XML における XPath 式の並列実行手法

関根　健太

近年，様々なデータを記述できるフォーマットとして XML(Extensible Markup Language)が普及しており，多数のシステムで利用されている．大量の XML データを管理・蓄積する場合，DTD(Document Type Definition)等のスキーマ言語を用いて XML データの構造をあらかじめ定義しておき，スキーマ定義に対して妥当なデータを作成・利用するのが一般的である．また，XML データへの問い合わせ言語として XPath(XML Path Language)がよく使われている．

ここで，XML データに対して XPath 式を用いて問い合わせを行うことを考える．近年，XML データのサイズが増大しているため，問い合わせ処理に要する時間が長くなってしまうという問題がある．これについては，XML データは文書の構造上サイズが大きくなってしまうこと，処理に要する時間が XML データのサイズだけでなく XPath 式のサイズに応じて増加することなどが処理時間の課題として挙げられる．近年，このようなデータサイズの増加という問題がある一方，計算機プロセッサの性能は向上しているため，この状況に応じた XPath 問い合わせ処理高速化手法の重要性が高まっている．Ruby などのプログラミング言語自体は複数のスレッドの生成・実行が可能であるが，XPath 処理系では複数スレッドの実行は考慮されていない．

本課題に関する研究として，Bordawekar らは，データ分割法，クエリ分割法，ハイブリット分割法という XML データや XPath 式を分割する 3 つの手法により，XML データに対する問い合わせの並列実行を提案している．しかし，これらの手法では，データの分割位置や XPath 式の分割位置を決定するために大量の統計情報を用いるため，分割位置決定処理の負荷が大きいという課題がある．また，これらのデータに更新が生じた場合，大量の統計情報を再構成する必要がある．

そこで本論文では，DTD を利用して XML データを論理的に分割し，その分割に基づいて XPath 式を複数スレッドで並列実行することにより処理時間の短縮を図る手法を提案する．本手法は DTD に基づいて複数出現し得る要素の数を XML データから求めており，容易に分割位置を決定できるという利点がある．また，生成される統計情報の量も少なくて済み，かつ高速に生成できる．提案手法の有効性を評価するため，XPathMark で定義されている 8 個の XPath 式を用いて評価実験を行った．その結果，提案手法を用いた場合，多くの場合において処理時間を大きく短縮できることを示した．

（指導教員　鈴木伸崇）