論文 / 著書情報
Article / Book Information

論題(和文)	Valable LCA探索のためのサイズを考慮した索引構成
Title(English)	Index Constructions for Valuable LCA Detection Considering The Size
著者(和文)	小林 径, 横田 治夫
Authors(English)	Kei Kobayashi, Haruo Yokota
出典(和文)	第73回情報処理学会全国大会，，， N4-4
Citation(English)	The 73rd National Convention of IPSJ，，， N4-4
発行日 / Pub. date	2011，3
権利情報 / Copyright	

Valable LCA 探索のためのサイズを考慮した索引構成

小林 径 † 横田 治夫 †

† 東京工業大学大学院情報理工学研究科計算工学専攻

1 はじめに

近年，XML(Extensible Markup Language) 文書の増加に伴い，XML 文書から必要な部分のみを検索したい，というユーザの要求がある．これに対し，我々は使い易さの観点から XML 文書に対しキーワード検索によって必要な部分を取得する手法に着目した．

XML 文書に対するキーワード検索では，XML 木構造中の各キーワードを含むノードの最も近い共通祖先である LCA(Lowest Common Ancestor) を根とする部分木を抽出する手法，およびそれに対する改良提案の 1 つである VLCA(Valuable LCA) がある [2]．VLCA は，精度で LCA より優れることを示しているが，その VLCA を抽出する手法は，高頻出語での検索性能が十分でないという問題がある．

この問題の解決のために，我々はこれまでに，効率的に VLCA を抽出するための索引構成手法を提案し，既存手法 VLCAStack との比較実験によってその効果を示した．また，その際に作成する索引データ量が大きくなる課題があるため，その削減手法を提案した．

しかし，検索語数 3 以上の場合の適用に課題がある．本稿では，検索語数 3 以上の場合にも効率的に検索を行うため，検索語数の増加に対し枝刈りを行うことで検索時間の増加を軽減するための手法を示す．

2 VLCA 抽出のための索引構成

2.1 VLCA とは

LCA とは各語を含むノードの最低の共通祖先のノードである．VLCA とは，2 つのノード u,v とそれらの LCA ノードを w とした時に，w-u 間，w-v 間に出現するノードの中に，同名タグを持つノードが存在しない LCA である．図 1 の文書に語”XML”,”John”という語を与えた場合の LCA はラベル 0,0.1,0.0.1 の 3 つのノードであり，VLCA はラベル 0.0.1 のノードである．逆に 0.1 については 0.1.1,0.1.2 にタグ名 paper が重複出現したため VLCA でないことがわかる．

2.2 BitMapKBSI 手法の概観

VLCA を求めるためには，表 1 で示す 3 つの手順を経る．本手法では，表で示す通り，手順 1 で各検索語

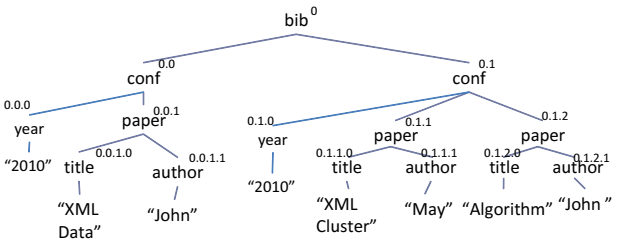

図 1: XML 文書の例

表 1: VLCA 抽出手順と提案手法が用いる索引

1. キーワードを含むノード取得	B+tree 索引
2.LCA の特定	シグネチャ索引
3.LCA が VLCA か判定	タグ名のビット列

のエントリに含まれる，検索語を含むノードを取得する．次に，2 番目の手順でノードに対応するシグネチャ索引を用い，3 番目の手順で，ノードタグ名に対応するビット列の索引を用いることで処理の効率化を行う．

2.3 BitMapKBSI 手法の索引構成

図 2 は，BitMapKBSI 手法の索引構成である．各キーワードのエントリの下に，その語を含むノードラベルの値と，そのノードに対応するノードシグネチャの値，およびその語が出現するノードから索引付けされるまでに出現したタグ名のビット列の論理和の値を B+tree 索引の対応するエントリに格納する．

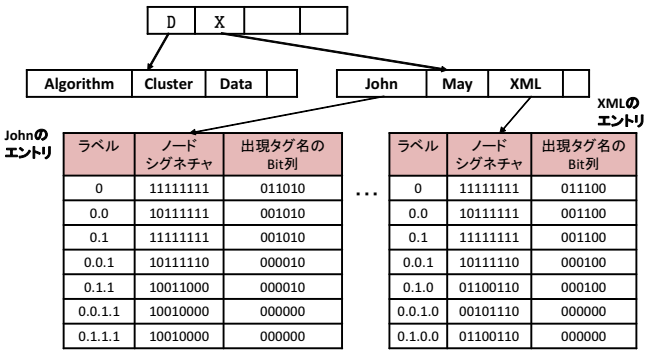

図 2: BitMapKBSI 手法の索引構成例

Index Constructions for Valuable LCA Detection Considering The Size

Kei KOBAYASHI† and Haruo YOKOTA†

†Department of Computer Science, Graduate School of Information Science and Engineering, Tokyo Institute of Technology

