Journal of

Software Engineering

ISSN 1819-4311

OPEN ACCESS Journal of Software Engineering

ISSN 1819-4311
DOI: 10.3923/jse.2016.356.364

Research Article

Research of XML Structural Join Based on the Cone Three-dimensional Coding

[1]Jiang Yan, [2]Fu Jiang-wei and [2]Wang Yu-Xuan

[1]School of Software, Shenyang University of Technology, 110870 Shenyang, China
[2]School of Information Science and Engineering, Shenyang University of Technology, 110870 Shenyang, China

Abstract

Background: This study proposes a new dynamic coding scheme- the three-dimensional XML coding based on cone, on the basis of the static coding scheme is extended, can effectively support the XML document update at the same time under the condition of the document is not updated with good performance. **Materials and Methods:** Solved the coding efficiency is low and the system cost is high caused by the entire XML tree need to re-encode when the document updates and solved the problems, such as the time and space overhead of the traditional dynamic coding scheme. **Results:** Due to the introduction of aspect-oriented thought, therefore, this study on the basis of analysis and comparison of the existing structural join algorithm, based on merging thought to solve structural join under multi-documents, proposes a structural join algorithm is suitable under multi-documents and can further reduce the cost of connection scanning by weaving document properties or index. **Conclusion:** Design the corresponding experiment and prove the document defines method, the coding scheme and the corresponding structural join algorithms proposed by this study have better performance and feasibility.

Key words: XML, coding scheme, weave, structural join, algorithm

Received: January 28, 2016 Accepted: April 11, 2016 Published: September 15, 2016

Citation: Jiang Yan, Fu Jiang-wei and Wang Yu-Xuan, 2016. Research of XML structural join based on the cone three-dimensional coding. J. Software Eng., 10: 356-364.

Corresponding Author: Jiang Yan, School of Software, Shenyang University of Technology, 110870 Shenyang, China

INTRODUCTION

One file format to be described in text form, XML (eXtended Markup Language) is becoming a mainstream form of data. The present study focus is on how to achieve efficient query XML data. Most XML related query technology based on modeling ability of XML DTD[1] and XML Schema[1,2] and the coding method to the XML tree[3,4] and the corresponding structural join algorithm.

Presently, in the aspect of structural join, some researchers have proposed a series of effective structural join algorithm, a kind of structural join algorithm MPMGJN can effectively realize containment join[5], but the circular scanning table will reduce the performance of the algorithm. In order to solve this problem, STD algorithm use the stack to store ancestor nodes may need to join[6] and respectively sequential scan once the AList list and DList list can realize the related structural join. Anc-Desc-B+algorithm[7], through the B+ tree index successfully skipped some nodes in the two lists can prejudge don't participate in the structural join, reduce the scanning costs and optimize the stack-tree algorithm. But these algorithms are mostly based on single document, when a document is inserted or updated nodes may lead to the query efficiency decreases. Due to the introduction of aspect-oriented thought, therefore, this study on the basis of analysis and comparison of the existing structural join algorithm, proposes a structural join algorithm is suitable under multi-document[8-10]. The researchers have proposed a number of dynamic coding method in relation to the issues, including floating-point number interval, CDBS and QED and so on[11].

MODEL AND DEFINITION OF THE XML DOCUMENT WITH THE CHARACTERISTICS OF AOP

The XML document of AOP characteristics is usually called AspectXML, that AspectXML is an XML document with AOP characteristics. Therefore, when the XML document is defined, on the basis of the object-oriented technology, introduced AspectXML document of aspect-oriented techniques XML model. Combining AOP characteristics and XML document definition formed AspectXML document. So, firstly presents the definition of AOP model and XML document model and then presents the definition of AspectXML document model.

Definition 1: Given a triad to AOP model, Aspect = (Vade, Vpt, Vadr). The Vade in the tuple is a set of advice, the Vpt is a set of pointcut, Vpt = $\sum_{i=1}^{n} \mathrm{jpi}$, jpi is the join point and the Vadr is a kind of binary relation, Vadr$\subseteq$Vade$\times$Vpt. If x$\in$Vade, y$\in$Vpt, then (x, y)$\in$Vadr represents y "activate" x.

Two basic elements and a binary relation contained in AOP model are given in definition 1 and through this binary relation shows the relationship between the two elements.

Definition 2: An XML document is an ordered tree, given an eight-tuple to an XML document, Xdoc = (ν, Ψ, τ, γ, R, E, A, T). In the tuple, ν represents the set of all elements, Ψ represents the set of attributes, τ represents the set of text, γ represents the document root element, R is a binary relation, R$\subseteq\nu\times\gamma$, if x$\in\nu$, y$\in\nu$, (x, y)$\in$R, represents the set of the root boundary, E is a binary relation, E$\subseteq\nu\times\nu$, if x$\in\nu$, y$\in\nu$, (x, y)$\in$E, represents the set of the element boundary, E is passed, reflexive and antisymmetric partial order relation, A is a binary relation, A$\subseteq\nu\times\Psi$, if x$\in\nu$, y$\in\Psi$, (x, y)$\in$A, represents the set of the attribute boundary, A is antisymmetric partial order relation, T is a binary relation, T$\subseteq\nu\times\tau$, if x$\in\nu$, y$\in\tau$, (x, y)$\in$T, represents the set of the text boundary, T is antisymmetric partial order relation.

