汎用 XML 文書符号化方式「XEUS」の検索性能評価

田中 卓弥† 井ノ上 直己† 浪岡 智朗‡ 三田村 好矩※

株式会社ＫＤＤＩ研究所† 通信・放送機構‡ 北海道大学大学院※

1．はじめに

インターネットの普及とともに WEB ベースのドキュメント管理や配信，データ交換・流通の分野で XML 検索の有効性が示され，様々な応用規格で利用されている．それは XML の大きな特徴である，プレーンテキストデータ形式であり自由にタグセットを定義できるという柔軟にデータ構造を変えることが可能な拡張性・可読性が高いことが挙げられる．しかし，この XML の特徴が，高い拡張性・可読性というメリットを生む反面，検索処理をおこなう際に冗長となり，コンピュータのメモリ消費を増大させ，処理負荷をかける．そこで本稿では，これらの問題点を解消するために，任意の XML 文書を対象とした汎用的な符号化方式の「XEUS」[1]（ゼウス）の特徴を利用した XML 文書データの効率のよい格納・検索方法を提案する．

本稿では，まず汎用 XML 文書符号化方式 XEUS の説明をおこない，その後 XUES による XML 文書の検索手法の説明をおこなう．そしてベンチマーク XML 文書データを用いて一般的な DOM による検索との性能評価比較を行い，本手法の有効性を示す．

2．XEUS について

XEUS とは「XML document Encoding with Universal Sheet」の略であり，任意の XML 文書をターゲットとした，汎用的な符号化方式であり，XML 文書の「論理構造」，「要素値／属性値のデータ型，符号長」，「要素名／属性名の符号化テーブル」等を定義した「XEUS シート」に従って XML 文書の符号圧縮をおこなう．

符号圧縮による伝送符号量だけでなく，受信側の処理負荷も低減することにより XML システム全体のパフォーマンスを向上させることが可能である．

3．XEUS による XML 検索手法

XML の検索は，XML 文書データから XML プロセッサが XML ツリーを作成し，その XML ツリーから検索アプリケーションが該当するノード集合を獲得し，そのノードの評価をおこなう．通常，XML プロセッサは XML 文書データを読み込むと，その XML 文書データ全てのパース処理をおこないメモリ上に展開しており，XML 文書のドキュメントサイズによっては，メモリサイズとメモリへの展開時間を大量に消費する．しかし XEUS による検索では，XEUS によってバイナリ化されたデータを読み込むと，そのバイナリからツリー（XEUS DOM）をメモリに展開するが，その展開される XEUS DOM は「XEUS シート」に書かれた「論理構造」，「要素値／属性値のデータ型，符号長」，「要素名／属性名の符号化テーブル」を参照することにより，検索対象となるノードのみが構築される．ノード集合の取得がおこなわれるときと，取得したノードの評価をおこなうときのみ，パース処理がおこなわれメモリ上に展開される為，イニシャルオーバーヘッドが少ない．また，この理由から通常では一旦メモリから消えると再使用時には再度パース処理をおこないメモリ上に展開するのに対し，XEUS DOM は再使用時のオーバーヘッドは少なくて済む．以上からメモリ消費量の観点からみても，検索の際にアクセスしたノードのみメモリ上に展開しているので，XML 文書のノード全てを辿るような検索（例えば count(//*)<XML 文書に含まれる要素ノードの個数>）をおこなわない限り，少なくて済む．

4.性能評価

XEUS による XML 検索性能を評価するために，検索時間の測定をおこなった．

計測方法は，XML 文書データを XEUS により符号化し，その符号化データから XEUS_DOM オブジェ

「Performance evaluation of Search method by XML document Encoding with Universal Sheet」

†Takaya Tanaka, Naomi Inoue・KDDI R&D Laboratories Inc.