kei@de.cs.titech.ac.jp ,yokota@cs.titech.ac.jp

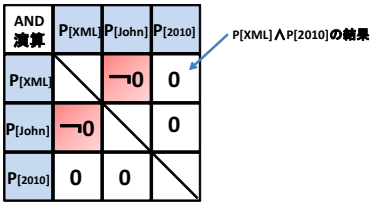

図 3: 条件 1 を満たすか否かの検査用マトリクス

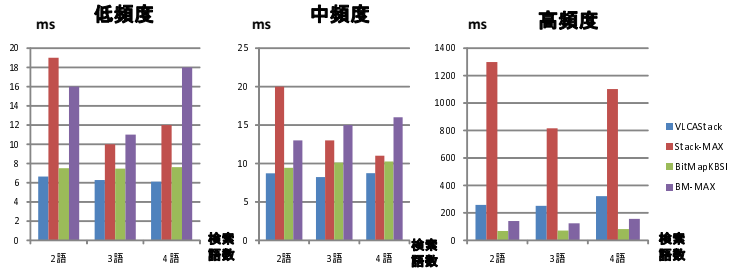

図 4: 文書 dictionary での検索時間比較

2.4 構成した索引による VLCA 探索手順

構成した索引を用いた VLCA 抽出手順は以下である．

(1) ノード シグネチャの値 NS と検索語のシグネチャ論理和 Q について，式 $Q = Q \wedge NS$ を満たす行のノード ラベルを抽出する．全検索語のエントリに出現したラベルが LCA である．

(2)(1) で得た LCA ノード ラベルについて，同ラベル同士で「タグ名のビット列」項の値の AND 演算を行い，結果が 0 となるノード ラベルを求める．

3 検索語数の増加に対する枝刈り手法の提案

2.4 で述べた検索方法では，検索語数 3 以上の場合，タグの重複出現と，VLCA 判定を行うノードへ至る手前でのパスの合流による同一ノードを含む場合を区別できない．これらを区別する処理を追加し，全ての場合について行うと性能が著しく劣化してしまう課題がある．そこで，m 語の VLCA が満たす条件に着目し，それを満たす場合のみ上記の区別する処理を行うことで，全場合に対し処理を行うより効率化できる．

m ノードの VLCA が満たす条件とは以下である．

「ある m 個のノード N_1,$\cdots N_m$ と，それらの LCA ノード w があるとする．w が VLCA であるならば，全てのノード n_i について，ノード w と 2 ノードの VLCA を構成するあるノード n_j が 1 つは存在する．」

これを用いて，VLCA 探索手法を次のように変更する．ただし，表 1 で示す手順 1,2 については変更がない．

(2) 手順 3 において，手順 2 で求めた LCA ノード w に対して，m 語のエントリの w のラベルに対応する m 個のタグ名のビット列 P[k_1],P[k_2],$\cdots$P[k_m] を得る．全ての P[k_i]，P[k_j] の組に対して P[k_i] $\wedge$ P[k_j] を計算する．その操作は，図 3 のようなマトリクスに対し，クロスする P[kw] の値どうしの AND 演算を行い，その結果が 0 か非 0 かを順次記入する操作に相当する．

(3) マトリクスを全て埋めた後，次の判断を行う．

1. 結果が全て 0 ならば，w は VLCA である．

2. 全ての横行について，1 つでも 0 が存在するかを調べる．それを満たす場合のみ，結果が $\neq$0 であった組に対して区別処理を行う．

4 比較実験

3 節で述べた枝刈り手法および VLCAStack を実装し性能を比較し有用性を検証する．なお，今回作成した索引は [3] で提案したデータ削減法を適用した．

実験で使用した計算機環境は [3] と同等である．使用文書は XBench[1] の生成器から生成した文書 dictionary(10.6MB) であり，シグネチャ長は 512bit とした．また，出現する語をその出現回数に応じ低頻度 (1-50)，中頻度 (51-500 回)，高頻度 (501 以上) の 3 つに分類した．検索時間は各クラスに属す検索語を 2〜4 語与えた検索を 100 回行い，その平均値と最大値を比較した．

図 4 は，性能比較実験の結果である．なお図中のデータラベル名 VLCAStack,BitMapKBSI は各手法の平均値であり，Stack-MAX,BM-MAX は VLCAStack,BitMapKBSI の最大値である．

今回の実験では，全文書で平均値では低中頻度時は同程度，高頻度時に平均値，最大値とも VLCAStack に大きく勝る結果を示した．また検索語を増加しても，検索時間の増加比率は高々1.1 倍であった．

5 まとめと今後の課題

本稿では，検索語数の増加に対する対策として，枝刈り手法を提案した．実験において，VLCAStack に対し低・中頻度で同程度，高頻度時に勝ること，また語の増加に対する検索時間の増加率も低い値を示した．

今後の課題として，様々な文書で実験を行い影響も観測したい．

謝辞

本研究の一部は，日本学術振興会科学研究費補助金基盤研究 (A)(#22240005) および文部科学省科学研究費補助金特定領域研究 (#21013017) の助成により行われた．

参考文献

[1] Xbench - a family of benchmarks for xml dbmss. http://se.uwaterloo.ca/ ddbms/projects/xbench/.

[2] G. Li, J. Feng, J. Wang, and Lizhu Zhou. Effective keyword search for valuable lcas over xml documents. In *KICM*, pp. 31–40, 2007.

[3] 小林径, 横田治夫. 索引サイズと検索語数を考慮した VLCA 用の索引構成と検索処理. In *DEIM Forum 2011*, 2011.