Definition 2 shows the XML document model, a document tree hierarchical relation and affiliation relation between elements and attributes or text, which can be obtained according to R, E, A, T quaternary relation, that results in an XML document tree γ as the document root element.

Definition 3: The XML document with the characteristics of AOP, AspectXML = (ν, Ψ, τ, γ, R, E, A, T, Vade, Vpt, Vadr). Expand the three concept of AOP techniques in the XML document: Advice, pointcut, advisor. Using the static weave method of AOP techniques to realize the definition of the elements of the expanded XML document, this can avoid directly modify the elements in the XML definition document. Also can use the dynamic weave method of AOP techniques to realize to the elements in the XML document to create, update, delete.

Definition 4: In AspectXML, $e_1\in\nu$, $e_2\in\nu$, the set of elements require pre-defined in the XML document is λ(x). Then $\lambda(x) = \{x|(x, e_1)\in E\wedge(x, e_2)\in E\wedge(e_1, e_2)\notin E\wedge(e_1, \gamma)\in R\wedge(e_2, \gamma)\in R\}$, denoted by $\lambda(x) = e_1{\sim}e_2$.

Theorem 1: Set A, B, C as the element of the XML document, then $\lambda(x)$ = A~B~C = (A~B)~C = A~(B~C). Prove to prove theorem 1 is tenable, only need to prove that the left side and the right side of the equation are equal.

First of all, by definition A = A~A, therefore, to prove that (A~B)~C, only need to prove that (A~B) C = (A~B) (B~C). The $\lambda(y) = A{\sim}B = \{y|(y, A)\in E\wedge(y, B)\in E\wedge(A, B)\notin E\wedge(A, \gamma)\in R\wedge(B, \gamma)\in R\}$, $\lambda(z) = B{\sim}C = \{z|(z, B)\in E\wedge(z, C)\in E\wedge(B, C)\notin E\wedge(B, \gamma)\in R\wedge(C, \gamma)\in R\}$, $\lambda(x) = (A{\sim}B){\sim}(B{\sim}C) = \lambda(y)\wedge\lambda(z) = (A{\sim}B){\sim}C = A{\sim}B{\sim}C$.

In the same way A~(B~C) = A~B~C, QED. Corollary 1 by theorem 1 can extend pre-defined form of the element to finite elements, namely:

$$\lambda(x) = A{\sim}B{\sim}C\ldots N$$

Rule 1: In a AspectXML document, $e_1 \in \nu$, $e_2 \in \nu$, $(e_1, \gamma) \in R$, $(e_2, \gamma) \in R$, then element e_1 and element e_2 exist the same pre-defined part, denoted by $\lambda(e_1) \approx \lambda(e_2)$.

Rule 2: In a AspectXML document, $e_1 \in \nu$, $e_2 \in \nu$ assuming that exists $e_3, e_4, e_5, \ldots, en \in \nu$, $(e1, \gamma) \in R \wedge (e_2, \gamma) \in R \wedge (e_3, \gamma) \in R \wedge (e_4, \gamma) \in R \wedge (e_5, \gamma) \in R \wedge \ldots \wedge (e_n, \gamma) \in R$, then $\lambda(e_1) \approx \lambda(e_n)$. Exist root boundary of two elements in AspectXML document, contain the same part of the pre-defined elements. Because of the pre-definition has transitivity, then the same root boundary element also has corresponding transitivity.

Definition 5 : In AspectXML, $e_1 \in \nu$, $(e_1, \gamma) \in R$ the elements set of e_1 requires pre-defined is $\lambda(e_1)$, the same as above, $\lambda(\gamma)$ is the set of all require pre-defined of e_1 and e_1's sibling elements:

$$\lambda(\gamma) = \lambda(e_1) \cup \lambda(e_2) \cup \ldots \cup \lambda(e_n) = \sum_{i=1}^{n} \lambda(e_i),\ \lambda(e_i): e_1\text{'s sibling elements}$$

Rule 3: In a AspectXML document, $e_1 \in \nu$, $e_2 \in \nu$, $e_3 \in \nu$, $(e_3, \gamma) \in R$, if exists $(e_1, e_2) \in E$, $(e_1, e_3) \in E$, $(e_2, e_3) \in E$ then $\lambda(\gamma) = \lambda(e_1) \cup \lambda(e_2) \cup \lambda(e_3)$.

