

Dariusz BERNATOWICZ

Politechnika Koszalińska, Wydział Elektroniki i Informatyki

PORÓWNANIE WYDAJNOŚCI REALIZACJI TRANSAKCJI W NATYWNYCH BAZACH DANYCH XML

Streszczenie. Celem poniższego artykułu jest porównanie wydajności składowania i przetwarzania dokumentów XML w natywnych bazach danych XML, obsługujących transakcje. Przedstawiono w nim charakterystykę i wymagania dla natywnej bazy danych XML. Zaprezentowano metodologię testowania i wyniki uzyskanych testów.

Słowa kluczowe: natywne bazy danych XML, transakcje, XQuery, Sedna, DB2 Express

PERFORMANCE COMPARISON OF TRANSACTIONS PROCESSING IN NATIVE XML DATABASE

Summary. The purpose of this article is to show the results of the comparison an efficiency of storage and processing of XML documents in a transaction native XML databases. This paper presents a characteristic and requirements for the native XML database. Moreover, it describes the methodology of testing and also contains results of those tests and conclusions.

Keywords: native DB XML, transactions, XQuery, Sedna, DB2 Express

1. Wprowadzenie

Rosnąca popularność języka XML (ang. Extensible Markup Language) [12] oraz szeroki zakres jego zastosowań, stworzyły potrzebę efektywnego składowania i zarządzania danymi XML. Powstały więc rozwiązania dla relacyjnych systemów baz danych, wspierających składowanie danych XML, oraz systemy natywne (dedykowane), realizujące składowanie dokumentów XML. W artykule skoncentrowano się głównie na przechowywaniu i przetwarzaniu dokumentów XML w natywnych bazach danych, obsługujących transakcje oraz opartych na

logicznym modelu danych, dedykowanym dla mających drzewiastą strukturę dokumentów XML, np. DOM [4] lub XQuery [13].

Dokument XML możemy uznać za poprawnie sformułowany, jeżeli spełnia definicję dokumentu etykietowego oraz jest zgodny z regułami poprawności sformułowania, podanymi w specyfikacji XML 1.0 [12]. Ze względu na strukturę, dokumenty XML możemy podzielić na [4]:

- dokumenty zorientowane na dane (ang. Data-centric XML documents), które charakteryzują się regularną strukturą i równomierną granulacją. Tworzone są najczęściej w przypadku wykorzystywania XML jako formatu transportu danych, np. w relacyjnej bazie danych. Kolejność elementów w takim dokumencie nie jest istotna,
- dokumenty zorientowane na dokument (ang. Document-centric XML documents), które charakteryzują się nieregularną strukturą i zróżnicowaną granulacją. Tworzone są najczęściej nieautomatycznie lub w wyniku transformacji z innego formatu tekstu. W tego typu dokumentach kolejność elementów ma znaczenie.

Rodzaj struktury dokumentu jest jednym z czynników, który ma wpływ na wybór sposobu składowania XML. Wyróżnić można dwa typy składowania informacji związanej z XML, a mianowicie składowanie danych XML oraz składowanie dokumentów XML [2, 7].

Jeżeli XML jest tylko formatem wymiany danych i nie przenosi ze sobą żadnych dodatkowych informacji, a jego znaczniki nie mają istotnego znaczenia, może zostać łatwo i najczęściej efektywnie przetransformowany na powszechnie wykorzystywaną, relacyjną strukturę bazy danych. W przypadku odwrotnym składowanie dokumentów realizuje się przez zastosowanie natywnych (dedykowanych) systemów składowania dokumentów XML.

2. Natywne bazy danych XML

Bazą danych XML nazywamy programowy system trwałych struktur danych, pozwalający na zapisanie tych danych w formacie XML [6]. Natomiast bazę dokumentów XML (ang. *XML document database*), zwaną również natywną, definiujemy jako kolekcję dokumentów XML i ich części, utrzymywaną przez system mający możliwość zarządzania i kontroli kolekcji oraz informacji reprezentowanej przez tę kolekcję [10].

