

巨大XMLデータを管理し検索できる 分散XMLデータベース

Large-Scale Distributed XML Database Management System

服部 雅一 新名 博 金輪 拓也

■ HATTORI Masakazu

■ NIINA Hiroshi

■ KANAWA Takuya

近年、XML (Extensible Markup Language) データ利用の広がりとともに、特許データやバイオデータなど数テラ (T: 10^{12}) バイトを超える巨大XMLデータが出現している。従来の分散データベースシステムでは、サーバ間のデータ通信がボトルネックとなり、サーバを増やしてもそれに見合った処理速度の向上が困難であった。

東芝は、このような巨大XMLデータに対して高速にデータアクセスするため、ネットワークに分散された複数サーバから構成される分散XMLデータベースを開発した。分散並列処理などの技術で、1 Tバイトを超えるデータに対しても、高速なデータアクセス応答性と拡張性を実現した。

With the widespread diffusion of the extensible markup language (XML) format, large-volume XML data of several terabytes in size have been appearing in fields such as patent data and biological data. In a conventional distributed database management system (DBMS), data communication bottlenecks occur between the distributed servers, creating difficulty in maintaining high performance proportional to the number of servers.

In order to provide reliable and real-time access to large-volume XML data, Toshiba has developed a distributed XML DBMS consisting of a collection of multiple, interrelated databases distributed over a computer network. By incorporating technologies such as distributed parallel processing, our new distributed XML DBMS realizes quick access response and server scalability, even with large XML data of more than 1 terabyte in size.

1 まえがき

電子政府をはじめとして、電子商取引、事務文書管理、新聞コンテンツ管理などインターネット上での様々なデータ交換領域で、汎用的なデータ表現の仕組みとして、データ構造の柔軟性を特長とするXMLの活用が進んでいる。

近年、XML活用範囲の広がりとともに特許データやバイオデータなど各分野で数百Gバイト、更に、1 Tバイトを超える巨大XMLデータが出現している。1 Tバイトのデータは、公開特許公報に換算すると約100年分の情報量に相当する。

このような巨大データを管理し、実用的な検索の応答性を確保するには、データをネットワーク上にある複数のサーバに分散配置させる分散化の手法が効果的である。

東芝は、1 Tバイトを超える巨大XMLデータに対して、実用的な検索応答性を達成するため、分散XMLデータベースを開発した。

ここでは、分散XMLデータベースの分散検索処理を中心に、システムの特長と概要について述べる。

2 分散XMLデータベースの特長

2.1 XQueryによるアクセス透過性を実現

問合せ言語としてXQuery (An XML Query Language)⁽¹⁾を提供している。XQueryは、W3C (World Wide Web Con-

```
for $sale in db("販売")//販売情報
where starts-with($sale/日付,"200503")
and contains(. //text(), "横浜")
return <sale> {$$sale/顧客/氏名}</sale>
```

図1. XQuery検索式の例 — 属性とキーワードの検索を組み合わせたXQuery検索式の例で、XQueryは、データベースにピンポイントでアクセスして、欲しいデータだけを取り出すことができる。

Example of XQuery

sortium) で標準化されている問合せ言語であり、XMLデータの取得や加工に威力を発揮する表現力の高い関数型言語である(図1)。

分散XMLデータベースで、ユーザーはサーバの位置や複数のデータベースのデータ定義を意識することなく、あたかも一つのXMLデータベースであるかのようにXQueryを使って検索できる。

2.2 分散化しても高い拡張性と柔軟性を実現

当社は、複数のサーバで動作させるデータベースとして、独自に開発してきたネイティブXMLデータベースを拡張して使っている^{(2), (3), (4)}。ネイティブXMLデータベースは、スキーマ (XML文書の構造) 定義を必要とせず多様なXMLデータを管理できるため、拡張性と柔軟性に優れている。