Rule 4: In a AspectXML document, $e_1 \in \nu$, $e_2\ e_1 \in \nu$, if exists $e_3, e_4, e_5, \ldots, e_n \in \nu$, $(e_n, \gamma) \in R$ and $(e_1, e_2) \in E$, $(e_1, e_3) \in E, \ldots, (e_1, e_n) \in E$, $(e_2, e_3) \in E$, $(e_2, e_4) \in E, \ldots, (e_2, e_n) \in E, \ldots, (e_{n-1}, e_n) \in E$ then $\lambda(\gamma) = \lambda(e_1) \cup \lambda(e_2) \cup \lambda(e_3) \ldots \cup \lambda(e_n)$.

Element e_1 with element e_2 and e_3 at the same time constitute element boundary relation in AspectXML document, element e_2 with e_3 also exist element boundary relation, then element γ should contain the pre-defined part of element e_1, e_2, e_3, as described in Rule 3. If γ exists multiple element boundary relation, then it can be generalized to Rule 4.

Definition 6: In AspectXML, $e_1 \in \nu$, $e_2 \in \nu$, $x \in \Psi$ the set of attributes require pre-defined is $\mu(x)$. Then $\mu(x) = \{x | (x, e_1) \in A \wedge (x, e_2) \in A \wedge (e_1, e_2) \notin E \wedge (e_1, \gamma) \in R \wedge (e_2, \gamma) \in R\}$ denoted by $\mu(x) = e_1 \infty e_2$.

Theorem 2: Set A, B, C as the element of the XML document, $x \in \Psi$, $(x, A) \in A$, $(x, B) \in A$, $(x, C) \in A$, then $\mu(x) = A \infty B \infty C = (A \infty B) \infty C = A \infty (B \infty C)$.

The same as the theorem 1 to prove. Corollary 2 by theorem 2 can extend pre-defined form of the attribute to finite element attributes, namely:

$$\mu(x) = A \infty B \infty C \ldots \infty N$$

Rule 5: In a AspectXML document, $e_1 \in \nu$, $e_2 \in \nu$, $(e_1,\gamma) \in R$, $(e_2,\gamma) \in R$ then element e_1 and element e_2 exist the same pre-defined attribute part denoted by $\mu(e_1) \approx \mu(e_2)$.

Rule 6: In a AspectXML document, $e_1 \in \nu$, $e_2 \in \nu$ assuming that exists $e_3, e_4, e_5, \ldots, e_n \in \nu$, $(e_1, \gamma) \in R \wedge (e_2, \gamma) \in R \wedge (e_3, \gamma) \in R \wedge (e_4, \gamma) \in R \wedge (e_5, \gamma) \in R \wedge \cdots \wedge (e_n, \gamma) \in R$ then $\mu(e_1) \approx \mu(e_n)$. Exist root boundary of two elements in AspectXML document, contain the same part of the pre-defined attributes. Because of the pre-definition has transitivity, then the attribute of the same root boundary element also has corresponding transitivity.

THREE-DIMENSIONAL XML CODING BASED ON CONE

The XML documents are usually represented by a tree structure for a tree of n layer, if makes the root node of the tree as the vertex of a cone, the nodes of each layer of the tree evenly occupy the bottom of different radius of the cone, the child nodes evenly cut up the sector region of the parent node occupies the bottom, resulting the XML document tree similar to a cone of the root node as the vertex, based on the thought, this study proposes a three-dimensional XML coding scheme based on cone.

Coding of the original document

Definition 7: The XML document node is a four-tuple (docID, H, rad, X). The docID represents the XML document number, which is used to distinguish the original document node and the weaving document node, the docID of the original document node and the weaving document node is different. The H represents the height of the node in the cone, also represents the layer of the node in the nodes tree at the same time, layer by layer increase from the vertex of the cone, the H of the root node is 0, the H of its child nodes is 1, layer by layer increase with 1 step size. The rad represents the radian of the node occupies the cone, any child node evenly cuts up the region occupied by the parent node, choosing the starting radian value to represent the node. The X represents the group marking of the node, which is used to represent brother nodes, brother nodes are in the same group and with the same group marking, marking are represented by letters, followed by a, b, c,..., z, aa, ba,..., za, ab, bb,..., zb,....

Figure 1 is an original XML document tree, Fig. 2 is the detailed scheme of the cone coding for Fig. 1 original XML document tree, among them the black node represents the root node, the red node represents the middle node, the green node represents the leaf node. Table 1 is the detailed cone coding, among them the docID of the original document node defaults to 1.

Coding of the weaving document: Usually, weaving the document has the following three methods:

Root node of the original document is woven: Figure 3 is the weaving operation for the root node of the original XML document in the Fig. 1, the root node of "people" is expanded, as shown by the red dashed line; after weaving, besides the structural relationship between the original document nodes need to judge may also need to judge the structural relationship between the node of the weaving documents and the root node of the original document, as shown by the blue dashed line. Figure 4 is the detailed coding scheme for the weave method 1 in Fig. 3, when the root node of the original document is the woven node, need to make the root node as the vertex of the cone again to form another cone upward according to the way of definition 7, among them the yellow node represents the root node of the weaving document, the blue node represents the node of the weaving document. Table 2 is the detailed cone coding of the weaving document.