Logiczny model bazy danych, realizujący składowanie dokumentów XML, musi umożliwiać zarządzanie nie tylko dokumentami, lecz również informacją zawartą w dokumentach XML. Baza danych powinna korzystać z modelu danych, który zapewni zachowanie równoważności składowanych dokumentów.

W celu realizacji oceny wydajności przetwarzania zapytań w dokumentach XML zorientowanych na dokument, skoncentrowano się na natywnych bazach danych XML, obsługujących transakcje i język zapytań XQuery. Do tego celu wybrano najbardziej popularne bazy danych,

a mianowicie Sedna i IBM DB2 Express v.9.5. Z powodu ograniczeń licencyjnych, dotyczących rozmiaru obsługiwanych dokumentów, zrezygnowano z systemu Tamino Software AG.

2.1. IBM DB2 Express v.9.5

DB2 Express jest SZBD firmy IBM, służącym do składowania i zarządzania danymi w modelu relacyjnym oraz XML. Do składowania dokumentów XML wykorzystuje technologię zwaną pureXML, opartą na hierarchicznym modelu danych, pozwalającym na przechowywanie dokumentów w kolumnach tabel przy użyciu nowego typu danych XML [3]. Hierarchiczny model danych jest natywnym i unikalnym typem w DB2, używającym wewnętrznej reprezentacji dla przetwarzania nowego typu, a transformacja do typu tekstowego odbywa się za pomocą wbudowanych funkcji. Dostęp do danych uzyskiwany jest przez wykonanie zapytań w języku SQL/XML lub XQuery. System zawiera procedury wspierające zarządzanie, walidację i modyfikację schematu plików XML oraz możliwości tworzenia indeksów dla elementów, atrybutów i wartości (węzłów tekstowych) poszczególnych dokumentów, w celu przyspieszenia wyszukiwania podzbiorów danych. Otrzymujemy więc system hybrydowy, który pozwala na integrację zaawansowanych możliwości przetwarzania modelu relacyjnego z właściwościami modelu hierarchicznego dla dokumentów XML.

2.2. Native XML Sedna

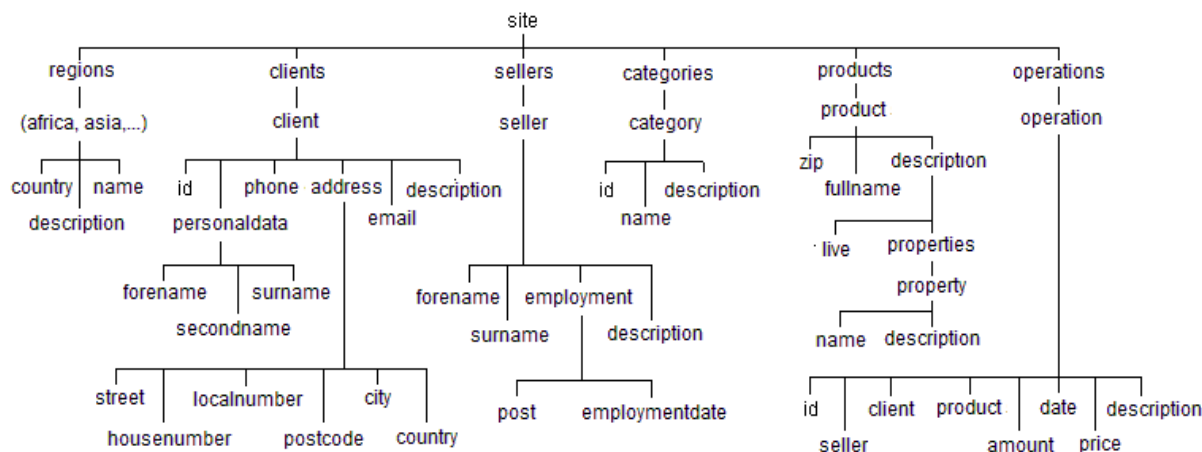
Sedna jest natywnym SZBD, zorientowanym na przetwarzanie dokumentów XML, w którym nacisk położono głównie na efektywność wykonywania zapytań i wydajność operacji aktualizacji na danych. Każdy przechowywany dokument XML dzielony jest na trzy części, a mianowicie na schemat opisowy struktury dokumentu, schemat nakazowy oraz zawartość tekstową węzłów w dokumencie. Generowany dynamicznie schemat opisowy przekształcaný jest do postaci drzewa, uzyskując w ten sposób dokładne odwzorowanie dokumentu, co pozwala na ich przechowywanie nawet w przypadku braku schematów nakazowych. Każdy węzeł schematu opisowego określony jest za pomocą etykiety zawierającej typ danego węzła oraz wskaźnika na miejsce położenia węzła na dysku. Typ węzła może wskazywać na element, atrybut lub tekst, a wskaźnik określa blok danych, zawierający listę dwukierunkową powiązanych ze sobą węzłów zgodnych z odpowiednim schematem. Bloki danych zawierają specjalne struktury, zwane deskryptorami, które posiadają wskaźniki umożliwiające określenie poprzedniego i następnego bloku danych, rodzica, rodzeństwa, zbioru wskaźników na dzieci, pola z identyfikatorem i pola zawierającego dane. Dostęp do danych uzyskiwany jest przez wykonanie zapytań w języku XQuery.