分散XMLデータベースでも、この高い拡張性と柔軟性を達成している。例えば、既にある別のXMLデータベースには手

を加えずに、分散XMLデータベース環境に追加登録できる。既存のXMLデータベースを追加登録するイメージの詳細については3章で述べる。これにより、散在した複数のデータベースを統合し、XQueryを使った横断検索などが可能となる。

2.3 分散並列処理技術による高速検索を実現

サーバの並列動作が期待される分散データベースで、データ受信待ちなどで逐次処理部分が増えて分散化効果が低下してしまう。そこで、以下のような技術でデータ受信待ちなどの逐次処理部分を削減し、高速検索を実現している。

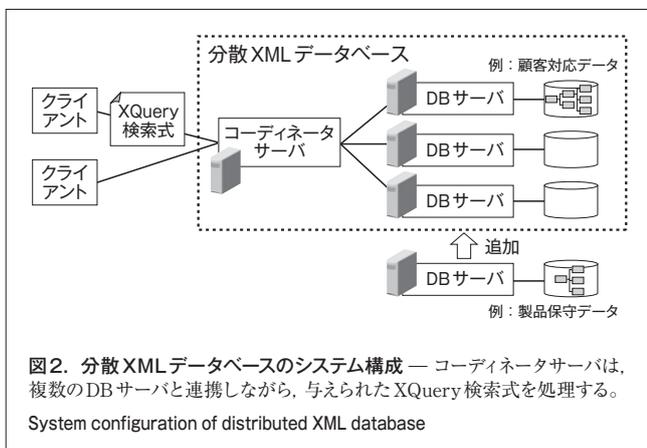
- (1) XQuery分散並列処理技術 パイプラインパラレル (サーバ間の手順操作に対する並列化) などの複数プロセスや複数サーバで並列化を進め、サーバの稼働率を向上させた (4.1節参照)。
- (2) XML転送コスト削減技術 分散データベースでは、サーバ間のデータ転送コストが性能劣化の要因となる。XMLデータの内部形式を工夫して、データ転送処理を効率化できた (4.2節参照)。

3 分散XMLデータベースのシステム概要

システム構成を図2に示す。1台のコーディネータサーバと複数のデータベースサーバ (以下、DBサーバと略記) から構成されている。DBサーバは、当社が開発してきた単体のネイティブXMLデータベースを使っている。XMLデータ全体を水平分割して各DBサーバに割り振り、DBサーバを取りまとめる中央サーバとしてコーディネータサーバを配置している。コーディネータサーバは、複数のDBサーバと連携しながら、与えられたXQuery検索式を処理する。

分散XMLデータベースにおける分散XQuery処理の概略手順を以下に述べる。

- (1) クライアントからXQuery検索式がコーディネータサーバに送信されると、コーディネータサーバは、各DBサーバと送受信する転送データの内部形式、及びそれら結果



リストの結合や集計の手順などの全体プランを作成し、関連する部分プランをDBサーバに送信する。

- (2) DBサーバは、部分プランを具体化するため各XMLデータベースのディスクI/O (Input/Output) を含めた局所的な手順に関する最適化を行い、詳細プランを作成する。
- (3) コーディネータサーバとDBサーバは同時にそれぞれのプランを実行する。コーディネータサーバは、DBサーバの処理結果を収集した後、XMLデータ列として結果リストをクライアントに送信する。

前述のとおりDBサーバはネイティブXMLデータベースであるため、コーディネータサーバはデータに関する統計情報など各データベースの情報を収集していない。そのため、コーディネータサーバとDBサーバで検索処理の最適化を図っている。

