様　式　Ｃ－１９、Ｆ－１９－１、Ｚ－１９　（共通）

科学研究費助成事業　　研究成果報告書

平成 ２９ 年　 ６ 月 １９ 日現在

機関番号：１３４０１
研究種目：基盤研究(C)（一般）
研究期間：2014～2016
課題番号：２６３３０１３２

研究課題名（和文）大規模動的木構造グラフの新しい実装方式とそのXMLデータベースへの応用

研究課題名（英文）An Implementation Scheme for Large Scale Dynamic Tree Graphs and Its application to XML Databases

研究代表者
都司　達夫（Tsuji, Tatsuo）

福井大学・学術研究院工学系部門・シニアフェロー

研究者番号：８０１１５３０２

交付決定額（研究期間全体）：（直接経費）　　3,600,000円

研究成果の概要（和文）：我々は，動的に増大する多次元データを効率良くエンコードする経歴・パターン法と呼ぶ方式を提案している．本研究では，このエンコード法に基づいて，大規模な動的木グラフの実現方式を提案すると共に，木グラフで表現されるXML文書の格納・検索システムのプロトタイプを作成し評価した。
具体的には、(1)ラベル付き木グラフを経歴パターン法で実現する時の問題点を明らかにした．(2)問題点解決のためのグラフ分割方式を提案・実装した.(3)大規模木グラフの分散格納と統合を支援するために，複数の木グラフの埋め込み方式を設計した(4)大規模分散XML文書処理への適用を目的として，(3)における設計を分散対応とした。

研究成果の概要（英文）：History-pattern encoding we are proposing is an efficient encoding scheme for dynamically increasing multidimensional tuple datasets. In this research, based on the encoding scheme, (i) a new implementation scheme was presented for dynamically growing large scale tree graph, (ii) a prototype system for storing and retrieving tree structures underlying in XML documents was constructed and evaluated. The following four achievements were accomplished: (1) The problems arising in encoding the labeled tree graphs were clarified in using the history-pattern encoding scheme. (2) To resolve these problems, a tree graph partitioning scheme was presented and implemented. (3) To support the distribution and integration of the large scale tree graphs, an embedding scheme of the guest tree graphs in a single host tree graph was designed. (4) In order to apply the embedding scheme in (3) to the distributed XML documents processing system, the scheme was adapted to the distributed environment.

研究分野：データ工学

キーワード：動的木グラフ　大規模木グラフ　木グラフエンコード　経歴・パターン法　木グラフ埋め込み　分散XML文書処理

１．研究開始当初の背景

近年，情報構造の複雑化に伴って，データ相互の関係を直接表現し，操作できるグラフデータモデルの有用性が注目され，様々なモデルが開発されている。中でも，木構造グラフ(以下，単に木グラフ)は重要なグラフ構造であり，有用性が高い。論理構造としての木グラフの表現力と操作性は広く認識されているが，木グラフの計算機内部での実現構造とその操作技法の開発には課題が少なくない。特にシステム運用時のデータ規模の動的な増大と構造の変化に効率良く対処することができる大規模木グラフの新たな実現技術の開発は特に構造を持つビッグデータを扱う上述のような応用分野において強く求められている。

我々はこれまでの研究において，タプル集合としての多次元データを効率良くエンコードする経歴・パターン法と呼ぶエンコード方式を提案し（引用文献①），種々の応用分野で使用するための多次元 DB 基盤システムを構築している。経歴・パターン法には以下の利点がある。

(1) （低記憶コスト） 多次元データの次元数に関わらず，＜拡張経歴値，座標パターン＞の２値でタプルをエンコードできる。

(2) （高速検索） 各次元の配列添字のシフトと論理積/論理和のレジスタ命令のみで高速にタプルを２値にエンコード可能である。

(3) （動的拡大） タプルの新たな属性値*v*の出現に対して，拡張可能配列における既存タプル集合の再エンコードは不要である。

本研究では，以上のような従前の研究成果に基づいて，大規模な動的木グラフの新しい実現方式を提案し，実装・評価する。

２．研究の目的

本研究では，１で述べた多次元データの新しいエンコード方式である経歴・パターン法を応用して，この分野の数ある研究には見られないアプローチにより，大規模なラベル付き動的木グラフ構造のためのエンコード方式と実装方式を提案し，その処理基盤システムを構築することを目的としている。動的な木グラフ構造の処理は経歴・パターン法の長所を最もよく活かし得る分野であると考えられ，ここでは特に XML データベースへの応用を試みる。

