

大規模でも高速な検索を実現する XML データベース TX1

TX1 XML Database Realizing High-Speed Searches in Large-Scale Environments

谷川 均

■ TANIGAWA Hitoshi

服部 雅一

■ HATTORI Masakazu

松井 浩二

■ MATSUI Koji

佐藤 実

■ SATO Minoru

コンテンツのXML (eXtensible Markup Language) 化が急激に進んでいる。その結果、コンテンツを扱う多くのIT (情報技術) ソリューションのプラットフォームにおいて、大量のXMLデータを効率的に蓄積、検索、管理する機能の重要度が高まっている。

東芝及び東芝ソリューション(株)は、このニーズに応えるものとして、XMLデータをそのまま格納でき、かつデータが大規模になっても高速に検索できるXMLデータベース“TX1”を開発、商品化した。小規模データでしか実用的な性能の出ない従来のXMLデータベースでは実現できなかったソリューションが、TX1により現実のものとなった。

With "XMLization," or the expression of electronic documents in Extensible Markup Language (XML), becoming more common, requirements for the efficient storage, searching, management, and processing of such XMLized documents are increasing in line with their importance as functions of IT solutions.

Toshiba and Toshiba Solutions Corp. have developed a native XML database software called TX1, which can search large-scale databases at high speed to realize practical new solutions.

特集

安心と安全

1 まえがき

XML (eXtensible Markup Language) は、データの意味を表すタグを目的に応じて定義できる“拡張可能なマークアップ言語”の国際標準データ記述言語である。

XMLはW3C (World Wide Web Consortium) から1998年に最初に勧告されたが、そのデータ記述の柔軟性から、当初はBtoB (企業間の取引) など企業間のデータ交換の記述言語として利用されてきた。その後、企業や行政などで扱われる文書が電子化されていくに従って、徐々にXMLの適用範囲は広まり、最近では、これらの文書もXMLで記述されることが一般的になってきている。

企業間や企業内のコンテンツで取り扱われるデータは、その構造化のレベルから、以下の三つに大別できる。

- (1) 形式が決まっている“構造データ”
- (2) 形式があいまいな“半構造データ” (議事録や設計データなど)
- (3) “非構造データ” (メモなど)

XMLは半構造データモデルに属する言語として、これら三つの異なるデータを同一の枠組みで表現することができる。そのため、コンテンツのXML化が急速に進んでおり、それに伴って、コンテンツを扱う多くのITソリューションのプラットフォームにおいて、大量のXMLデータを効率的に蓄積、検索、管理する機能の重要度が高まっている。

東芝及び東芝ソリューション(株)は、このニーズに応えるものとして、XMLデータをそのまま格納でき、かつデータ量が大規模になっても高速に検索できるネイティブXMLデータベース(以下、XMLDBと略記)“TX1”を開発、商品化した。

2 TX1の特長

2.1 RDBや従来のXMLDBとの比較

XMLデータの管理には、RDB (Relational DataBase) やXMLDBが用いられることが多い。一般的なRDBと従来のXMLDB及びTX1について、データ量、データ構造、検索方法の違いによる検索性能の比較モデルを表1に示す。

RDBは、取り扱うXMLデータが形式の決まっている構造データの場合には検索性能がよい。しかし、構造の定義が困難な非構造データでは、通常CLOB (Character Large Object) のようなバイナリ管理になってしまうため、検索性能が大幅に劣化してしまうことがわかっている。また、XMLの高い記述力から要求される、全文検索や階層構造に関する検索 (親子や兄弟などの要素を組み合わせる階層構造をたどったり、条件を満足する部分階層を抽出したりする検索で、階層パターン照合と呼ぶ) といったXML特有の検索で性能を出すのが難しい。

