Vícerozměrný přístup pro indexování XML dat

Michal Krátký[1]

Katedra informatiky, VŠB – Technická univerzity Ostrava
17. listopadu 15, 708 33 Ostrava–Poruba
Česká republika
michal.kratky@vsb.cz

Abstrakt. Značkovací jazyk *XML* (*Extensible Markup Language*) je v současné době chápán jako nový přístup pro reprezentaci dat. Slovy databázové technologie je XML *jazyk pro modelování dat. Správně strukturovaný* (*well-formed*) XML dokument nebo množina dokumentů je XML databáze a příslušné DTD jejím schématem. Implementace systémů vhodných pro efektivní uložení a dotazování XML dokumentů (tzv. *nativní XML databáze*) vyžaduje vývoj nových technik a je dnes jednou z klíčových otázek světa informačních technologií. V tomto příspěvku je popsán *vícerozměrný přístup pro indexování XML dat.* Tento přístup využívá vícerozměrných perzistentních datových struktur, jako je např. *UB-strom.* V článku je shrnuta dosavadní práce doktoranda.

Klíčová slova: indexování XML dat, vícerozměrné datové struktury

1 Úvod

Značkovací jazyk *XML* (*Extensible Markup Language*) [30] je v současné době chápán jako nový přístup pro reprezentaci dat. Slovy databázové technologie je XML *jazyk pro modelování dat* [28]. *Správně strukturovaný* (*well-formed*) XML dokument nebo množina dokumentů je XML databáze a příslušné DTD jejím schématem. Implementace systémů vhodných pro efektivní uložení a dotazování XML dokumentů (tzv. *nativní XML databáze*) vyžaduje vývoj nových technik [28, 26] a je dnes jednou z klíčových otázek světa informačních technologií.

XML dokument je obyčejně modelován jako graf, ve kterém jsou korespondující uzly abstraktní objekty a hrany jsou označeny názvy elementů. Nejčastěji je tento graf stromem (tzv. *XML strom*). Pro získání dat z XML databáze byly vyvinuty různé dotazovací jazyky, např. *XPath* [32] a *XQuery* [31]. Společným rysem těchto jazyků je použití regulárních výrazů pro vyjádření cesty grafem. Kde cesta je sekvence názvů elementů (nebo atributů) od kořenového elementu k listovému. Uživatel se potom v XML dokumentu naviguje pomocí různě dlouhé cesty v XML stromu vyjádřené regulárním výrazem.

Pro efektivní uložení a dotazování XML dat není možné použít existující databázové modely (ať již relační, objektový nebo objektově–relační). Při vykonávání dotazu daného regulárním výrazem cesty je nutné procházet XML

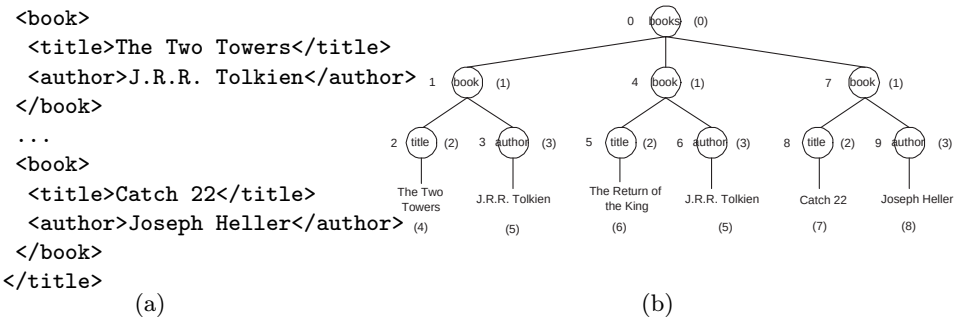
[1] Školitel doc. RNDr. Václav Snášel, CSc. (vaclav.snasel@vsb.cz).

stromem. V tomto případě konvenční přístupy jako například SQL nebo OQL selhávají nebo nejsou příliš efektivní. V současné době existuje několik přístupů pro indexování XML dokumentů nebo obecně semistrukturovaných dat. Tyto přístupy je možné rozdělit do zhruba tří skupin: *přístupy založené na relační dekompozici* (využívají databázovou technologii), *přístupy založené na trie reprezentaci XML dokumentu* a *vícerozměrné přístupy*. Úplnější přehled přístupů pro indexování XML dat najdeme např. v [1]. Pro posouzení kvality přístupů po indexování XML dat musíme mít k dispozici sadu testovacích dokumentů a dotazů - tzv. *XML benchmark* (např. [29]).

