維持コストを考慮したXPath問い合わせのビュー選択問題に関する研究*

青木 慎平†　古瀬 一隆‡　陳 漢雄§
†,‡,§ 筑波大学 システム情報工学研究科　〒305-8577 茨城県つくば市天王台 1-1-1

1 はじめに

現在XML形式のデータの利用が現在幅広く行われてきており，これに伴い，XMLの効率的な検索が重要になってきている．問い合わせ処理の効率的な手法の一つとして，実体化ビューを用いた問い合わせ機構の研究が注目されているが，このとき実体化するビューを限定し，利用価値の高いビューを選択しなければ効率的に運用できないことが知られている．

本研究ではデータ更新時の実体化ビュー維持コストに着目した，XPath問い合わせのビュー選択アルゴリズムを提案する．ビューの維持コストを考慮することにより，元データの更新によって発生するビューの更新コストを下げ，実体化ビューのより効率的が運用を行えるようになり，従来手法と比べてより有益なビューを選択できる．

2 問題定義

より多くのビューを実体化すると，それらを参照することで問い合わせ処理性能を向上させることができるが，更新コストや検索コスト，実体化空間の制限などにより，ビューを効率的に利用するためには，ビューの実体化量を制限し，一定量の有用なビューを選択することが求められる．具体的には問い合わせのログからビュー候補集合を生成し，これらを利益評価関数によって評価して利用価値の高いものを実体化する．このように問い合わせによるビュー候補から，有益なビューを選択して解を見つけることをビュー選択問題と呼ぶ．

本研究におけるビュー選択問題は，入力はDBに対する問い合わせ集合Qと，実体化空間制限B，出力は選択されたビュー集合MVとする．

3 従来手法

従来手法として，XQPMiner[4]，FastXMiner[3]について説明する．XQPMinerでは入力集合のXPath問い合わせを全て木構造として扱い，それらを組み合わせて大きな木構造を構築する．問い合わせの木構造を問い合わせパターン木(QPT)，組み合わせたものをグローバルな問い合わせパターン木(G-QPT)と呼ぶ．G-QPTを利用して，問い合わせ集合内の部分木ごとの頻度を計算し，頻出する部分問い合わせを発見して実体化ビューとして選択する．FastXMinerではXPathのワイルドカード演算による包含関係を考慮した頻度計算を行ってビュー選択を行う．

これらの手法は問い合わせ頻度のみに着目した手法であり，ビューの包含関係によるビュー間のデータ重複，ビューの維持コスト等への考慮が無いなどの問題点がある．

4 提案手法

4.1 概要

本手法では従来手法では考慮されなかった維持コストに関するパラメータを各ノードに保持する．また実体化ビュー同士のデータ重複を回避するため，ビュー包含グラフを構築する．本手法ではまず，ビュー包含グラフを構築してビューの包含関係を表現する．次に各ビューの維持コストを考慮した利益評価を行い，最後に空間領域制限の範囲内で利益の高いものから繰り返しビューを選択する．

4.2 ビュー包含グラフ

ビュー包含グラフは，ビュー間の包含関係をノードとエッジで表現するグラフ構造である．この構造を利用することでビュー同士のデータの重複を避けたビュー選択をすることが可能になる．図1にビュー包含グラフの例を示す．

4.3 利益評価式とビュー選択

選択されるビューに求められる条件は，更新頻度と実体化サイズの値は小さく，問い合わせ頻度が高く，実体化されたとき入力問合せの処理時間を削減することができるビューである．これらを考慮するため，各ビュー

*XPath View Selection Technique Exploiting Maintenance Costs
†Shinpei Aoki, Graduate School of SIE, University of Tsukuba
‡Kazutaka Furuse, Graduate School of SIE, University of Tsukuba
§Hanxiong Chen, Graduate School of SIE, University of Tsukuba