Definition 8: The weaving document node is a four-tuple (docID, H, rad, X). The docID represents the XML document number, which is different from the docID of the original XML document. The H represents the height of the node in the cone, layer by layer decrease from the vertex of the cone, the H of the root node is 0 and because of the root node of the weaving document as its child node, then the H is -1, layer by layer decrease with 1 step size. The rad represents the radian of the node occupies the cone, any child node evenly cuts up the region occupied by the parent node, the root node of the weaving document is alone to occupy the region occupied by the root node of the original document, choosing the starting radian value to represent the node. The X represents the group marking of the node, which is used to represent brother nodes, brother nodes are in the same group and with the same group marking, marking are represented by letters, followed by a, b, c,..., z, aa, ba,..., za, ab, bb,..., zb,....

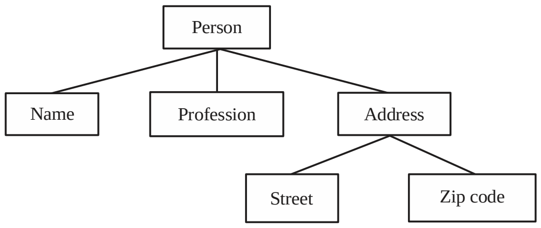

Fig. 1: Original XML document tree

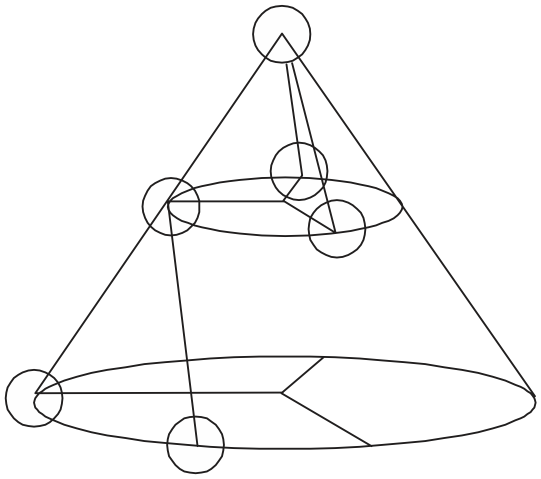
Fig. 2: Cone coding scheme

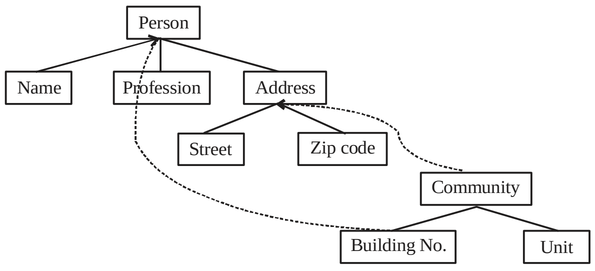

Fig. 3: Weave method 1

Table 1: Cone coding

Name of the node	Cone coding
Person	1, 0, 0, X
Name	1, 1, 0, a
Profession	1, 1, 120, a
Address	1, 1, 240, a
Street	1, 2, 240, a
Zip code	1, 2, 300, a

Table 2: Weave 1 cone coding

Name of the node	Cone coding
Contact	2, -1, 0, a
Phone	2, -2, 0, a
Email	2, -2, 180, a