Tento příspěvek je shrnutím dosavadní práce doktoranda. V kapitole 2 je krátce popsán vícerozměrný přístup pro indexování XML dat. V závěru je shrnut obsah článku a ukázány možnosti budoucí práce.

2 Vícerozměrný přístup pro indexování XML dat

Vícerozměrný přístup pro indexování XML dat byl uveden v [6] a dále rozvíjen v [7, 8]. Tento přístup využívá pro indexování XML dat vícerozměrné datové struktury, jako je např. *UB* [25] nebo *BUB-strom* [27]. UB-strom je perzistentní, stránkovaný strom, proto jej můžeme použít pro indexování velkého množství velkých XML dokumentů, a tedy i pro implementaci nativní XML databáze.



Obr. 1. Ukázka XML dokumentu a příslušného XML stromu s jedinečnými čísly termů (názvy elementů a jejich hodnoty) – čísla v závorkách – a elementů.

Vícerozměrný přístup rozkládá XML strom na všechny cesty od kořene ke všem listům. V [7] byly uvedeny pojmy *vícerozměrný bod reprezentující obsah cesty* a *cestu*. Takto získané body jsou pak vkládány do vícerozměrné datové struktury a XML dotazy jsou prováděny pomocí dotazů nad touto datovou strukturou. Vektorové vícerozměrné datové struktury umožňují efektivně provádět *bodové* a *rozsahové dotazy* [25]. Bodovým dotazem zjistíme, zda je vektor v datové struktuře přítomen. Takto můžeme získat neindexovaná data. Rozsahový dotaz hledá všechny vektory v definovaném vícerozměrném hyperkvádru. V [9, 20] jsme publikovali nový algoritmus rozsahových dotazů pro datovou strukturu UB resp. BUB-strom.

Nyní uvedeme revizi vícerozměrného přístupu, která navíc obsahuje *index termů*. Tento index slouží pro uložení a efektivní dotazování všech řetězců v XML dokumentu, jako jsou názvy a hodnoty elementů a atributů. Na obrázku 1(a) vidíme ukázku XML dokumentu obsahujícího informace o knihách a jejich autorech. XML strom tohoto dokumentu vidíme na obrázku 1(b). V revidovaném přístupu jsou definovány tři indexy:

– **index termů (term index)**. Na obrázku 1(b) vidíme XML strom ukázkového dokumentu s *jedinečnými čísly* (*id*) termů v závorkách. Vidíme tedy že název elementu `books` má id 0. Pro uložení termů můžeme využít přístupu indexování termů publikovaného v [11, 2, 3, 14]. Tento přístup umožní klást dotazy na částečnou shodu definovanou regulárními výrazy, jako např. `books/book [title="*computer*"]` (najde všechny knihy o počítačích).

– **index cest (path index)**. V indexu cest jsou uloženy body reprezentující cesty a tak tento index slouží i pro uložení struktury dokumentů.

Na obrázku 1 vidíme, že dokument obsahuje dvě jedinečné cesty `books/book/title` a `books/book/author`. Pomocí id názvu elementů vytvoříme vektory `(0,1,2)` resp. `(0,1,3)`, které vložíme do vícerozměrné struktury s id 0 resp. 1.

– **index struktury (structure index)**. V indexu struktury jsou uloženy body reprezentující obsah cesty. Tento vektor se skládá z jedinečného čísla příslušné cesty, jedinečných čísel jednotlivých elementů cesty a jedinečného čísla hodnoty posledního elementu cesty (list XML stromu). Tento index tedy zachycuje samotnou strukturu dokumentu.

