5M-06

XMLStreamの時系列イベント処理の性能評価

内田　友樹†　　松田　達希†　　藤田　悟‡
法政大学大学院 情報科学研究科†　　法政大学 情報科学部‡

1. はじめに

近年，高速ネットワーク上で時々刻々と変化する大規模なストリームデータをリアルタイムに取得し，活用できるようになった．そして，このデータを分析して価値を見出す CEP(Complex Event Processing)技術に注目が集まっている．一方，インターネット上でのデータ交換フォーマットとして XML が注目され，株価情報，気象情報，センサ等のデータが XML ストリームとして提供され，これを高速に処理するための CEP エンジンが求められている．そこで，XML ストリームからの複雑な検索要求に応えられる VPA(Visibly Pushdown Automaton)[2]ベースの XSeq[1]が提案された．Xseq は，時系列を組み込んだ検索ができるものの，単一ストリームしか扱うことができない．

本研究では，複数の XML ストリームに対して検索を行うために QLMXS(Query Language for Multiple XML Streams)という問い合わせ言語を開発した[4]．この言語では検索エンジンのコアに VPA を利用している．一つの検索要求から複数の VPA が生成されることがあり，それらを組み合わせて高速に実行することが求められる．

本稿では，QLMXS の検索を実現する VPA エンジンを実装する．そして，VPA の特性を生かしたエンジンの最適化を行い，性能を評価する．

2. VPA

VPA はプッシュダウンオートマトンの制約を強めたものであり，スタック操作がプッシュ，ポップ，インターナルの 3 種類に分かれていることが特徴である．スタック操作が明確化されたことにより，VPA は，和集合，積集合，補集合，連結，クリーネ*に対して閉じた性質を持っている．そのため有限状態オートマトンと同等の最適化を行うことが可能である．また，スタックの特性を活かし，XML，JSON ファイルのような入れ子構造のデータをモデル化することに適したオートマトンであると言える．

3. QLMXS

QLMXS は，XPath やその他 CEP 向け問い合わせ言語[1][3]を参考に設計した． XML ストリームから単純なデータの検索・抽出だけでなく，複数の XML に跨った解析を行うための複雑な条件記述が可能である．

```
Example. return stock_stream2
         select stocks/stock/price
         from stock_stream
```

上記の Example は，株情報ストリームから売値を抽出し，新しいストリームとして出力するためのクエリ例である．

4. QLMXS 検索エンジン

この検索エンジンでは，問い合わせ言語 QLMXS を用い，検索エンジンのコアには QLMXS の複雑なパターンを表現することができる VPA を用いる[5]．開発した QLMXS 検索エンジンの概要を下記の図 1 に示す．

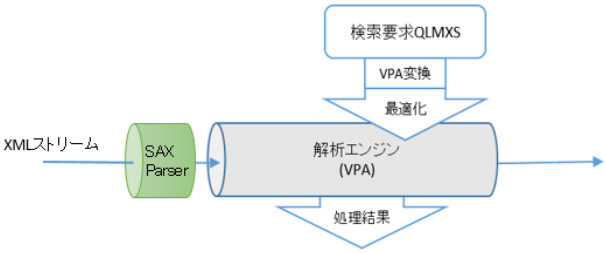

図 1　QLMXS 検索エンジン

蓄積されたデータを解析する従来のビックデータ処理とは異なり，CEP では次々に送られてくるデータを高速に解析するために，事前に解析エンジンを作成しておく必要がある．この解析エンジンでは，XML の階層構成の解釈のために VPA のプッシュダウンスタックを利用する．VPA の遷移関数にはオペレータを持たせており，トークンを用いて状態遷移を繰り返しながら，XML のキャッシュ，そして条件判定等の操作を行い，検索要求が満たされた時に出力を行う．

5. エンジンの最適化

QLMXS エンジンでは大量に送られてくるデータをリアルタイムに処理しなければならない．そのためエンジンの最適化をいくつか実装する．

5.1 VPA の状態数の削減

XML スキーマが利用可能な場合，条件，操作等で参照されていない，加えて，下位の状態から常に予測できる上位のパス表現から生成された状態を

Performance Evaluation of Sequential Event Processing for XML Streams
†Yuki Uchida, Tatsuki Matsuda, Graduate School of Computer and Information Sciences, Hosei University
‡Satoru Fujita, Faculty of Computer and Information Sciences, Hosei University

VPA から削除する．これにより削除した部分でのトークン操作を行う必要がなくなるため大幅な高速化を図ることができる．

5.2 VPA の合流

一つのストリームに対して複数の問い合わせを同時に処理したい時に，解析エンジンでは複数の類似した VPA を並行して処理する必要がある．各 VPA の処理には共通状態・遷移が存在し，この共通状態を合成した新たな VPA を作成することで最適化できる．これにより，トークンの無駄な重複操作が減り，VPA のスケーラビリティが向上する．