3. Metodologia realizowanych testów

Celem przeprowadzonych badań było sprawdzenie wydajności przetwarzania transakcji w natywnych bazach danych XML. Testy polegały na podziale i zrównolegleniu tego samego zbioru operacji i ich wykonaniu przez wielu klientów (wątkach). Dokonano porównania czasu wykonywania kwerend w ramach współbieżnie realizowanych transakcji w bazach danych Sedna i DB2 Express.

Wykonano testy badające czas wykonania współbieżnych transakcji w wyżej wymienionych systemach, w zależności od wielkości dokumentu XML, liczby klientów współbieżnie realizujących transakcje oraz rodzaju kwerend w transakcji.

Strukturę dokumentu testowego, ukazującego hipotetyczny sklep internetowy, przedstawiono na rys. 1. Do głównych encji dokumentu zaliczane są: region (*regions*), klienci (*clients*), sprzedawcy (*sellers*), kategorie (*categories*), produkty (*products*) oraz operacje (*operations*).



Rys. 1. Schemat dokumentu shop.xml

Fig. 1. Scheme of document shop.xml

Po zapoznaniu się z istniejącymi aplikacjami testującymi bazy danych XML, tj. XMark, XBench, MBench, MemBeR, TPoX, zdecydowano się na utworzenie własnego systemu testującego na podstawie języka Java. Na potrzeby realizacji testów utworzono aplikację generującą i wypełniającą danymi testowy dokument XML o zadanych parametrach w postaci liczby elementów w każdym węźle oraz rodzaju dodanego schematu. Wszystkie wygenerowane dokumenty charakteryzowały się równomiernym rozłożeniem elementów w poszczególnych węzłach w celu zminimalizowania wpływu struktury dokumentu na jego sposób przechowywania. Następnie, korzystając ze zdefiniowanego magazynu danych wejściowych, losowo wypełniono wygenerowany wcześniej dokument XML.

Drugim programem napisanym na potrzeby realizacji testów była aplikacja wielowątkowa, pozwalająca określić charakterystykę wydajnościową przez pomiary i analizę czasu przetwarzania dużego zbioru instrukcji w ramach transakcji. Należy ona do grupy benchmarków synte-

tycznych i umożliwia symulację działania współbieżnych klientów oraz określenie parametrów połączenia, liczby współbieżnych klientów i liczby powtórzonych pomiarów oraz na wykonanie zdefiniowanego testu składającego się z zestawu kwerend w ramach jednej lub wielu transakcji. Obliczenie czasu wykonania testu określono na podstawie różnicy czasów systemowych przed i po wykonaniu testu, czyli wykonaniu transakcji przez wszystkich klientów. W celu określenia charakterystyki wydajnościowej zastosowano dwa podstawowe wskaźniki, a mianowicie:

- przyśpieszenie wykonania programu:

$$S = \frac{t_s}{t_r}, \quad (1)$$

- efektywność zrównoleglenia:

$$E = \frac{t_s}{(t_r \cdot k)}, \quad (2)$$

gdzie: t_s – oznacza czas wykonania programu przez pojedynczego klienta, t_r – oznacza czas wykonania programu równolegle, k – określa liczbę klientów.