また、RDBには、これまでに蓄積されてきたノウハウが利用できるというメリットもあるが、データ構造に変更があった

表 1. XML データの管理方法の比較

Comparison of database systems for XML data

項目		一般的な RDB	XMLDB				
			従来の XMLDB		TX1 のターゲット		
			完全一致性能追求型	全文検索エンジン型			
検索性能	データ量とデータ構造	小規模 (~1G バイト)	ばらつき小 (構造)	○	○	○	○
			ばらつき大 (非構造)	×	○	○	○
	中規模 (~100 G バイト)	ばらつき小 (構造)	○	△	△	○	
		ばらつき大 (非構造)	×	△	△	○	
	大規模 (100 G バイト~)	ばらつき小 (構造)	○	×	×	○	
		ばらつき大 (非構造)	×	×	×	○	
検索方法	一致検索		◎	◎	×	◎	
	全文検索		△	×	○	◎	
	XML 特有の検索 (XQuery)		×	○	×	◎	
更新性能		○	○	○	○	○	

◎ ○ △ ×
得意 ← → 不得意

いる“XQuery 1.0: An XML Query Language”⁽¹⁾ (XQuery) が業界標準になりつつある。TX1でも XQuery の主要機能をサポートしている。XQuery は基本的には RDB における SQL (Structured Query Language) と類似した問合せ言語であると言えるが、SQL にない機能として、XML データの“階層パターン照合”がある。XQuery の例を図 1 に示す。

```
FOR $book IN document("bib.xml")/book
LET $a:= $book/author
WHERE contains($book/company,"toshiba")
RETURN
<book> {
  $book/title,
  <count>Number of authors: {count($a)}</count>
}</book>
```

図 1. 問合せの例 — 出版社が東芝である本のタイトルと著者の数の一覧を表示している。

Example of query data

場合、それに合わせて表構造を変更しなければならないというデメリットもある。

一方、XMLDB は、多種多様な階層構造を持つ XML データを“ネイティブに(そのまま)”格納できる DB である。XML データの格納や取得にあたって、マッピング処理のようなデータ構造を考慮した処理が存在しないため、ソリューション開発中やビジネス環境の変化により発生するデータ構造の変更にも、比較的柔軟に対応できる。

XMLDB は XML データをそのまま格納できるため、非構造データを容易に扱えるが、XML 特有の検索を高速化するためにはメモリ指向にならざるをえず、その結果、スケーラビリティが低く、小規模なデータ量を扱うシステムに用途が限定されてしまっていた。これが、結果的に XMLDB の普及を妨げた最大の要因であると考えられる。

TX1 は、データ量にかかわらず、構造及び非構造のいずれのデータでも、更に、一致検索、全文検索、及び XQuery (XML データに対する標準問合せ言語) のいずれの検索でも、高性能であることが最大の長である。