‡Tomoaki Namioka ・ TelecommunicationsAdvancement Organization of Japan

*Yoshinori Mitamura・Hokkaido University

クトをメモリ上に展開し，XPath[2]式により検索をおこなった．比較として，同じ XML 文書データを JDOM[3]により DOM ツリーオブジェクトとしてメモリ上に展開し XPath にて検索をおこなった場合の測定もおこなった．XPath 式の値の評価をおこなう XPath エンジンはともに Jaxen[4]を用いた．評価に使用した環境は表 1 のとおりである．

CPU	Pentium-III-S 1.26GHz
Memory	512M バイト
OS	Red Hat Linux 7.3 2.96-110 (Kernel Linux version 2.4.18-3)
Java	Java2SDK 1.4.1_01-b01 (起動オプションにて，503MByte のメモリを確保)

表 1：評価環境

ベンチマークとなる XML 文書は，XML データベースのベンチマークプロジェクトである XMark プロジェクト[5]の xmlgen を利用し，XML 文書を生成した．xmlgen は ScalingFactor (SF)を指定することで，SF にほぼ比例したデータサイズのドキュメントを生成するツールである．データサイズは SF=0．01 で 約 1．1M バイト， SF=0．05 で 約 5．5M バイトである．今回の評価データとして，SF=0.0012 として XML 文書を生成した．生成した XML 文書のデータサイズは 1,339,240byte で，要素数 2054 個，属性数 423 個，テキストノード数 3770 個のデータである．評価の為の検索クエリは XMark プロジェクトで用意されているクエリのうち，XPath 表現が可能なもの 5 つを用意した．(表 2)

Q1	/site/regions/namerica/item[@id="item20748"]/name/text()
Q2	/site/open_auctions/open_auction/bidder[1]/increase/text()
Q3	count(/site/closed_auctions/closed_auction/price[text() >= 40])
Q4	/site/closed_auctions/closed_auction/annotation/description/parlist/listitem/parlist/listitem/text/emph/keyword
Q5	/site/closed_auctions/closed_auction[annotation/description/parlist/listitem/parlist/listitem/text/emph/keyword]/seller/@person

表 2：検索クエリ

5.評価結果

計測をおこなった結果を表 3 に示す．

検索クエリ	XEUS[msec]	JDOM[msec]
Q1	66	4259
Q2	54	4357
Q3	31	4293
Q4	39	4462
Q5	39	4284

表 3:検索時間

計測した結果，JDOM による DOM ツリーオブジェクトを構築する検索の場合と比較して XEUS_DOM オブジェクトを構築した検索の場合，64 倍から 138 倍高速に検索できた．また JDOM の場合，検索時間のほとんどがパース処理をおこない DOM ツリーオブジェクトを構築しメモリ上に展開する時間となるため，検索時間が検索クエリに依存していない．一方，XEUS は検索に必要なノードのみパース処理をおこない XEUS_DOM を構築しメモリ上に展開しているため，展開するノードが少ない検索クエリほど検索時間が速い．

6.まとめ

本稿では XEUS によって符号化された XML 文書の検索性能評価をおこなった．提案手法では，パース処理をおこないメモリ上へ展開するイニシャルオーバーヘッドのコストが圧倒的に少ない効果があり，検索時間の比較評価実験をおこなうことにより，その効果を確認できた．今後は更に詳細な評価をおこない，本手法の改良および有効性の検証をおこなう予定である．

謝辞

本研究にいつもご指導・ご助言頂いている KDDI 研究所松本取締役及びグラフィック処理グループの小林氏に感謝致します．

参考文献

[1] 小林ほか"XML 文書汎用符号化方式「XEUS」", 電子情報通信学会信学技報，DE2001-9，pp65-72

[2] XML Path Language (XPath) Version 1.0 http://www.w3.org/TR/xpath

[3] JDOM http://jdom.org/

[4] Jaxen http://jaxen.org/

[5] Xmark(An XML Benchmark Project) http://www.xml-benchmark.org/