5.3 遅延評価

QLMXS で生成される VPA では，条件評価対象の要素が見つからない時は，検索条件が満たされることはない．そこで，順次トークン操作を行わず，ノードを一旦キューにためておき，対象の要素の入力が確認できたとき，はじめて，キューのノードを利用し、トークン操作を開始することができる．それ以外の場合，トークン操作は行わない．この処理は，XML の構造が規則的でない場合に有効である．

5.4 不要ノード処理の簡略化

クエリのパス表現に含まれないノードは通常は操作，条件等にも使わない不要ノードである．そこで，QLMXS エンジンでは通常一つ一つのノードに対して操作を行うが，連続する不要ノード処理については文字列のまま読み飛ばし，トークン操作を簡略化する．

6. 検証と考察

5.2 で述べた複数 VPA の合流を行うことによる効果を実験により検証する．実験環境を表 1 に示す．表 2 の問合せ Q1 と Q2 を VPA に変換し，解析エンジンにより並行に処理した時と，Q1 と Q2 の VPA の共通状態を一つに合成したもの(問い合わせ式では Q3)を処理した時，そして 5.1 および 5.4 で述べた最適化をした時の速度比較を行う．

図 2 の実験結果より，共通状態を合成した Q3 では，Q1+Q2 より 1.3 倍ほど処理速度が向上した．これは Q1 と Q2 が/issue/articles/article を共通部分として持ち，通常各 VPA にそれぞれノードを送り逐次トークン処理する所を，VPA の合流によりトークン操作数を半分にし，処理時間が大幅に短くなったためである．次に 5.1 の状態数削減を行った Q3(5.1)では，1.1 倍程の速度向上が示された．最後に 5.4 の不要ノード処理の簡略化を行った Q3(5.4)では，2 倍程の速度向上が示された．検索クエリが局所的なものであればあるほど操作不要ノードは増え，トークン操作数は減り高速になる．このことは表 3 の各トークン操作回数の差に顕著に表れており，図 2 と合わせてみることでトークン操作数が速度に影響を与えていることが分かる．また，XML のデータ容量が大きくなればなるほど最適化前との処理時間の差は大きくなることが示された．

表 1　実験環境

CPU	Intel(R) Core(TM) i5-2500K 3.3GHz x 4
メモリ	8GB
XML	SIGMOD Record

表 2　問合せセット

問合せ	問合せ式
Q1	/issue/articles/article/title/text()
Q2	/issue/articles/article/endPage/text()
Q3	/issue/articles/article[title/text() or endPage/text()]

表 3　データ容量 467KB 時のトークン操作回数

Q1+Q2	Q3	Q1+Q2 (5.1)	Q3 (5.1)	Q1+Q2 (5.4)	Q3 (5.4)
69518	34759	66376	33188	24198	15107

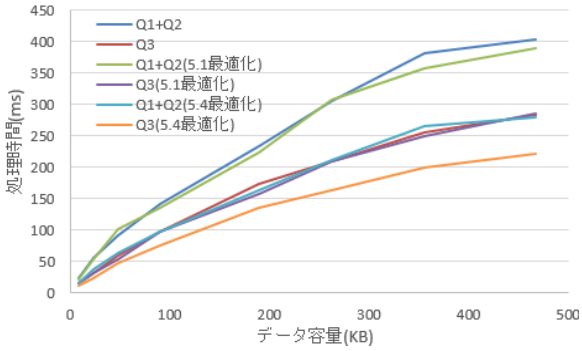

図 2　速度比較実験結果

7. まとめ

本稿では，QLMXS 検索エンジンの実装と VPA の最適化について述べた．そして XML を扱う上での QLMXS と VPA の特性に合わせた効果的な最適化を実現した．そして実験により大幅な速度向上を確認した．

今後の課題として，本稿では触れていない VPA の特性を生かした最適化を進めることによるエンジンの高速化と，実験で扱ったような単純な検索だけではなく，QLMXS で要求されるストリーム分割や複数の VPA を連携させる等のエンジン全体で見た時の高速化を進めていきたい．

参考文献

[1] Mozafari B., Zeng K., Zaniolo C., “High-Performance Complex Event Processing over XML Streams”, Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data, p. 253-264 (2012).

[2] Alur R., Madhusudan P., “Visibly Pushdown Languages”, Proceedings of the thirty-sixth annual ACM symposium on Theory of computing, p. 202-211(2004).

[3] Demers A. J., Gehrke J., Panda B., Riedewald M., Sharma V., White W. , “A General Purpose Event Monitoring System”, CIDR, Vol. 7, p. 412-422 (2007).

[4] Tatsuki M., Yuki U., Satoru F., “XMLStream 向け検索言語からの VPA の生成”, FIT2014, (2014).

[5] Yuki U., Tatsuki M., Satoru F., “VPA を用いた XMLStream 向け CEP エンジン”, FIT2014, (2014).