W celu uniknięcia wpływu połączenia z bazą danych i przekazywania zapytania do poszczególnych klientów, czynności te były realizowane przed sygnałem rozpoczęcia pomiaru czasu i wykonania transakcji. Czas synchronizacji przesyłania komunikatów między klientami a aplikacją główną, ze względu na małą wartość i znikomy wpływ w stosunku do czasu przetwarzania głównych transakcji, został pominięty.

Bazy danych testowano na tych samych dokumentach o następujących wielkościach 24,5 MB, 86 MB i 122 MB. Wszystkie testy wykonano na tym samym komputerze wyposażonym w 2 GB pamięci RAM, procesor AMD Athlon II x2 3,0 GHz i system operacyjny MS Windows 7 64bit. Wszystkie testowane systemy baz danych posiadały zaimplementowane biblioteki XML DB API, co umożliwiło bezproblemową komunikację testującej aplikacji z SZBD XML. Wszystkie zapytania przekazane do bazy danych były w postaci odpowiednio sformatowanych ciągów znakowych. Przed każdym wykonaniem testu następowało otwarcie sesji z bazą danych, a po otrzymaniu wyniku sesja była zamykana.

4. Wyniki przeprowadzonych testów

Podczas realizacji testów przetwarzania transakcji w środowisku pracy współbieżnej skupiono się głównie na dwóch rodzajach kwerend dotyczących aktualizacji i odczytu określonych danych z dokumentu XML. Wszystkie kwerendy zdefiniowano w języku XQuery, dla którego uwzględniono specyficzne różnice występujące w różnych systemach baz danych.

Przykładową kwerendę dokonującą aktualizacji nazwy produktu, określonej przez element <fullname> na podstawie zadanego numeru identyfikacyjnego elementu <zip>, przedstawiono poniżej:

a) dla systemu Sedna:

```
update
replace $p in doc("shop1")/site/products/product[zip = "P:3433366"] with
<product>
  {($p/@*, $p/node() [not (self::fullname)], <fullname>Kakao</fullname>)}
</product>
```

b) dla systemu DB2 Express:

```
xquery
transform
  copy $prod := db2-fn:sqlquery('select spdoc from shop1 where
  id = 1')/site/products
Modify do replace value of $prod/product[zip = "P:3433366"]/fullname
with "Kakao"
return $prod/product[zip = "P:3433366"]/fullname/text()
```

Kwerenda odczytu węzła produktu, w zależności od zadanej wartości kodu elementu zip dla obydwu systemów, wygląda następująco:

```
for $y in doc('shop1')/site/products/product[zip = "P:3433366"]
return $y
```

W trakcie realizacji testów badano czas wykonania zestawu instrukcji na dokumencie XML, w celu określenia przyspieszenia wykonania programu S i efektywności zrównoleglenia E. Testy przeprowadzono dla liczby klientów $L_k = \{1, 4, 8, 12\}$ i rodzaju wykonywanych operacji $O_p = \{O, A, AO, AA\}$, gdzie:

- O – określa zestaw 120 tys. instrukcji odczytu ceny losowo wybranego produktu (ograniczenie do jednej ścieżki w strukturze pliku),
- A – określa zestaw 120 tys. instrukcji aktualizacji ceny losowo wybranego produktu (ograniczenie do jednej ścieżki w strukturze pliku),
- AO – określa zestaw zawierający po 60 tys. instrukcji aktualizacji i odczytu ceny losowo wybranego produktu, realizowanych na przemian (ograniczenie do jednej ścieżki w strukturze pliku),
- AA – określa zestaw zawierający po 60 tys. instrukcji aktualizacji nazwy produktu i zmiany imienia klienta, realizowanych na przemian (wykorzystanie dwóch ścieżek w strukturze pliku).

We wszystkich pomierzonych konfiguracjach wykonano tę samą liczbę instrukcji (120 tys.), a w przypadku większej liczby klientów liczbę instrukcji podzielono równomiernie na liczbę klientów, np. 4 x 30 tys. Każdą instrukcję wykonano w oddzielnej transakcji.