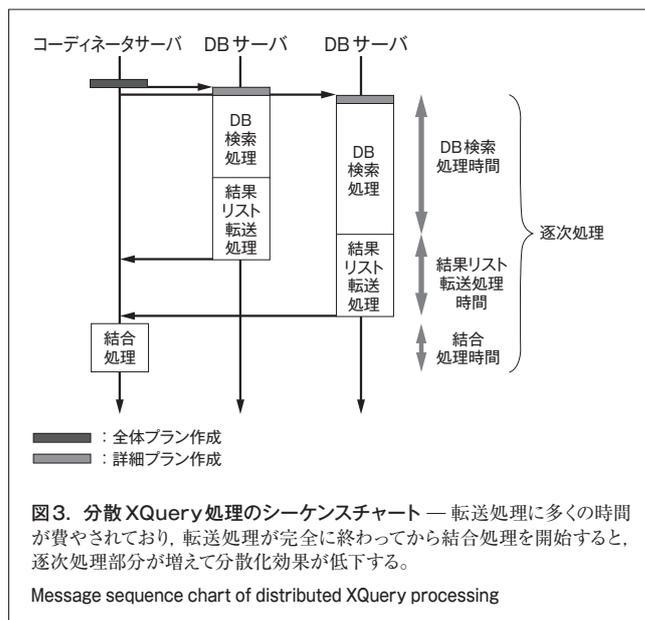
4 分散XQuery処理の高速化

従来手法に基づく分散XQuery処理のシーケンスチャートを図3に示す。転送処理に多くの時間が費やされていることがわかる。更に、転送処理が完全に終わってから結合処理を開始すると、逐次処理部分が増えて分散化効果が低下してしまう。

そこで、分散XQuery処理を高速化するために、XQuery分散並列処理技術とXML転送コスト削減技術、の二つの技術を開発した。

4.1 XQuery分散並列処理技術

DB検索処理と結果リスト転送処理を対に並列化して実行するXQuery分散並列処理技術を開発した。以下の2種類の並列化を実現している。



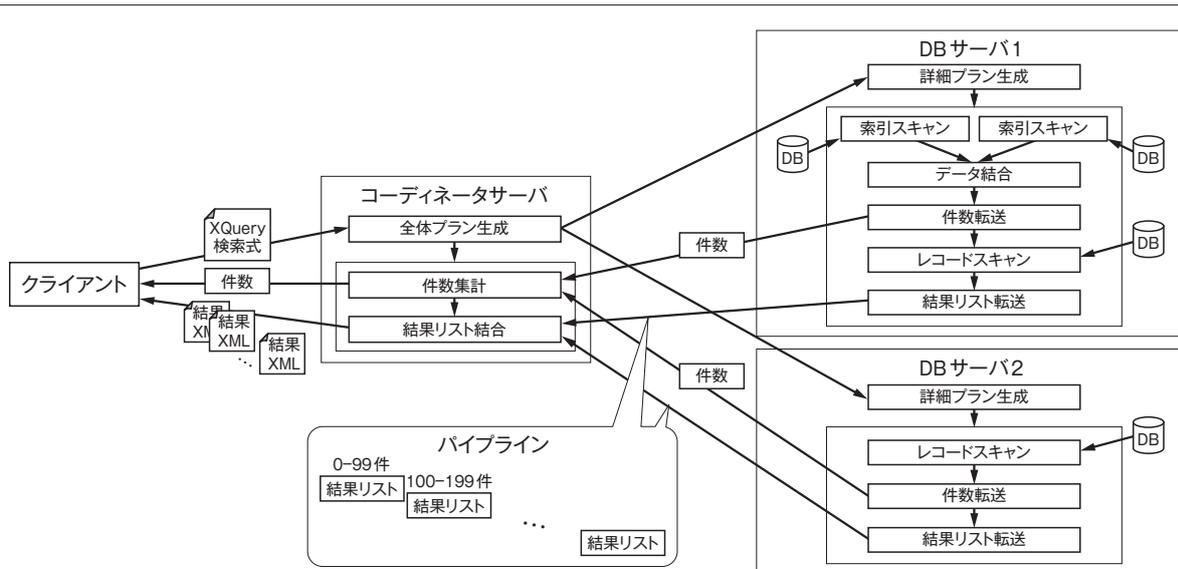


図4. 分散XQuery処理の一例 — DBサーバ1とDBサーバ2は、それぞれ索引スキャンとレコードスキャンから順に処理を開始し、得られた結果リストをコーディネータサーバに転送する。コーディネータサーバは、二つのDBサーバから結果件数を収集し、XMLデータ列として結果リストを作成する。