Na obrázku 1(b) vidíme jedinečná čísla elementů. Vezmě si např. cestu k hodnotě `The Two Towers`. Jedná se o cestu `book/book/title` s id 0 (viz index cest). Po vložení jedinečného čísla cesty, jedinečných čísel elementů a id termu `The Two Towers` získáme vektor (0,1,2,4), který je vložen do vícerozměrné datové struktury.

Takovýmto způsobem je zpracován celý XML dokument. Zpracování dotazů potom probíhá ve třech fázích. Vezměme např. dotaz `books/book[author="Joseph Heller"]` (chceme tedy získat všechny knihy napsané tímto autorem):

– **nalezení id termů dotazu v indexu termů**. V tomto případě hledáme id termů `books`, `book`, `author` a `Joseph Heller`.

– **nalezení id cest dotazu v indexu cest**. V datové struktuře hledáme jedinečné číslo cesty `books/book/author`, která byla transformována na bod reprezentující cestu. Id cesty 1 tedy získáme bodovým dotazem $(0, 1, 3)$.

– **nalezení vektorů v indexu struktury**. Vytvoříme dva vektory definující hyperkvádr, kterým hledáme vektory odpovídající dotazu. V tomto případě se jedná o vektory $(1, 0, 0, 8)$ a $(1, max_d, max_d, 8)$. Kde v první souřadnici se nachází id příslušné cesty nalezené v předchozím kroku, v poslední pak id termu `Joseph Heller`. Jelikož hledáme vektory s libovolnými hodnotami ve 2. a 3. souřadnici, první bod obsahuje nejmenší hodnoty domén vektorového prostoru a druhý pak největší hodnoty.

Je zřejmé, že některé XPath dotazy musí být vykonány více rozsahovými dotazy. Experimentální výsledky algoritmu rozsahových dotazů byly publikovány např. v [9, 14].

2.1 Výsledky experimentů

Pro testy byl použit XML dokument obsahující databázi bílkovin z XML projektu University of Washington [29]. Velikost dokumentu je 683MB, obsahuje 21 305 818 elementů a 1 290 647 atributů. Jako vícerozměrná datová struktura byl použit BUB-les [14], který slouží pro indexování vektorů různých dimenzí, jako jsou právě vektory reprezentující cesty a obsahy cest. Celková velikost všech indexů byla 1214MB. V tomto případě je index struktury tvořen dvěma BUB-stromy indexujícími zvlášť vektorové prostory dimenze 7 a 9.

Byly testovány dva XPath dotazy:

```
ProteinDatabase/ProteinEntry/[protein/name='hypothetical protein YDL110c']
ProteinDatabase/ProteinEntry/[reference/refinfo/authors/author='Smith, E.L.']
```

První dotaz byl proveden v BUB-stromu indexujícím vektorový prostor dimenze 7, výsledek obsahoval 1 element (1 vektor charakterizující obsah cesty). Druhý dotaz pak v BUB-stromu indexujícím vektorový prostor dimenze 9, výsledek obsahoval 33 elementů. Efektivita byla měřena[2] především počtem prohledávaných listových uzlů a počtem diskových přístupů (DAC) (viz následující tabulka). Hodnoty v závorkách znamenají kolik procent z celkového počtu listových uzlů muselo být prohledáno.

	Relevantních listových uzlů	Prohledávaných listových uzlů	DAC	Čas procesu [s]
Dotaz 1	1	116 (0.05%)	324	0.125
Dotaz 2	15	6 327 (2.3%)	15 275	5.8
Průměr	8	2 147.7 (0.9%)	7 799.5	2.9

Vidíme tedy, že pro tyto dotazy bylo prohledáno průměrně pouze 0.9% indexu. V [4] byla zkoumána struktura reálných XML dokumentů. Ukázalo se, že průměrná a maximální hloubka nabývá poměrně malých hodnot (typicky do 10) a proto efektivita vícerozměrného přístupu není degradována tzv. *prokletím dimensionality* [25].