Tabela 1

Czas wykonywania kwerend dla Sedna i DB2 Express [min]

Liczba klientów	Rodzaj operacji	Sedna			DB2 Express		
		24,5 MB	86MB	122 MB	24,5 MB	86 MB	122 MB
$L_k = 1$	O	2,32	4,17	6,04	3,36	4,25	5,29
	A	10,67	16,39	19,22	13,57	16,04	17,06
	AO	7,56	12,32	14,35	10,22	11,75	12,88
	AA	11,74	18,03	21,14	14,11	17,00	18,42
$L_k = 4$	O	2,16	4,06	4,21	3,24	3,51	4,36
	A	8,25	13,53	17,50	12,13	11,15	13,37
	AO	6,03	10,15	12,59	8,90	8,45	10,20
	AA	9,08	17,26	19,25	13,51	11,82	16,58
$L_k = 8$	O	2,02	3,50	4,17	2,53	3,21	4,04
	A	7,12	12,23	16,00	10,34	9,17	11,50
	AO	5,28	11,25	11,69	7,47	7,11	8,92
	AA	7,83	17,08	17,60	14,25	15,37	12,42
$L_k = 12$	O	2,08	4,05	4,14	2,18	3,29	3,12
	A	7,47	12,44	16,32	9,55	8,16	11,18
	AO	6,59	9,49	13,56	9,33	6,54	11,07
	AA	8,22	13,68	17,95	9,93	8,65	12,07

Na podstawie uzyskanych wyników, przedstawionych w tabeli 1, możemy zauważyć spadek wydajności przetwarzania kwerend przy wzroście rozmiaru dokumentu XML. Nie jest on jednakowy dla obydwu testowanych baz danych. Dla stosunkowo małych dokumentów (24,5 MB) system Sedna wykazuje o 15 – 25% krótszy czas przetwarzania transakcji, ale już dla większych dokumentów (86+ Mb) system DB2 Express zaczyna osiągać przewagę.

Tabela 2

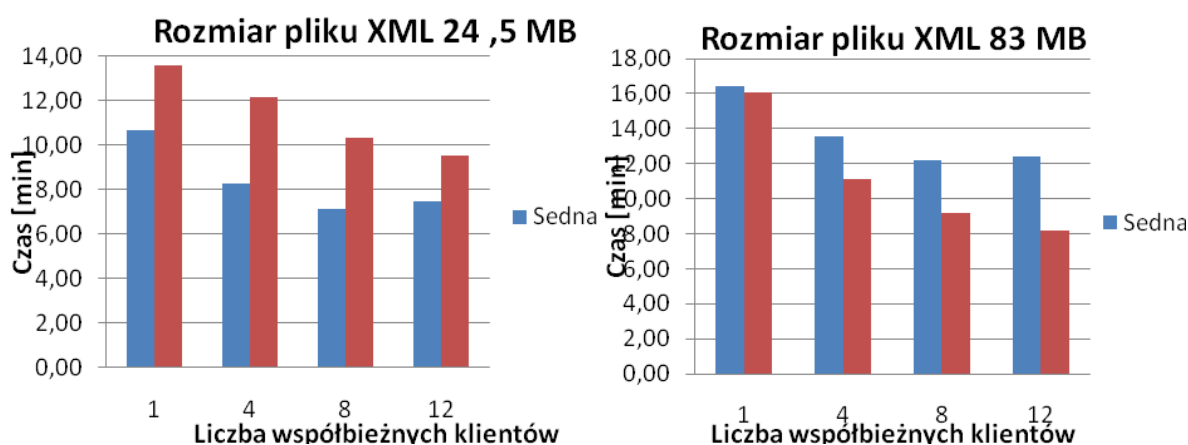
Uśrednione wartości przyspieszenia wykonania programu S i efektywności zrównoleglenia E dla różnej liczby klientów i rozmiaru badanych dokumentów XML

Rozmiar dokumentu XML	Sedna						DB2 Express					
	4		8		12		4		8		12	
	S	E	S	E	S	E	S	E	S	E	S	E
24 MB	1,27	0,32	1,42	0,18	1,30	0,11	1,16	0,29	1,38	0,17	1,56	0,13
86 MB	1,13	0,28	1,15	0,14	1,23	0,10	1,26	0,32	1,44	0,18	1,46	0,12
122 MB	1,17	0,29	1,25	0,16	1,20	0,10	1,21	0,30	1,42	0,18	1,50	0,12