３．研究の方法

(1) 本研究において対象とするラベル付き木グラフモデルの経歴・パターン法に基づいた記憶方式を設計する。

(2) 構造木を分割する方式とそれを実現するための経歴パターン法の活用方式を検討する。

(3) 大規模木グラフの分散格納と統合を支援するために，複数の木グラフの埋め込み操作を設計する。

(4) 大規模 XML 文書の処理基盤への適用を目的として，(3)における設計を分散対応とする。

４．研究成果

［経歴・パターン法の概要］

経歴・パターン法は拡張可能な論理配列空間における座標のエンコード方式であり，任意の座標を経歴値とパターンの組で表現することができる(図1)。論理配列空間は必要に応じて，現在の配列サイズと同一の新たな論理部分配列を付加することにより拡張される。

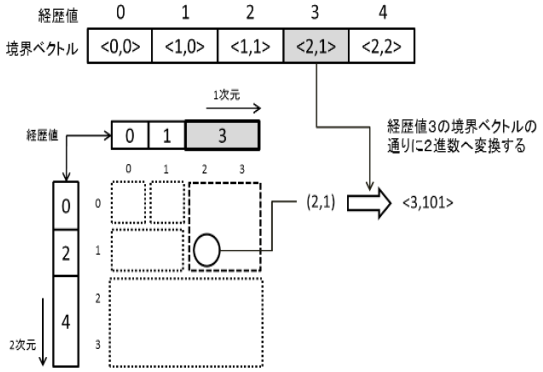

図1 経歴・パターン法によるエンコードの例

部分配列を拡張・付加した順序を経歴値と呼び，これは，部分配列を同定する。拡張の履歴は一次元の境界ベクトルテーブルとして記憶される。このテーブルの配列添字は部分配列の経歴値であり，境界ベクトルは各次元の添字を何ビットで表現するかを表す。また，論理配列の各次元の添字を境界ベクトルの各次元のビットサイズにしたがって，ビット連結したものを座標パターンと呼ぶ。HPMD は経歴・パターン法の一実装方式である。図 1 にエンコード例を示す。

(1)　ラベル付き木グラフの格納方式

ここでは木グラフモデルとして，XML木をモデルとした。すなわち，子節点への枝は重複を許してラベル付けされる(図2)。

木グラフは，木の高さレベルをHPMDの次元に対応させ，上記の拡張可能配列にマッピングすることにより，経歴・パターン法でエンコードする。節点に対応する次元までの各次元添字は子節点に引き継がれる。ラベル付き木グラフは既存方式(DataGuide)を採用し，ラベルの連接である経路式を扱う経路木と，ノードの兄弟順を表す順序数を付けて木グラフの構造を管理する構造木の二つの木グラフに分離する(図2)。

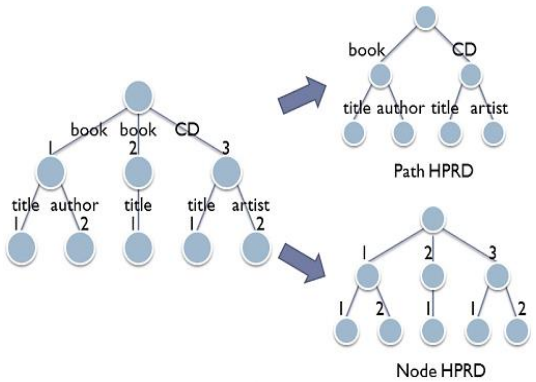

図 2　ラベル付き木構造の分離

図2の2つの木グラフのうち構造木の格納について，次の２つの問題点が指摘できる。
① 兄弟ノードの祖先パターンは同一であり兄弟ノードのエンコードにおいて重複して格納される。
② 木グラフの最大レベルより低いレベル位置においてノードの追加が行われると，追加したノードのレベルに対応する次元より後の次元の添字はすべて0となる。例えば，拡張可能配列の次元数が6であるときに，レベル2の位置にノードの追加が行われると，そのノードの座標は(1, 2, 0, 0, 0, 0)のように次元3からの添字はすべて0となる。これは経歴パターン法による記憶コストの劣化につながる。