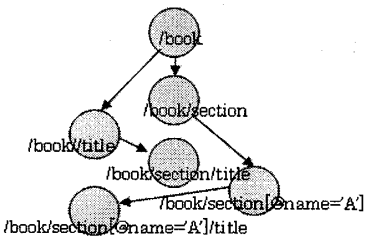

図 1: ビュー包含グラフ

v_i はそれぞれ，問合せ頻度 f_i，更新頻度 u_i，実体化サイズ s_i を持つ．またビュー集合 MV がすべて実体化された時の入力クエリ q の処理時間を $t_{MV}(q)$ とする．このときビュー v_i の利益は以下の式で表すこととする．

$$E_{v_i} = \frac{f_i}{u_i s_i \sum_{q \in Q} t_{MV \cup \{v_i\}}(q)}$$

処理時間については，$XP(/, //, *, [])$ に含まれる問い合わせについては文献 [1] によって，$O(|T| * |Q|)$ の処理時間で概算できることが示されている．$|T|$ は XML データのサイズ，$|Q|$ は問い合わせのステップ数を表す．これらの情報はビュー包含グラフ構築時に計算して設定する．

4.4 アルゴリズム

まず Q の問い合わせを解析し，ビュー包含グラフを構築，ビュー包含グラフのノード集合 V を候補集合とし，前項の利益評価の手法を用いてビューを選択，選択したビュー集合を出力に返す．以下にアルゴリズムの擬似コードを示す．

```
MV ← φ
V ← (Q のビュー包含グラフ頂点集合)

while |MV| < B ∧ V ≠ φ do
    v ← argmax_{v_i} f_i / (u_i s_i Σ_{q∈Q} t_{MV∪{v_i}}(q))
    MV ← MV ∪ {v}
    V ← V − {v}
end
```

5 評価実験

XMark ベンチマーク [2] によって生成した約 1GB の合成 XML データに対し，全体の 20%の領域に 80%の問い合わせを集中させた機械生成の 1000 クエリでビュー選択をした．選択されたビューを用いて同じ生成ルールで生成した 100 クエリの平均処理時間を，元データの何%の容量で実体化するかを変化させながら計測した．結果は図 2 である．

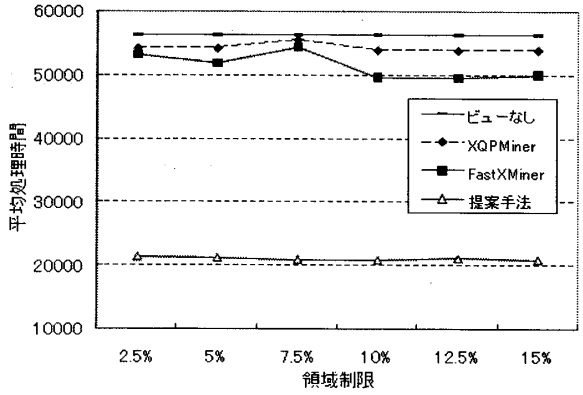

図 2: 問い合わせ時間平均

頻度のみに着目すると大きなビューを優先的に選択するため，この条件では実体化空間制限の超過を起こして効率的な選択ができず，むしろ検索コストを上げ性能を悪化させた．一方提案手法では小さいビューが有利なため，問い合わせ性能が向上するが，検索コストと実体化ビュー数はトレードオフの関係であるため，入力によっては効率的に利用できない場合もある．

6 まとめ

本研究では XPath 問い合わせにおける実体化ビューを用いた機構におけるビュー選択問題について取り組んだ．ビューの包含関係と各ビューの持つ維持コストを考慮したビューの選択手法を提案し，評価実験においてその性能について示した．

本手法の問題点としては，グラフ構築およびパラメータの設定の際に頻繁に DB にアクセスし，従来手法に比べて選択にかかる時間が著しく増大することが挙げられる．ビューの選択は頻繁に行う処理ではないが，DB アクセス回数を削減できるパラメータ取得方法を考案することで構築時間を削減でき，より実用上好ましいシステムの実現が可能となる．

参考文献

[1] G. Gottlob, C. Koch, and R. Pichler. Efficient algorithms for processing XPath queries. In *Proceedings of the ACM Transactions on Database Systems (TODS)*, volume 30. ACM, 2005.

[2] A. Schmidt, F. Waas, M. Kersten, M. J. Carey, I. Manolescu, and R. Busse. XMark: A benchmark for XML data management. *Proceedings of the 28th Int. Conference on Very large data bases(VLDB)*, 2002.

[3] L. H. Yang, M. L. Lee, and W. Hsu. Efficient mining of XML query patterns for caching. *Proceedings of the 29th Int. Conference on Very large data bases(VLDB)*, 2003.

[4] L. H. Yang, M. L. Lee, and W. Hsu. Mining frequent query patterns in XML. *Proceedings of the 8th Int. Conference on Database Systems for Advanced Applications(DASFAA)*, 2003.