Example of distributed XQuery processing

(1) 各DBサーバにおける検索処理 DBサーバは、受け取った部分プランから詳細プランを作成し、検索処理を進める。詳細プランは、スキャンやソートなどの複数の基本処理のデータフローとして表現される。各DBサーバの基本処理は、独立して並列に動作し、前段の基本処理からの入力イベントを受けるとデータ処理を実行するパイプライン動作を行い、結果リストを作成する。

(2) 連携する複数サーバ間における転送処理 DBサーバは、前述の検索処理で得られた結果リストを順次コーディネータサーバに転送し、コーディネータサーバは収集した結果リストを処理する。この動作が、各DBサーバから並行して行われる。

図1のXQuery検索式に対する分散XQuery処理の一例を図4に示す。DBサーバ1及びDBサーバ2は、それぞれ索引スキャンとレコードスキャンから順に処理を開始し、それぞれ得られた結果リストをコーディネータサーバに転送する。コーディネータサーバは二つのDBサーバから結果件数を収集し集計するとともに、転送された結果リストも収集して結合し、XMLデータ列として結果リストを作成する。

4.2 XML転送コスト削減技術

分散データベースでは、サーバ間のデータ転送コストが性能劣化の要因となる。ツリー構造を持つXMLデータの内部形式をDOM (Document Object Model) などのノードとアークで表現した場合、従来はサーバ間でそのままでは転送できず、直列化するなど転送前にデータ変換処理を施す必要があった。

そこで、XMLデータの内部形式をノードとアークを使わない

転送に適したデータ列として表現し、検索処理ではそのデータ形式のまま演算できるように設計したことで、転送前のデータ変換処理を削減し、結果リストの転送処理時間を短縮できた。

5 分散全文検索機能

分散XMLデータベースではXQueryによる検索だけでなく、高度な全文検索機能を使った検索も可能である。全文検索機能の主なものを以下に述べる。

- (1) Nグラム索引方式による漏れがない全文検索
- (2) 形態素索引を用いた自然文検索
- (3) XQueryを全文検索拡張した言語XQFT (XQuery and XPath Full Text)⁽⁵⁾に対応
- (4) 同一視検索, 近傍検索, 及び辞書を用いた検索
- (5) スコアリング機能

XQFTを使った全文検索の一例を図5に示す。これは、bibというデータベース中の<book>タグの下にある<title>タグの文字列の中に、“web”, “site”, “usability”という三つの単語が大文字と小文字を区別せずに6単語以内に出現するものを検索し、そのスコアとともに表示するものである。

この検索式を処理する場合、まず各DBサーバで検索条件に合致する文書をスコア計算しながら結果リストを作成する。次いで、結果リストをコーディネータサーバに転送し、コーディネータサーバで結果リストを結合することにより、XMLデータ列として結果リストが得られる。

このような分散全文検索を、特別なアプリケーションを新たに開発することなく問合せ一つで実現できる。

```

for $x score $y in
db("bib")/book[./title ftcontains
"web" && "site" && "usability"
window 6 words
case insensitive]
return <result score="{ $y }">{ $x / title } < / result >

```

↓

```

検索結果
ヒット件数：2512件
----- [1/2512] -----
<result score="1.00">
  <title>web site usability information</title>
</result>
----- [2/2512] -----
<result score="0.98">
  <title>WEB Usability: this site is ...</title>
</result>
.....