Wzrost efektywności przetwarzanych transakcji uzyskujemy przez dostęp współbieżny klientów do bazy danych i jest on zależny zarówno od wielkości dokumentu, jak i liczby klientów. W tabeli 2 przedstawiono wartości przyspieszenia wykonania programu S dla różnej liczby klientów. Możemy zauważyć, że wraz ze wzrostem liczby równoległe uruchomionych klientów, przyspieszenie wykonania programu rośnie i osiąga najwyższą wartość w przedziale 7 – 9 klientów (30 – 40% dla 8 klientów). Dalsze ich zwiększanie powoduje powolny spadek wydajności, co spowodowane jest dużym stopniem zrównoleglenia i oczekiwaniem na zwolnienie blokad w przypadku operacji aktualizacji.

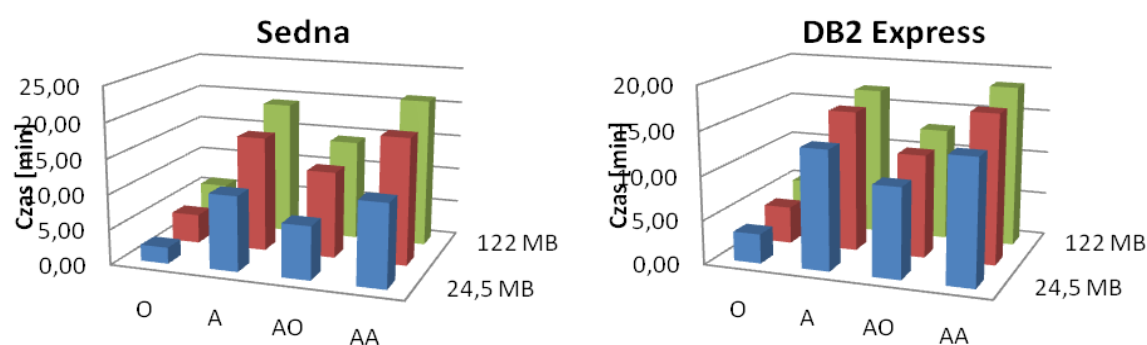
Natomiast przy wzroście rozmiaru pliku przyśpieszenie wykonania programu maleje, a dynamika spadku zależy od zastosowanego SZBD. Dla bazy Sedna zwiększenie rozmiaru dokumentu z 24,5 MB do 122 MB spowodowało 10 – 15% spadek przyśpieszenia wykonania programu, natomiast dla DB2 Express spadek ten jest już minimalny i wynosi do kilku procent. Znaczącą różnicę, około 20 – 30%, otrzymujemy przy zwiększeniu liczby klientów z jednego do czterech, natomiast dalsze zwiększanie liczby klientów nie powoduje już takiego przyrostu wydajności.

Podsumowując, system DB2 Express wypada znacznie lepiej podczas przetwarzania współbieżnego dla większej liczby klientów (8 – 12) oraz przy dużych rozmiarach dokumentów, natomiast w systemie Sedna mniejszy czas przetwarzania uzyskujemy maksymalnie dla ośmiu współbieżnych klientów, a po dalszym ich wzroście system zaczyna wykazywać spadek wydajności.



Rys. 2. Czas wykonania operacji aktualizacji „A” w bazach danych Sedna i DB2 Express, w zależności od uruchomionej liczby współbieżnych klientów

Fig. 2. The time of performed update of statement „A” in Sedna and DB2 Express database according to number of activated concurrency clients



Rys. 3. Czas wykonania operacji aktualizacji i odczytu dla różnych rozmiarów dokumentów XML w bazach danych Sedna i DB2 Express

Fig. 3. The time of performed update and read of statements for different size of XML documents in Sedna and DB2 Express database

5. Wnioski

W pracy przedstawiono porównanie wydajności przetwarzania dokumentów XML w ramach transakcji realizowanych w natywnych bazach danych XML. Przedstawiono w nim charakterystykę i wymagania dla natywnych baz. Zaprezentowano metodologię testowania i wyniki uzyskanych testów wraz z wyciągniętymi z nich wnioskami. Kwerendy testowe napisane zostały w języku XQuery i wykonane na dokumentach o różnym rozmiarze, umieszczonych w bazach danych Sedna i DB2 Express.