2.2 TX1 の高速化技術

TX1 において“XML データを大規模でも高速に検索”を実現するために、以下の三つの技術を考案した。

- (1) 原文の構造抽出・格納技術
- (2) XQuery に適した問合せ最適化技術
- (3) ディスク入出力 (I/O) の効率化・最適化技術

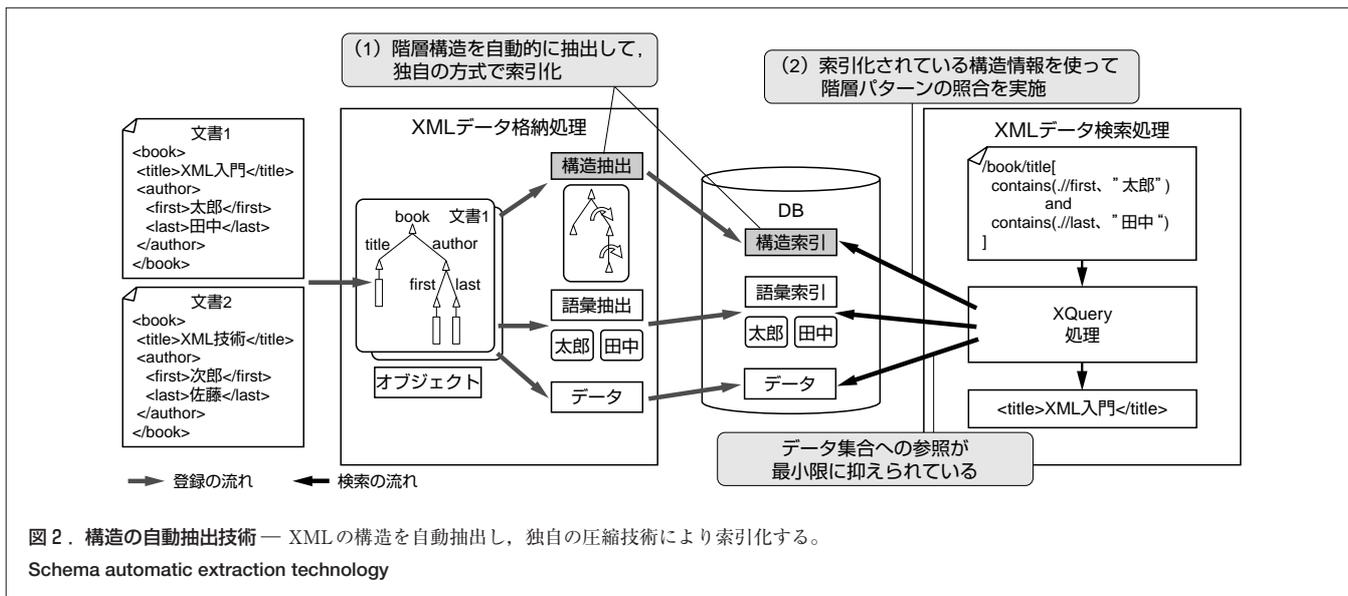
2.2.1 原文の構造抽出・格納技術

現在、XML データに対する問合せ言語としては、W3C において策定されて

前述のとおり、親子や兄弟などの要素を組み合わせる階層構造をたどったり、条件を満足する部分階層を抽出したりする検索機能が階層パターン照合である。データ構造があらかじめ定義されていれば、データ出現位置が特定できるため階層パターン照合は不要であるが、XMLDB が対象とする、データ出現位置が特定できない半構造データや非構造データの検索には、この階層パターン照合が必須である。

従来、階層パターン照合は、一般的に XML データ内の階層構造を順にたどる方式で実装されていたため、ディスク I/O 時間や CPU 使用時間といった処理コストが高く大規模なデータでは、応答性を保証することが難しかった。TX1 では、次に述べるような索引方式を考案し、大規模データでも、高速な階層パターン照合を実現した。

- (1) 構造情報の自動抽出と索引作成 データ構造が定義されていない XML データから、登録時に、緩やかな構造情報を自動抽出する。要素の繰返し回数など XML データごとの厳密な構造情報の抽出も可能ではあるが、その場合、XML の記述性の高さから膨大な数の構造情報が生成され、結局、格納や検索時の性能劣化につながってしまう。そこで、DB には共通した部分構造を持つ XML データが多く格納されるという傾向に着目し、それを表現する緩やかな構造情報を抽出する。そして、抽出した構造情報を基にした独自の索引を作成する。
- (2) 自動抽出した構造情報を使った検索 XQuery で記述された問合せにおいて階層パターン照合が発生した場合、自動抽出された構造情報を基にした索引を使って DB 内を参照する。この構造情報索引を基に