(2) 構造木の分割
構造木に関する(1)の2つ問題点①②にグラフの水平・垂直分割により対処した(図3)。
構造木Tは複数の部分木T_1, T_2, …, T_nに分割される。これに伴い，TのルートノードからT_iのノードにいたるTの経路式Pは2つに分割される。ここでは，この分割された経路を経歴・パターン法により効率良くエンコード／デコードするための方式を提案した。さらに，T_iのルートノードに至る経路のエンコードをT_i にアクセスするための索引として機能させている。T_iの各ノードのエンコードにおいて，この経路を共有させることにより，①の問題点に対処している。また，索引による経路式検索や軸検索などTの構造検索の速度向上を図っている。

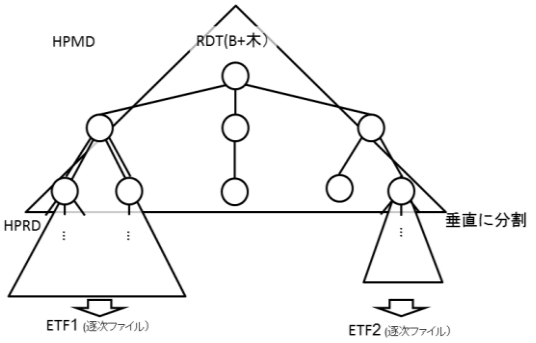

図 3 構造木の水平・垂直分割

上述の構造木の分割に対応して，任意のノードの経路式をエンコードした<経歴値，パターン>において，パターンは共有部分を表す共有パターンと分割点となるノード以降のローカルパターンに２つに分割される共有パターンは図3のRDTと呼ぶ索引（B+木）に格納され，ローカルパターンは分割により生成されるETFと呼ぶ逐次ファイルとして実装される部分木に格納される（図4)。

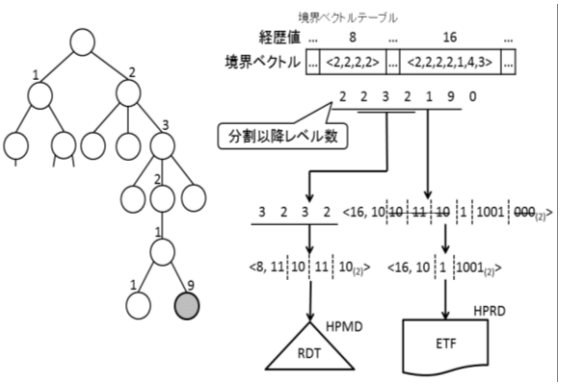

図 4 経路式の分割と格納

共有パスの先頭にはメタ情報として，分割レベル以降のローカルパスのパス長が付与され，共有パターンにエンコードされる。

(3) 複数の木グラフの埋め込み操作の設計
前述の(1)(2)の研究は大規模な論理木グラフを分割により効率よく実現するための研究であるが，ここでは，経歴・パターン法により，エンコードされている木グラフ(以下ホスト木という) に, ネットワーク上の他の木グラフ(以下ゲスト木という)を埋め込み, 効率よく操作する方式を設計・実現する後，大規模・分散木構造グラフを実現することを目標とする。本方式により，例えば，XML文書における名前空間のように独立して機能する木グラフをゲスト木として，埋め込み，論理的に単一のグラフとして統合して操作することが可能である。また，木グラフの実装にHPMDを使用することにより，ホスト木とゲスト木の集合を内部的に効率良く管理することができる。基本方針として，ホスト木にもゲスト木にも埋め込み情報を保持する必要がない方式とした。
（論理ノード）埋め込み操作を実現するため，論理ノードの概念を導入する。ゲスト木が埋め込まれるホスト木における位置には，論理ノードを埋め込む（図5)。図5において，TIDは木グラフのIDであり，LnIDは論理ノードのIDである。そして，その位置に埋め込まれるゲスト木のルートノードの情報を外部のデータ構造で管理する（図6)。これにより，埋め込みによる既存の構造の修正を最小限にして，ホスト木とゲスト木の論理的な関連付けを実現する。