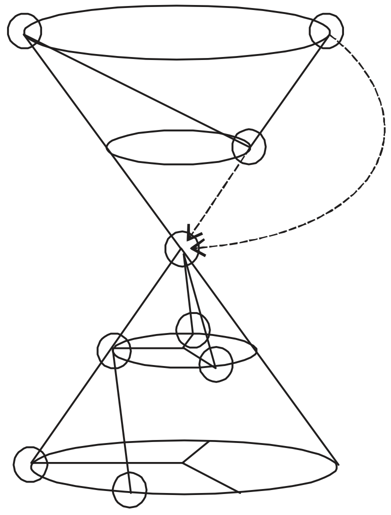

Fig. 4: Weave 1 coding scheme

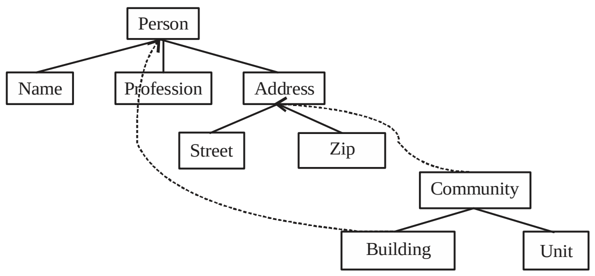

Fig. 5: Weave method 2

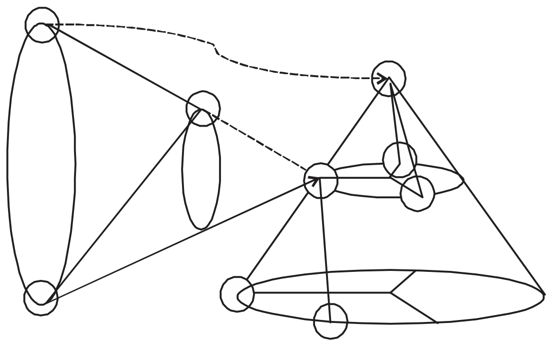

Fig. 6: Weave 2 coding scheme

Middle node of the original document is woven: Figure 5 is the weaving operation for the middle node of the original XML document in the Fig. 1, the middle node of "address" is expanded, as shown by the red dashed line; after weaving, besides the structural relationship between the original document nodes need to judge may also need to judge the structural relationship between the node of the weaving documents and the root node of the original document, as shown by the blue dashed line. Figure 6 is the detailed coding scheme for the weave method 2 in Fig. 5, when the middle node of the original document is the woven node, need to make the woven middle node as the vertex of the cone to form another cone outward according to the way of definition 7, among them the yellow node represents the root node of the weaving document, the blue node represents the node of the weaving document. Table 3 is the detailed cone coding of the weaving document and the independent coding of the woven middle node of the original document.

Table 3: Weave 2 cone coding

Name of the node	Cone coding
Address*	1, 1, 0, a
Community	2, 2, 0, a
Building No.	2, 3, 0, a
Unit	2, 3, 180, a

Definition 9: The weaving document node is a four-tuple (docID, H, rad, X). The docID represents the XML document number, which is different from the docID of the original XML document. The H represents the height of the node in the cone, layer by layer increase from the vertex of the cone, if the H of the woven middle node of the original document is h and because of the root node of the weaving document as its child node, then the H is h+1, layer by layer increase with 1 step size. The rad represents the radian of the node occupies the cone, any child node evenly cuts up the region occupied by the parent node, the root node of the weaving document is alone to occupy the region occupied by the woven middle node of the original document, the woven middle node of the original document need to be coded independently at this time, its rad is returned to 0, choosing the starting radian value to represent the node. The X represents the group marking of the node, the same as definition 7.

Leaf node of the original document is woven: Figure 7 is the weaving operation for the leaf node of the original XML document in the Fig. 1, the leaf node of "profession" is expanded, as shown by the red dashed line; after weaving, besides the structural relationship between the original document nodes need to judge may also need to judge the structural relationship between the node of the weaving documents and the root node of the original document, as shown by the blue dashed line. Figure 8 is the detailed coding scheme for the weave method 3 in Fig. 7, when the leaf node of the original document is the woven node, need to make the woven leaf node as the parent node for the root node of the weaving document and continue to form the next layer of the cone according to the way of definition 9, among them, the

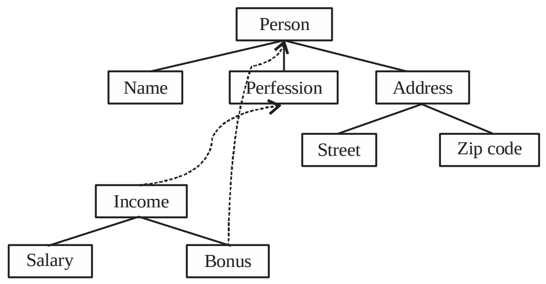

Fig. 7: Weave method 3

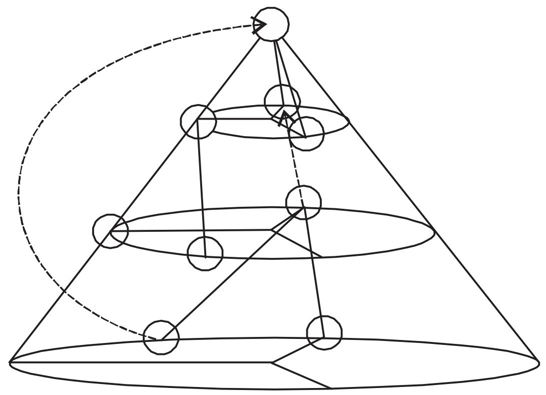

Fig. 8: Weave 3 coding scheme

Table 4: Weave 3 cone coding

Name of the node	Cone coding
Income	2, 2, 120, b
Salary	2, 3, 120, a
Bonus	2, 3, 180, a

yellow node represents the root node of the weaving document, the blue node represents the node of the weaving document. Table 4 is the detailed cone coding of the weaving document.

Definition 10: The weaving document node is a four-tuple (docID, H, rad, X). The docID represents the XML document number, which is different from the docID of the original XML document. The H represents the height of the node in the cone, layer by layer increase from the vertex of the cone, if the H of the woven leaf node of the original document is h and because of the root node of the weaving document as its child node, then the H is h+1, layer by layer increase with 1 step size. The rad represents the radian of the node occupies the cone, any child node evenly cuts up the region occupied by the parent node, the root node of the weaving document is alone to occupy the region occupied by the woven leaf node of the original document, choosing the starting radian value to represent the node. The X represents the group marking of the node, the same as definition 9.