3 Závěr

V tomto příspěvku byl prezentován vícerozměrný přístup pro indexování XML dat. Jelikož přístup využívá robustní perzistentní datovou strukturu BUB-strom, je možné jej použít pro implementaci nativní XML databáze. V budoucí práci se chceme zaměřit na vylepšení vícerozměrné datové struktury BUB-tree, tak aby rozsahové dotazy byly prováděny efektivněji. Rovněž chceme implementovat některý z dotazovacích XML jazyků (např. XPath), popř. jeho podmnožinu. Ukazuje se, že pro indexování tzv. *document-centric* XML dokumentů, je vhodné použít kombinaci stávajícího přístupu pro indexování XML dat a přístupů pro indexování nestrukturovaných dat [15, 21].

[2] Testováno na počítači Intel Pentium ®4 2.4GHz, 512MB DDR333, Windows XP.

Udělené ceny a granty

Udělena podpora pořadatelů pro účast na 29th International Conference on VLDB 2003, Berlín, Německo.

Člen řešitelských týmů grantů

GAČR 201/00/1031, *Inteligentní vyhledávání v dokumentografických informačních systémech*. Hlavní řešitel: prof. RNDr. Jaroslav Pokorný, CSc., Univerzita Karlova v Praze

GAČR 201/03/0912, *Vyhledávání a indexování XML dokumentů*. Hlavní řešitel: prof. RNDr. Jaroslav Pokorný, CSc., Univerzita Karlova v Praze

GAČR 201/03/1318, *Inteligentní analýza obsahu a struktury WWW*. Hlavní řešitel: Ing. Vojtěch Svátek, Ph.D., Vysoká škola ekonomická v Praze

Reference

1. D. Barashev, M. Krátký, and T. Skopal: *Modern Approaches to Indexing XML Data*. Sborník vědeckých prací VŠB-Technická univerzita Ostrava 2003, accepted.
2. J. Dvorský, M. Krátký, T. Skopal, and V. Snášel: Benchmarking the Multidimensional Approach for Term Indexing. *In Proceedings of DATESO 2003*, Desná–Černá Říčka, ISBN 80–248–0330-5.
3. J. Dvorský, M. Krátký, T. Skopal, and V. Snášel: Term Indexing in Information Retrieval Systems. *In Proceedings of CIC 2003*, Las Vegas, USA, 2003.
4. J. Kosek, M. Krátký, and V. Snášel: Struktura reálných XML dokumentů a metody indexování. *Accepted at ITAT 2003*, High Tatras, Slovakia.
5. M. Krátký: Distribuovaný systém pro práci s prostorovými daty a jejich vizualizaci na WWW v prostředí CORBA. *In Proceedings of GIS 2002*. Ostrava, ISSN 1213-239X.
6. M. Krátký, J. Pokorný, and V. Snášel: Indexing XML data with UB-trees. *In Proceedings of Advances in Databases and Information Systems, ADBIS 2002*, Bratislava, Slovakia, ISBN 80-227-1744-4, 2002.
7. M. Krátký, J. Pokorný, T. Skopal, and V. Snášel: The Geometric Framework for Exact and Similarity Querying XML Data. *Proceedings of First EurAsian Conferences, EurAsia-ICT 2002*, Shiraz, Iran, Springer–Verlag, LNCS 2510, 2002.
8. M. Krátký, J. Pokorný, T. Skopal, and V. Snášel: The Geometric Approach for Indexing XML Data. *In Proceedings of DATAKON 2002*, Brno, ISBN 80-210-2958-7, 2002.
9. M. Krátký and T. Skopal: Benchmarking the UB-tree. *In Proceedings of DATESO 2003*, Desná–Černá Říčka, ISBN 80–248–0330-5.
10. M. Krátký, T. Skopal, and V. Snášel: Geometrické indexování a dotazování multimediálních dat. *In Proceedings of DATAKON 2002*, Brno, 2002.
11. M. Krátký, T. Skopal, and V. Snášel: Vícerozměrný přístup pro netriviální vyhledávání termů. *In Proceedings of Znalosti 2003*, Ostrava, ISBN 80-248-0229-5.
12. M. Krátký, T. Skopal, and V. Snášel: Efektivní vyhledávání v kolekcích obrázků tváří. *In Proceedings of DATAKON 2003*, Brno.
13. M. Krátký, T. Skopal, and V. Snášel: Image Compression Using Space-Filling Curves. *Accepted at ITAT 2003*, High Tatras, Slovakia.
14. M. Krátký, T. Skopal, and V. Snášel: Multidimensional Term Indexing for Efficient Processing of Complex Queries. *Kybernetika*, Journal of the Academy of Sciences of the Czech Republic, 2003, accepted.