W ramach dalszych prac można przebadąć wydajność przetwarzania transakcji w zależności od poziomów izolacji, specyfiki stosowanych kwerend, zwiększenia rozmiaru i zmiany struktury dokumentu oraz wpływu indeksów na czas wykonania transakcji.

BIBLIOGRAFIA

1. Bernstein P., Hadzilacos V., Goodman N.: *Concurrency Control and Recovery in Database Systems*. Addison Wesley, Massachusetts 1987.
2. Bourret R.: XML Database Products, <http://rpbourret.com/xml/XMLAndDatabases.htm>.
3. Chen W., Goutev D., Ahuja R.: *DB2 9 pureXML Guide*. IBM, 2007.
4. Dom Working Group: *Document Object Model (DOM) Level 1 Specification*. ver. 1.0, 1998.
5. Fonichev A., Grinev M., Kuznetsov S.: *Descriptive Schema Driven XML Storage*. Institute for System Programming of Russia Academy of Science.
6. Graves M.: *Projektowanie baz danych XML*. Wydawnictwo Helion, Gliwice 2002.
7. Herder T., Reuter A.: *Principles of Transaction-Oriented Database Recovery*. ACM Computing Surveys, 1983.
8. Java Architecture for XML Binding (JAXB). Final Release of this Specification, JSR-222, <http://jcp.org/aboutJava/communityprocess/final/jsr222/index.html>.
9. Ozsu M.T., Valduriez P.: *Principles of Distributed Database Systems*. Second Edition, Prentice Hall, 1999.
10. Quin L.: *Open Source XML Database Toolkit*. John Willey & Sons, Inc, USA 2000.
11. Salminen A., Tompa F. W.: *Requirements for XML Document Database Systems*. ACM Symposium on Document engineering, 2001.
12. XML Core Working Group: *Extensible Markup Language (XML) 1.0 (Fifth Edition)*. W3C Recommendation 26 Nov 2008, <http://www.w3.org/TR/2008/REC-xml-20081126/>.

13. XML Query Working Group and XSL Working Group: XQuery 1.0 and XPath 2.0 Data Model. W3C Working Draft 29 Oct 2004,
<http://www.w3.org/TR/2004/WD-xpath-datamodel-20041029/>.

Recenzenci: Dr inż. Bożena Małysiak-Mrozek
Dr inż. Dariusz Mrozek

Wpłynęło do Redakcji 31 stycznia 2011 r.

Abstract

A growth of the popularity of the XML language and its wide range of applications has made a necessity of efficient storage and management of XML data. Therefore, solutions for relational database systems were formed to support storage of XML data. For the same reason, native systems were created to realize storage of XML documents. Generally, in this paper the author focused on XML document processing and storing in a databases, which process transactions. Such XML documents are based on the logical data model dedicated for tree-structure document for example DOM or XQuery.

Realization of the transaction must synchronize operations in the sufficient degree to ensure its transaction isolation. One of the ways to obtain this effect is the usage of serial processing of transactions. Due to the fact that such solution is often non-efficient, concurrent processing should be aimed at. This processing should be characterized by serial equivalence, in which the result of concurrent processing is the same as in the case of serial processing of the transaction.

The main goal of this paper is to show results of the comparison between storage and processing efficiency of XML documents in a transaction XML database. This paper presents a characteristic and requirements for the native XML databases. Moreover, it describes the methodology of testing, and it also contains results of those tests and conclusions. These tests were made for chosen type of queries defined in XQuery language. This research considers some parameters like XML document size, usage of XML schema and number of queries in a transaction.

Adres

Dariusz BERNATOWICZ, Politechnika Koszalińska, Wydział Elektroniki i Informatyki,
ul. Śniadeckich 2, 75-453 Koszalin, Polska, dariusz.bernatowicz@tu.koszalin.pl.