IMPROVED ASPECT-ORIENTED XML STRUCTURAL JOIN ALGORITHM

On the basis of solving internal join in the document, realizing the structural join operation between the documents is the key of the aspect-oriented structural join algorithm. The elements of the ancestor nodes list AList and the descendant nodes list DList are divided into different subsets according to the document number docID, then the structural join operation is correspondingly converted into the join operation between the subsets.

Proposition 1: The nodes list List is ordered arrangement according to (docID, H, rad) assuming that exists any x of node, x∈List then all the descendant nodes of the node x in the List will be the n+1 segment continuous nodes immediately after the node x. Among them, n is the number of the weaving document for the node.

Prove the nodes list List is ascending order arrangement according to (docID, H, rad), if under the single document, namely the docID is unique, then all the descendant nodes of the node x in the List will be a segment continuous nodes immediately after the node x. When under the multi-document in the same way, the descendant nodes will be the n+1 segment continuous nodes.

Proposition 2: The nodes list List is ordered arrangement according to (docID, H, rad), assuming that exists any x of node, x∈List, then the first possible descendant node of the node x in the List is the first node of the first segment of the n+1 segment continuous nodes. Prove by proposition 1 shows, all the descendant nodes of the node x in the List will be the n+1 segment continuous nodes. Each segment nodes is ascending order arrangement according to (docID, H, rad) known that the node x is the right node of the first segment nodes.

The improved ancestor/descendant Queue-tree algorithm is the merging join algorithm based on the queue. On the basis of the original Queue-tree algorithm that has realized the structural join between the nodes under the single document, the improved Queue-tree algorithm will realize the structural join operation under the multi-document, called the Ext-Queue-tree algorithm. The algorithm based on index technology to realize the operation that skipping the nodes don't participate in the join in the process of structural join, which improves the query efficiency. By the index to skip the ancestor nodes don't participate in the join and access the node of the weaving document be shown in Fig. 9.

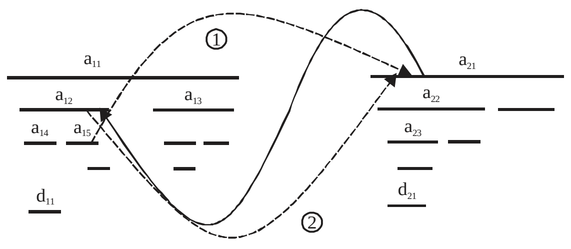

Fig. 9: Schematic skipping ancestor node to access the weaving document node with index

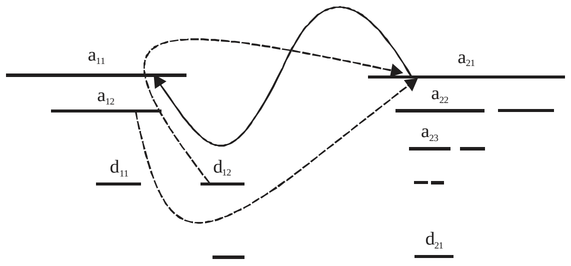

Fig. 10: Schematic skipping descendant node to access the weaving document node with index

In Fig. 9, the thick line represents the region rad occupied by the node of the ancestors list AList; the fine line represents the region rad occupied by the node of the descendants list DList; the arrowed solid line represents the weaving information, the head of the arrow points to the woven node, the tail of the arrow points to the weaving node. The dotted line ① represents the skip ability of the node and the dotted line ② represents the mark of the node localization by the index. The node a_{1i} represents the node of the ancestor nodes list in document 1, the node d_{1i} represents the node of the descendant nodes list in document 1; the node a_{2i} represents the node of the ancestor nodes list in document 2, the node d_{2i} represents the node of the descendant nodes list in document 2.

The Ext-Queue-tree algorithm uses the thought of the Anc-Desc-B+ algorithm to realize that skipping the ancestor nodes don't participate in the join, realized skipping the nodes don't participate in the join between the multi-document. The process by the index of the Anc-Desc-B+ algorithm is making all the possible descendant nodes $\{d_i\}$ in the DList list of the current node r of the AList list into the queue and output all the join results and making the other nodes of r's nodes subset to match ancestor/descendant relationship successively in the queue DQueue, output the join result until the node is the ancestor node of d_{12}, cycle ends when the AList list and the DList list is empty.

Figure 9 shows the algorithm how to use the index to realize that skipping the nodes don't participate in the join between the multi-documents. The Ext-Queue-tree algorithm uses the thought of the Queue-Tree algorithm, realized the operation of the structural join between the nodes under the single document. Because the parent/child relationship join between the node of the original document and the node of the weaving document could only happen in the root node of the weaving document, so if realizes the parent/child relationship join, only need to delete the last line code of the Ext-Queue-tree algorithm.