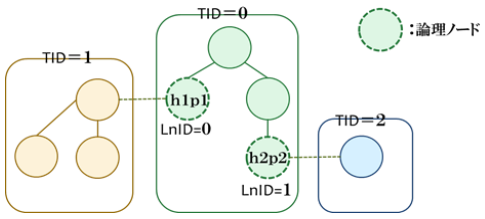

図 5 論理ノードへの埋め込み

埋め込み操作は複数の木に及ぶため，各木にはID(Tree ID:TID)を，論理ノードにもID(Logical node ID:LnID)を付加し判別を行う。

（論理ノードの管理）論理ノードによるホスト木とゲスト木の関連付けを管理するデータ構造として，ホスト木テーブル(Host_Table)とゲスト木テーブル(Guest_Table)を導入している。図6は図5に示した埋め込み情報を管理するホスト木テーブルおよびゲスト木テーブルである。Host_Tableの添字はTIDに対応しており，添字で識別される木のルートノードがホスト木のどの論理ノードに対応しているかを示している。h_TIDは論理ノードが属するホスト木のTID，h_hpはホスト木における論理ノードの経歴値・パターンを表している。また，Guest_tableは当該ホスト木に埋め込まれているゲスト木集合のTIDを格納しているGuestTableのアドレスである。 GuestTableのg_TIDはHostTableの添字に対応しており，ホスト木におけるゲスト木の埋め込み情報を参照するのに使われる。

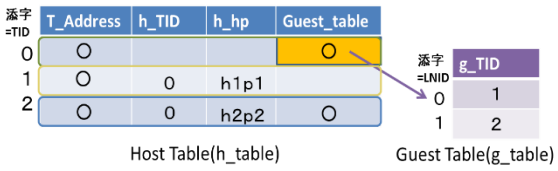

図 6 論理ノードによる図5の関連付けの管理

（論理ノードの検索）ホスト木およびゲスト木のノードのエンコード結果はそれぞれ別個のETFファイルとして格納される。基本方針として，ホスト木にもゲスト木にも埋め込み情報（論理ノード情報）を保持する必要がない方式とするために，ホスト木の検索に際して，論理ノードの存在を確認する必要がある。検索パス式の各ステップにおいて当該ノードが見つからない場合にHost Tableをサーチして論理ノードの検索を行う。このサーチ時間は本方式の検索オーバヘッドとなり得る。

(4) 大規模XML文書の処理基盤への適用

ネットワーク上の関連サイトで開発されている木グラフデータをホスト木，ゲスト木として使用することにより論理的に単一の大規模木グラフを構成することを前提として，論理ノードに関する上記方式の分散対応を行った。ネットワーク上のメッセージ通信にはMPI(Message Passing Interface)を用いた。

XML文書には”名前空間”の概念があり，ネットワーク上に分散したXML文書を参照する機能があるが，検索の際には，常にこの名前空間名をプレフィクスする必要があるのに対して，本方式ではこのような必要はなく，よりよく分散透過性を達成している。以下では，実装したプロトタイプシステムについて，計測結果を示す。測定環境は以下に示す。

(使用マシン)

① CPU：Xeon X5690(3.47GHz)，メモリ：49GB
② CPU：Xeon E5-2609(2.50GHz)，メモリ：32GB
③ CPU：AMD Opteron Processor 6172(2.1GHz)，メモリ：64GB.

(使用データ) XmarkのXML生成プログラムxmlgenにより生成したものから，属性値やテキストなどを取り除き，<要素名>と</要素名>のみで再構成したデータセット(以下，DS)。

・ファイルサイズ 約3.33GB
・ノード数 167,095,844
・最大レベル 12

(埋め込み木の生成) DSの木グラフからランダムに0, 10, 50, 100個選定したノードをルートとする部分木をゲスト木として生成。

(絶対経路式検索) DS からノードをランダムに選定し，ルートノードからの絶対経路式の検索時間の平均をマシン①により，測定した（図 7)。ゲスト木の数（論理ノード数）にかかわらず，検索時間はほぼ一定であることがわかる。この結果から，論理ノードの導入によるオーバヘッドはほとんど無視できることが確認できた。

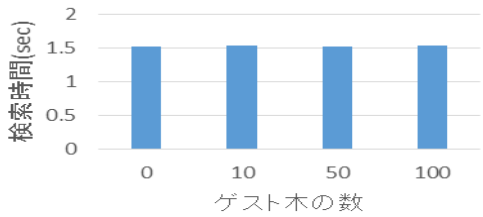

図 7 経路式の検索時間

(分散環境下における経路式検索) 上述の２つのマシン②③にゲスト木集合を均等に配置して絶対経路式の検索時間の平均を測定した(図8)。ゲスト木の数にはほとんど影響されないが，分散配置せずに単一マシンに配置した場合に比べて1.8%程度増加していることが確認できた。

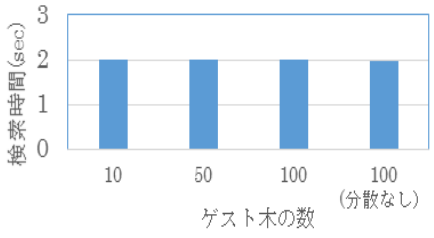

図 8 分散環境下での経路式検索時間

(分散環境下における構造検索)