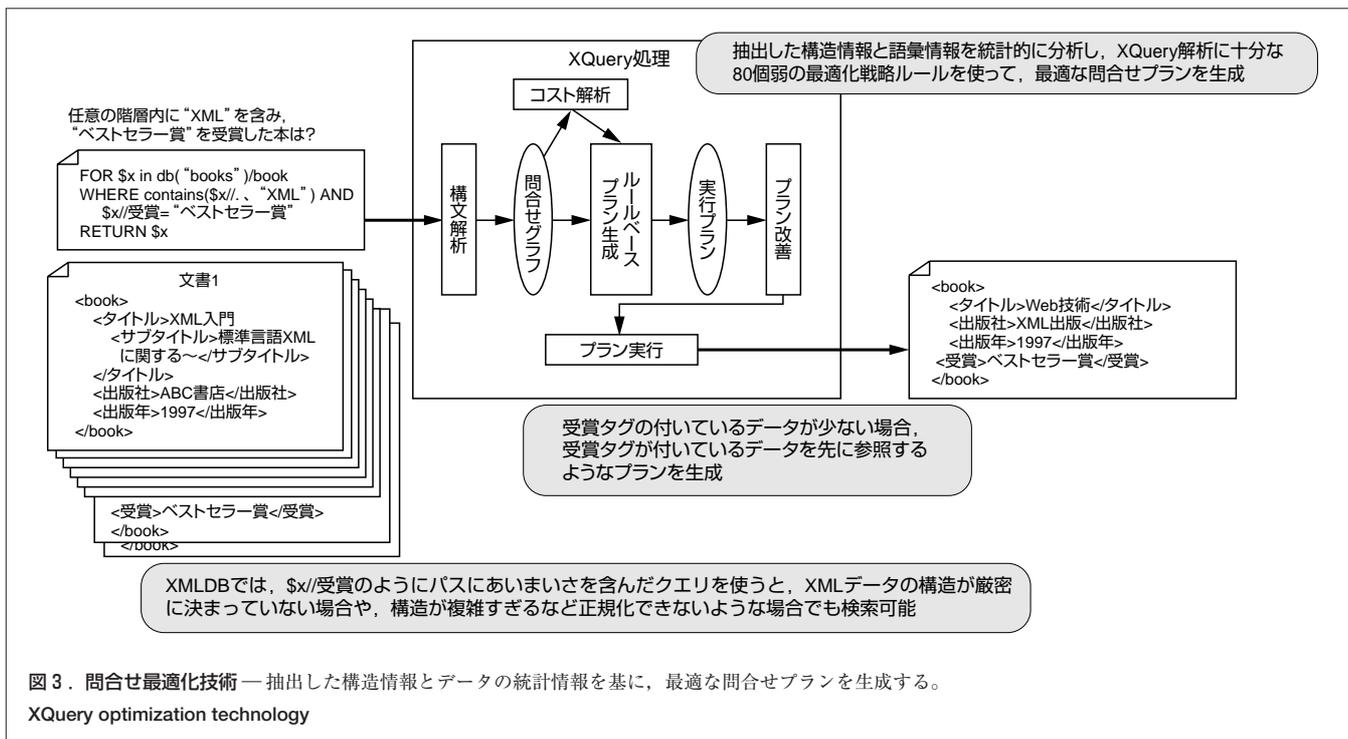
データ発生箇所を特定することで、参照範囲を限定したり、参照自体をスキップしたりし、高速な階層パターン照合を実現した。構造情報の自動抽出と検索技術の概念を図2に示す。

2.2.2 XQueryに適した問合せ最適化技術 XQueryは、階層パターンを表すノード間の親子、兄弟、先祖などに代表される制約関係に変換できる。クエリの制約関係を解いていく手順(プラン)が適切でなければ大幅な性能劣化が生じ、大規模データになればその傾向は顕著になる。

現在、TX1にはXQueryを解析するのに十分な“語彙(ごい)

索引適用ルール”、“ノード上昇スキャン適用ルール”、“構造索引適用ルール”など多くの最適化戦略ルールが組み込まれている。索引や自動抽出した構造に関する統計情報をヒントに、これらの戦略ルールを適用していき、ほぼ最適なプランを高速に生成する。生成されたプランは、重複した処理を一つにまとめるなどのプラン改善を行ってから実行される。

図3は、“任意の階層内にXMLを含み、ベストセラー賞を受賞した本”に関する問合せである。このクエリの場合、“全文検索(中間一致検索)の高速化に用いる語彙索引を参照して、構造が合っているものを探すプラン”と“構造が合ってい



るものから語彙が一致するものを探すプラン”が候補となる。TX1では、語彙索引の統計情報と自動抽出した構造の統計情報を基に、そのつど状況に応じた低コストプランを生成する。例えば、受賞タグの付いているデータが少ない場合、“\$x//受賞”のようなパス指定にあいまい性を含んでいるクエリであっても、受賞タグが付いているデータを先に参照する。