Besides, also can skipping the descendant nodes don't participate in the join. By the index to skip the descendant nodes don't participate in the join and access the node of the weaving document be shown in Fig. 10. Figure contains some descendant nodes don't participate in the join in the DList list, a_{21} as the root node of the weaving document is woven into a_{11} of the original document, forming a new child node, using the index to realize that skipping the nodes don't participate in the join between the multi-document.

When the Ext2-Queue-tree algorithm scans the DList list, using the B+ tree index directly locating to the root node a_{21} of the weaving document. When processing the structural join of the parent/child relationship, only the root node of the weaving document participated in the join, therefore, only need to delete the last line code of the Ext2-Queue-tree algorithm.

RESULTS

In order to test the effectiveness, rationality and practicability of the coding scheme and structural join algorithm of this study, the corresponding experiment is designed. Comparing the Cone Coding Scheme (CCS) of this study with CDBS and QED of region coding (Fig. 11a). Under the condition of different proportion of the number of the weaving nodes, the performance test of the Ext-Queue-tree algorithm (EQT) and Ext2-Queue-tree algorithm (E2QT) is conducted (Fig. 11b). In this study, the experimental platform is 2.53 GHz Intel core dual-core processor, memory is 4 GB, operating system is Windows 7, using the Visual c++ 6.0, based on the DOM programming technology to realize. The chosen test datasets is generated by XMark of the XML automatic generation tool, the information such as the depth and number of nodes of the document tree, as shown in Table 5.

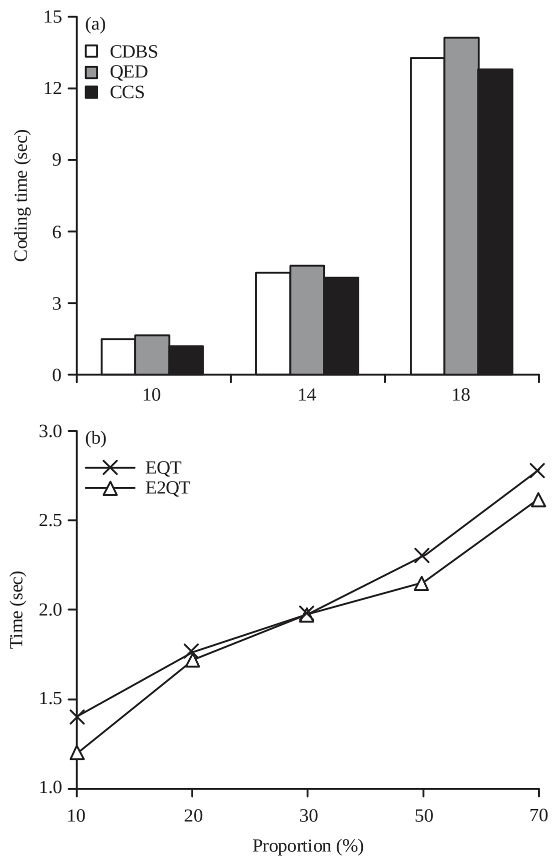

Fig. 11(a-b): Performance study on dynamic coding comparison of coding (a) Bit length and (b) Time

Table 5: XML test datasets

Dataset	Total number of nodes	Maximum depth	Average depth
Xmark_1	475844	8	3.12
Xmark_2	1034867	4	2.96
Xmark_3	2254776	32	7.68

STRUCTURAL JOIN PERFORMANCE ANALYSIS

Testing the query time of the algorithm under the condition of the weaving proportion of 10, 20, 30, 50 and 70% respectively. This study chooses two test query examples, which is the person/name and address/unit respectively, testing the query performance of the two algorithms under the different datasets. Judging from the results, when the XML documents have the same number of nodes with the increasing number of the weaving document, the query time of the two algorithms also will increase, the efficiency decline. The experimental results show that, compared with the existing structural join algorithm, the improved structural join algorithm has a significant improvement on the access time and the times of access nodes, the result is shown in Fig. 12.

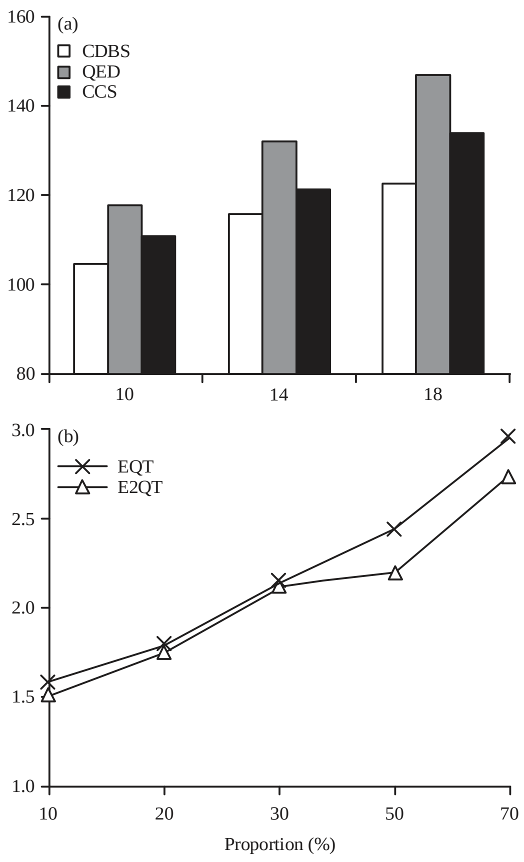

Fig. 12(a-b): Performance study on join algorithm with different weave proportion, (a) Element person/element name and (b) Element address/element unit