```

図5. XQFTを使った全文検索の例 — bibというデータベースの中の<book>タグの下にある<title>タグの文字列の中に、“web”、“site”、及び“usability”という三つの単語が大文字と小文字を区別せずに6単語以内に出現するものを検索し、そのスコアとともに表示する。
Example of XQuery and XPath Full Text

6 性能評価

分散検索機能の性能を検証するために、1 Tバイトを超える規模のデータに対する検索応答時間の計測実験を行った。実験では、パイプライン処理など今回の分散XQuery処理技術を適用する改善版と、適用しない従来版の2種類の分散XMLデータベースを用いて応答速度のデータを取得し、その応答速度の改善比を求めた。

測定環境は次のとおりである。コーディネータサーバ1台と

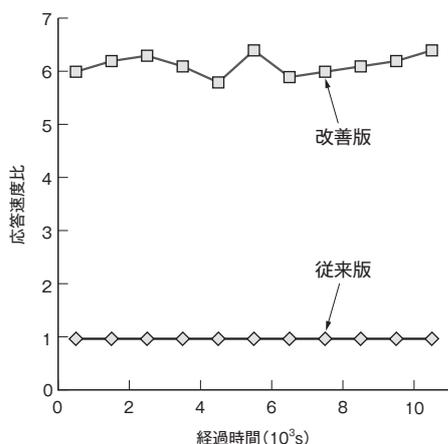


図6. 分散XQuery処理技術の効果 — 分散XQuery処理技術を適用すると、1 Tバイトを超える規模のデータに対する検索応答速度は、適用しない場合に比べて定常的に約6倍速くなった。
Effect of distributed XQuery processing

DBサーバ10台の分散構成とした。データは6,000万件の技術文書で約2 Tバイトのデータを、DBサーバ10台に対して均等に格納した。そして、1台のクライアントから同時に4トランザクションで検索を連続実行した。

実験開始から3時間での平均応答速度の比を定期的にプロットしたものを図6に示す。今回の技術により、従来と比較して約6倍の応答速度を定常的に実現できた。

7 あとがき

XQuery分散並列処理技術やXML転送コスト削減技術などを用いて、特許データやバイオデータといった1 Tバイトを超える巨大XMLデータに対しても、実用的な検索の応答性と拡張性を実現できるようになった。また、事前にデータ構造を明確に定義しなくてもよいので、多種多様なXMLデータを管理できる。

更に、XMLの高い柔軟性を生かして、既に運用されている複数のデータベースを、仮想的に統合したデータ統合システムへ適用することも期待されている。今後、これらの用途に向けて早期の実用化を目指す。

文献

- (1) World Wide Web Consortium. "XQuery 1.0: An XML Query Language". World Wide Web Consortiumホームページ. 入手先<<http://www.w3.org/TR/XQuery/>>. (参照2009-02-16).
- (2) 服部雅一, ほか. 高速性と信頼性を両立したコンテンツ管理向けネイティブXMLデータベース. 東芝レビュー. 59, 2, 2004, p.54-57.
- (3) 谷川 均, ほか. 大規模でも高速な検索を実現するXMLデータベースTX1. 東芝レビュー. 60, 7, 2005, p.71-75.
- (4) 服部雅一. 巨大XMLデータを管理・検索する分散XMLデータベース. 東芝レビュー. 62, 10, 2007, p.62-63.
- (5) World Wide Web Consortium. "XQuery and XPath Full Text 1.0". World Wide Web Consortiumホームページ. 入手先<<http://www.w3.org/TR/xpath-full-text-10/>>. (参照2009-02-16).



服部 雅一 HATTORI Masakazu

研究開発センター 知識メディアラボラトリー研究主幹。
XMLデータベースの研究・開発に従事。情報処理学会会員。
Knowledge Media Lab.



新名 博 NIINA Hiroshi

研究開発センター 知識メディアラボラトリー主任研究員。
XMLデータベースの研究・開発に従事。
Knowledge Media Lab.



金輪 拓也 KANAWA Takuya

研究開発センター 知識メディアラボラトリー研究主務。
XMLデータベースの研究・開発に従事。
Knowledge Media Lab.