoura. M., Josifovski. V., Shekita. E., Yang. B. Optimizing Cursor Movement in Holistic Twig Joins, In Proc. CIKM Conference: 784 – 791(2005)
9.Jiang, H., Wang, W., Lu, H., and Xu Yu, J. Holistic Twig Joins on Indexed XML, In Proc. VLDB Conference:273-284 (2009.)
10. Lu. J., Chen. T., and Ling. T. W. Efficient proceSSing of XML twig patterns with parent child edges: a look-ahead approach. In Proc. CIKM Conference: 533 - 542(2006)
11. Yang. B., Fontoura. M., Shekita. E., Rajagopalan. S., and Beyer. K. S.. Virtual Cursors for XML Joins, In CIKM: 523-532(2004)
12.Garofalakis. M. N., Gionis. A., Rastogi. R., Seshadri. S., Shim. K. XTRACT: A system for extracting document type descriptors from XML documents. In Proc. ACM SIGMOD Conference: 165 - 176 (2000)
13. Goldman. R., Widom. J. DataGuides: Enabling Query Formulation and Optimization in Semistructured Networks. In Proc. 23rd VLDB Conference: 436—445(1997)
14. Nestorov S., Ullman J., Wiener J., and Chawathe S., Representative Objects: Concise Representations of Semi structured, Hierarchical Data, In Proc. ICDE: 79– 90(1997)
15. Chung, C., Min, J., Shim, K. Apex: An adaptive path index for XML data. In Proc ACM Conference on Management of Data SIGMOD: 121 - 132(2005)
16. Cooper. B., Sample. N., Franklin. M., Hjaltason. G., Shadmon. M. A Fast Index for Semistructed Data, In Proc. 14th VLDB conference: 341 – 350(2001).
17. R. Kaushik, P. Shenoy, P. Bohannon, and E. Gudes. "Exploiting Local Similarity for Indexing Paths in Graph-Structured Data". In IEEE/ICDE, pages 129--140, San Jose, California, 2002.
18. Kaushik. R., Bohannon. P., Naughton J. and Korth. H, Covering Indexes for Branching Path Queries, In Proc. 11rd SIGMOD Conference: 133 – 144(2005)
19. Kaushik. R., Krishnamurthy. R., Naughton. J., and Ramakrishnan. R. On the integration of structure indexes and inverted lists, In Proc SIGMOD Conference: 779 - 790 (2002)
20. Milo T., and Suciu D.., Index Structures for Path ExpreSSions. In Proc. 7th.ICDT: 277 - 295(1999)

21. Rizzolo. F. and Mendelzon. A., Indexing XML Data with ToXin, in Proc.5th. WebDB conference（2001）
22. Dewey. M. Dewey Decimal ClaSSification System. http://www.mtsu.edu/~vvesper/dewey.html
23. O'Neil. P. E., Pal. S., Cseri. I., Schaller. G., Westbury. N., ORDPATHs: InsertFriendly XML Node Labels., In Proc. SIGMOD Conference: 903-908 (2004)
24. Ley. Chael, DBLP Computer Science Biblography, http://www.informatik.unitrier.de/ley/db/index.html(2003)
25. Yang, B., Fontoura, M., Shekita, E., Rajagopalan, S., and Beyer, K. S. "Virtual Cursors for XML Joins", *In CIKM,* 523-532(2004)
26. Hilippe. P, Mihaila. O, Siméon. J, "Integrating Efficient XPath Evaluation Algorithms in a Complete XQuery Implementation" *IBM(*2003)
27. Chien, Y, Zografoula. V, Zhang, D "staircasejoin implementation in a commercial disk-based RDBMS",. *In Proc. of the 21st ACM SIGMOD Conference,* 109-120(2005)
28. M. Tamer Ozsu, Patrick Valduriez, "Immediate and Partial Validation Mechanisms for the conflict resolution of Not operations in XML databases ". *In proc, PORDO* 365- 378(2005)
29. Cohen. S, Mamou, J, Yehoshua, Y "A Semantic Search Engine for XML" *In Prec, at VLDB*: 311-320(2003).
30. Kim, L "Integrating Keyword Search into XML Query ProceSSing" *in Proc, IEEE ICDE:* 345-354(2004)
31. ONALD, K "The State of the Art in Query ProceSSing' *In Proc ACM Sigmod,* 777-781(1999).
32. Papadimos, P. Maier, D "xml partitioning" *In Prof, CIKM ACM SIGMOD*: 453-349(2001)
33. Michael J. Franklin, KoSSmann, D. "Performance trade for client-server query proceSSing". *In Proceedings of the ACM SIGMOD Conference:* 149–160, (1996).
34. Braumandl, R. Keidl, M, KoSSmann, S. Seltzsam, and Konrad, S "ObjectGlobe: Open Distributed Query ProceSSing Services on the Internet." *IEEE Data Eng. Pages:*65–69(2001).
35. Trevor, J and Dan, S. "Dynamically Distributed Query Evaluation". *In Proc of the 12th*.

12/15/2013