DISCUSSION

In the static performance analysis[5-7], the coding of QED based on quaternary and the "0" of quaternary only be used for the coding end mark and the "0" does not appear in the effective bit, therefore, the coding storage proportion is lower. While the fixed length storage efficiency of CCS is higher than the variable length storage efficiency of CDBS, therefore, it can be seen from the figure, compared with CDBS, the coding bit length of CCS is shorter. The QED and CDBS need to recursively generate the sequence of the code value, then use the generated sequence of the code value to encode, while CCS traverses only once mark is simple, therefore, it can be seen from the figure, CCS has the best time performance.

In the dynamic performance analysis[5-7], the coding of CDBS based on binary, the storage proportion is higher, while the effective bit of QED coding only have the "1", "2", "3" of quaternary, the storage proportion is lower, it can be seen from the figure, the bit length of CDBS is slowest-growing, while QED is fastest-growing, then CCS is between CDBS and QED. When calculating the coding of the new node, CDBS and QED need to rescan the existing coding, compared with QED, the length of CDBS coding is shorter, so CDBS costs less time, while the time cost of CCS is least, this is mainly due to CCS don't need to rescan the existing coding, saved the time. When calculating the coding of the new node, CDBS and QED need to rescan the existing coding, compared with QED, the length of CDBS coding is shorter, so CDBS costs less time, while the time cost of CCS is least, this is mainly due to CCS don't need to rescan the existing coding, saved the time.

CONCLUSION

This study studies the query technology of the aspect-oriented XML document and its realization. First of all on the basis of the original XML document definitions, introducing the thought of aspect-oriented, realizing the definition of elements and attributes of the XML document, discussed the expanded document definition model, proposing the related model definition and using mathematical methods to describe the elements and attributes. The aspect-oriented XML Schema improves the modularity and portability of the document definitions, andenhances the reusability of the coding. Secondly, this study proposes a new efficient XML tree dynamic coding scheme the three-dimensional XML coding based on cone, which has a good static performance, at the same time, supports the document update and update efficiency is higher. Finally, this study improves the existing structural join algorithm, realizes the structural join algorithm under the multi-document and outputs the judgment of related relationships of the node. Through theoretical analysis and related experiments show that in this study, the aspect-oriented document model and definition, the coding scheme and the corresponding structural join algorithm is effective and feasible.

ACKNOWLEDGMENT

We gratefully acknowledge the valuable cooperation of my graduate students and the members of our laboratory.

REFERENCES

1. Yao, B., C. Ma, X. Na, X. Qi and Y. Guo, 2014. Dynamic prefix XML encoding scheme based on fraction. J. Shangqiu Normal Univ., 30: 71-74.
2. An, D. and S. Park, 2011. Efficient access control labeling scheme for secure XML query processing. Comput. Standards Interfaces, 33: 439-447.
3. Lu, J., X. Meng and T.W. Ling, 2011. Indexing and querying XML using extended Dewey labeling scheme. Data Knowl. Eng., 70: 35-59.
4. Guo, L.H., J. Wang and H. Du, 2013. An XML encoding scheme based on concentric circular cutting. Comput. Eng., 39: 52-54.
5. Zhang, C., J. Naughton, D. DeWitt, Q. Luo and G. Lohman, 2001. On supporting containment queries in relational database management systems. Proceedings of the 2001 ACM SIGMOD International Conference on Management of Data, May 21-24, 2001, California, USA., pp: 425-436.
6. Jagadish, H.V., S. Al-Khalifa, C. Adriane, V. Laks and S. Lakshmanan *et al*., 2002. TIMBER: A native XML database. Int. J. Very Large Data Bases, 11: 274-291.
7. Chien, S.Y., Z. Vagena, D. Zhang and V.J. Tsotras, 2002. Efficient structural joins on indexed XML documents. Proceedings of the 28th International Conference on Very Large Data Bases, August 20-23, 2002, Hong Kong, China, pp: 263-274.
8. Kong, L.B., S.W. Tang, D.Q. Yang, T.J. Wang and J. Gao, 2007. Querying techniques for XML data. J. Software, 18: 1400-1418.
9. Huang, Y. and W.W. Yang, 2007. Structural join algorithm in XML query. Comput. Aided Eng., 16: 74-75.
10. Changxuan, W. and L. Xiping, 2008. XML Database Technology. Tsinghua University Press, Beijing.
11. Zhu, X.T. and B. Xu, 2011. A fuzzy association rules algorithm for XML document. Sci. Technol. Eng., 26: 6467-5470.