2.2.3 ディスクI/Oの効率化・最適化技術 数十Gバイトオーダーの中～大規模データになれば、索引情報を含めた全データをメモリ上にキャッシュすることはできない。ディスクアクセスを前提にしたI/Oの低減と最適化は高速化のためには不可欠である。特に語彙索引は数十Mバイト単位のスキャンが行われ、また、指定されるキーワードも様々であるため、キャッシュヒット率は必然的に低くなる。

このような場合でも、高速なI/Oを実現するため、同一種類のデータ(例えば、同じ語彙の索引)を連続的に配置する方式を考案した。この方式をベースにすることで、主に検索時に、次のような高速化手段が実現できる。

- (1) 連続データへの非同期一括先読み
- (2) ディスクアレイを使った並列読出し

一方、このようなデータ配置にすると、ディスク書込みがランダムになるため登録時のI/O性能は劣化する。しかしTX1では、以下の方式を考案して性能劣化を防止している。

- (1) 登録データ内に現れる同じ語彙は一括して登録する。
- (2) 通常、大量に発生する語彙種のパラツキを、検索性能にインパクトを与えない範囲で統合する。これにより登録時のI/O回数を大幅に削減する。
- (3) ランダムな更新ページを一時格納データファイルに連続配置して、書込みコミット時の応答時間を短くする。
- (4) 一時格納データファイルから、DBへの実更新は複数トランザクション分を遅延してまとめて行う。

また、TX1では、“トランザクションによりデータ更新が発生すると、そのオリジナルデータのコピーを作るリカバリ方式”を採用しているが、そのオリジナルとコピーのバージョンを適切に制御することで、問合せと更新の各トランザクションの競合を最小限に抑えている。

3 TX1の性能

3.1 大規模データ

従来のXMLDBでは対処できなかったサイズのデータとして、特許公開公報8年分(原文サイズ:約100Gバイト、文書数:約300万件)のデータに対する検索性能を測定した結果を図4に示す。

TX1は、全文検索や、タグとキーワードを組み合わせたきめ細やかな検索が高速で、また、仮に大量にヒットしたケースでも結果確定(件数取得)が高速に行われる。

```

■キーワード検索 (972件該当) . . . 0.69秒
db("patent")/"/jp-official-gazette[contains(./text(),"XML")]

■分類コード検索 (2万8千件該当) . . . 1.6秒
db("patent")/"/jp-official-gazette
[bibliographic-data/classification-ipc/main-clsf/text()
[starts-with(., "G06F3") or starts-with(., "G06F7")] //invention-title
G06F : 電氣的デジタルデータ処理の分類コード

■分類コード+キーワード (59件該当) . . . 1.9秒
db("patent")/"/jp-official-gazette
[bibliographic-data/classification-ipc/main-clsf/text()
[starts-with(., "G06F3") or starts-with(., "G06F7")]
and contains(./text(), "XML")] //invention-title

*ハードウェア仕様 CPU : Intel® Xeon™ 3.06 GHz×2,
メモリ : 2 Gバイト, ストレージ : AF2000

```

図4. 特許検索のクエリ例 — 特許検索のクエリ例とその性能測定結果である。

Example of query for patent search

なお、特許文書は、RDBで管理するには構造が複雑で、また、従来のXMLDBで管理するには文書数及びデータ量ともに多く、一般的にどちらも難しかった。

3.2 ベンチマーク

XQueryの代表的なベンチマークにXMark⁽²⁾がある。XMarkは、インターネットオークションをモデルとしたデータに対する20種類のクエリを提示している。約100Mバイトのデータを用いた、XMarkによるTX1のベンチマーク結果が図5である。