15. P. Moravec, M. Krátký, and V. Snášel: Random Projections for Dimension Reduction in Information Retrieval Systems. *Accepted at IMAMM'03 Conference.*
16. T. Skopal, M. Krátký, and V. Snášel: Porovnání některých metod pro vyhledávání a indexování multimediálních dat. *In Proceedings of Kybernetika 2002.* Žilina.
17. T. Skopal, M. Krátký, and V. Snášel: Properties of Space Filling Curves and Usage with UB-trees. *In Proceedings of ITAT 2002*, Brdo, High Fatra, Slovakia.
18. T. Skopal, M. Krátký, and V. Snášel: Efektivní implementace vektorového modelu pro dokumentografické informační systémy. *In Proceedings of DATAKON 2003*, Brno.
19. T. Skopal, J. Pokorný, M. Krátký, and V. Snášel: Revisiting M-tree Building Principles. *In Proceedings of ADBIS 2003.* Dresden, Germany, Springer–Verlag, LNCS 2798, 2003.
20. T. Skopal, M. Krátký, J. Pokorný, and V. Snášel: A New Range Query Algorithm for the Universal B-trees. *Submitted at EDBT 2004.*
21. T. Skopal, P. Moravec, J. Pokorný, M. Krátký, and V. Snášel: Efficient Implementation of Vector Model in Information Retrieval. *Accepted at RCDL 2003*, St. Petersburg, Russia.
22. T.Skopal, V.Snášel, and M.Krátký: Image Recognition Using Finite Automata.*In Proceedings of Prague Stringology Conference'02*, Prague, ISBN 80–01–02616–7.
23. T. Skopal, V. Snášel, M. Krátký, and V. Svátek: Searching Internet Using Topological Analysis of Web pages. *In Proceedings of CIC 2003*, Las Vegas, USA.
24. V. Snášel, D. Ďuráková, and M. Krátký: Navigation through Query Result Using Concept Lattice. *In Proceedings of DATESO 2002*, Desná-Černá Říčka.

25. R. Bayer: The Universal B-Tree for multidimensional indexing: General Concepts. *In Proceedings of WWCA'97*, Tsukuba, Japan, 1997.
26. R. Bourret: *XML and Databases.* 2001, `http://www.rpbourret.com/xml/XMLAndDatabases.htm`.
27. R. Fenk: The BUB-Tree *In Proceedings of 28rd VLDB International Conference on VLDB.* Hongkong, China, 2002.
28. Jaroslav Pokorný: *XML: a challenge for databases?*. Chap. 13 In: Contemporary Trends in Systems Development. Kluwer Academic Publishers, Boston, 2001.
29. University of Washington's database group: *The XML Data Repository.* 2002, `http://www.cs.washington.edu/research/xmldatasets/`.
30. W3 Consortium: *Extensible Markup Language (XML) 1.0.* 1998, `http://www.w3.org/TR/REC-xml`.
31. W3 Consortium: *XQuery 1.0: An XML Query Language, W3C Working Draft.* 15 November 2002, `http://www.w3.org/TR/xquery/`.
32. W3 Consortium: *XML Path Language (XPath) Version 2.0, W3C Working Draft.* 15 November 2002, `http://www.w3.org/TR/xpath20/`.

Annotation.

Multidimensional Approach for Indexing XML Data

Using the terminology usual in databases, it is possible to view XML as a language for data modelling. To retrieve XML data from XML databases, several query languages have been proposed. The common feature of such languages is the use of regular path expressions. This paper describes a multidimensional approach for indexing and querying XML data. We use the UB and BUB-tree for indexing the vector spaces.