親検索での検索時間の差が大きい(図9)。全ての処理を高性能マシン②で行う非分散環境下に比べて，分散環境下での親検索は，主な処理を性能の低いマシン③で行っていることが一因であると思われる。子検索の場合は主な処理をが高性能マシン②で行っているため，分散環境下，非分散環境下どちらの場合でも，検索結果に大きな違いは見られない。

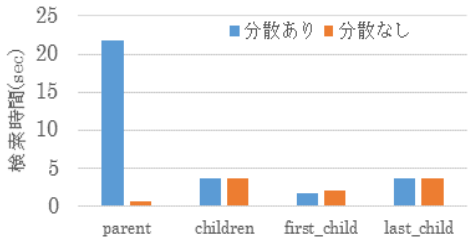

図 9 親子検索時間

＜引用文献〉

① Makino M., Tsuji T., Higuchi K. : History-Pattern Encoding for Large-Scale Dynamic Multidimensional Datasets and Its Evaluations, IEICE Transactions 99-D(4), 989-999, 2016. 査読有

５．主な発表論文等

〔雑誌論文〕（計 2 件）

① Makino M., Tsuji T., Higuchi K. : History-Pattern Encoding for Large-Scale Dynamic Multidimensional Datasets and Its Evaluations, IEICE Transactions 99-D(4), 989-999, 2016. 査読有

② 吉田惇人, 樋口健, 都司達夫: 線形ハッシュ法と Skip Graph を用いた分散キーバリューストアの実装と評価，電子情報通信学会論文誌 D, Vol. J98-D No. 5 pp. 742-750, 2015. 査読有

〔学会発表〕（計 14 件）

① Sudoh A., Tsuji T., Higuchi K.: A Partitioning Scheme for Big Dynamic Trees, Proc.of DASFAA Workshops 2017,18-34, 2017. 査読有

② Chiba Y., Kitajima S., Tsuji T., Higuchi K.: A scheme of dynamic attributes addition for tuple datasets. Proc. of iiWAS 2016: 288-292, 2016. 査読有

③ 千葉陽介，都司達夫，樋口健：属性の動的追加を考慮したデータキューブの差分構築の一方式，第 15 回情報科学技術フォーラム講演論文集，D020, 79-80, 2016. 査読無

④ 菅安唯伽，都司達夫，樋口健：タプルデータセットの一エンコード方式，第 15 回情報科学技術フォーラム講演論文集，D020,8 1-82, 2016. 査読無

⑤ 鈴木雄也，都司達夫，樋口 健：木構造の埋め込み操作の実装と性能評価，平成 28 年度電気関係学会北陸支部連合大会，講演番号 F1-9, 2016. 査読無

⑥ 須藤篤志，鈴木雄也，都司達夫，樋口健：動的木グラフの分割と再構成，情報処理学会研究報告，情報基礎とアクセス技術，2016-IFAT-122(1),1-6, 2016. 査読無

⑦ Makino M., Tsuji, T, Higuchi K.: History-Pattern Implementation for Large-Scale Dynamic Multidimensional Datasets and Its Evaluations. DASFAA (2) , 275-291, 2015. 査読有

⑧ Higuchi K., Yoshida M., Miyamoto N., Tsuji T.: A Routing Algorithm for Distributed Key-Value Store Based on Order Preserving Linear Hashing and Skip Graph. Applied Computing & Information Technology (Springer), 127-139, 2015. (SpringerBook chapter), 127-139, 2015. 査読有

⑨ 須藤篤志，都司達夫，樋口健：動的木グラフの分割による格納と検索， DEIM Forum 2015, 講演番号 E3-6,2015. 査読無

⑩ 千葉陽介，北嶋翔吾，都司達夫，樋口健：属性の動的追加を考慮したタプルデータセットの実装方式，情報処理学会，第 77 回全国大会講演論文集,1N-06, 611-612, 2015. 査読無

⑪ Tsuji T., Shimono R., Higuchi K. : Design and Evaluation of a Storage Framework for Supply Chain Management, Proc. of DEXA (2) 392-408, 2014. 査読有

６．研究組織

(1)研究代表者

都司　達夫　（TSUJI TATSUO）

福井大学・学術研究院工学系部門・シニアフェロー

研究者番号：80115302

(2)研究分担者

樋口　健　（HIGUCHI KEN）

福井大学・学術研究院工学系部門・准教授

研究者番号：50293410