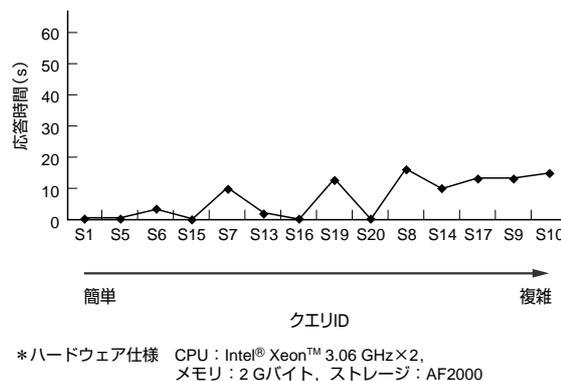


図5. XMarkによるベンチマーク — XMarkを用いたTX1の性能ベンチマーク結果である。

Benchmark for XML database by XMark

XMarkにより提示された20種類のクエリのうち、図5に提示した、実際のシステム運用でよく用いられるパターンのクエリでは、実用的な検索性能が出ていることがわかる。

なお、この値は、既存のXMLDB製品と比較して、簡単なクエリでは同等、複雑なクエリでは数倍以上の性能である。

4 TX1を使ったソリューション例

大規模なXMLデータに対しても高速な検索が可能になったため、次のような様々な業種や業務への適用が可能となる。

4.1 大量XMLデータの管理(既にXML文書が大量にあるケース)

既にXMLフォーマットで配布されている特許文書、ニュース記事、規定集などはデータ量が大きいため、従来のXMLDBでは運用が難しかった。TX1を使用することで、大規模データでも高速かつ効率的に検索を行うことが可能になる。

4.2 非定型で構造の変更が多いデータの管理(XMLDBの効果が発揮されるケース)

電子カタログ、電子申請、ウェブコンテンツ管理などコンテンツごとに格納すべき項目が異なる場合や、項目の追加及び削除が比較的多く発生する場合、RDBでは、スキーマの変更とそれに伴うアプリケーションの改変作業に手間がかかる。XMLDBを使うことにより、これらを大幅に軽減することが可能になる。XMLDBにTX1を使用すれば、システム運用後にデータ量が増えても、性能に影響を与えることがない。

4.3 異なるデータ形式をXMLで統合管理(XMLのメリットが発揮されるケース)

XMLの高い記述力により、企業内で独立に運用されている複数システムの異種DBに散在する各種データを統合することができる。

また、XMLDBを利用することで、XMLによって統合された複数システムのデータを統合的に管理し、共通項目によって横ぐしで検索するシステムを容易に構築することができる。既存システムからデータを取り出してXMLDBに格納すれば、既存システムに影響を与えることもない。

更に、XMLDBにTX1を使用すれば、統合する既存システムが増えたり、管理するデータ量が増えたりしても、性能に影響がない。

5 あとがき

XMLは、その高い記述性により、今後もデータ交換やデータ蓄積のフォーマットなど、より広く普及していくと考えられる。そのため、情報システムのプラットフォームのコンポーネントとして、XMLDBのニーズは急速に高まっている。

従来のXMLDBは小規模データでしか性能が出なかったが、今回、大規模なXMLデータでも高速に検索できるネイティブなXMLDBのTX1を開発、製品化したことにより、今までできなかった新たなソリューションが実現可能になった。

文献

- (1) W3C.XQuery 1.0: An XML Query Language.
<<http://www.w3.org/TR/xquery/>>, (参照2005-04-25).
- (2) CWI.XMark - An XML Benchmark Project.
<<http://www.xml-benchmark.org/>>, (参照2005-04-25).
- (3) 服部雅一, ほか. 高性能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.



谷川 均 TANIGAWA Hitoshi

東芝ソリューション(株) プラットフォームソリューション事業部 ソフトウェア開発部主任。XMLデータベース及びナレッジマネジメントの開発に従事。
Toshiba Solutions Corp.



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主任研究員。XMLデータベース及びナレッジマネジメントの研究・開発に従事。
Knowledge Media Lab.



松井 浩二 MATSUI Koji

東芝ソリューション(株) プラットフォームソリューション事業部 ソフトウェア開発部参事。XMLデータベース及びナレッジマネジメントの開発に従事。
Toshiba Solutions Corp.



佐藤 実 SATO Minoru

東芝ソリューション(株) プラットフォームソリューション事業部 プラットフォームソリューション第四部参事。コンテンツ管理ソフトウェアの開発及びサービス業務に従事。
Toshiba Solutions Corp.