

## 14 Benchmarking von XML-Datenbanksystemen

Timo Böhme, Erhard Rahm

### Kurzfassung

*Zum Vergleich der Leistungsfähigkeit von XML-Datenbanksystemen sind Benchmarks erforderlich, welche den Spezifika der XML-Datenverarbeitung Rechnung tragen. Das Kapitel beschreibt die wesentlichen Anforderungen an geeignete Benchmarks. Ferner werden drei konkrete Benchmarks vorgestellt und miteinander verglichen: XMach-1, Xmark und XOO7.*

### 14.1 Einführung

Wie in den vorangegangenen Kapiteln dargelegt, gibt es eine Reihe unterschiedlicher XML-Datenbanksysteme. Der Bedarf dafür resultiert zum einen aus der zunehmenden Verbreitung von XML als Datenaustauschformat, u.a. in zahlreichen E-Business-Anwendungen. Zum anderen speichern immer mehr Anwendungs- und Systemprogramme ihre Daten in XML ab. Dazu gehören Konfigurationsdateien, Beschreibungsdokumente u.a. Die Verwaltung dieser Daten in einem Dateisystem reicht für viele Anwendungen nicht aus, so dass der Bedarf an XML-Datenbanksystemen sich ständig erhöht.

Die verfügbaren XML-Datenbanksysteme weisen erhebliche Unterschiede bezüglich der zugrunde liegenden Architektur, Funktionalität und interner Realisierung auf. Bei der Auswahl eines Systems für eine bestimmte Anwendung sind diese Aspekte und die daraus resultierende Leistungsfähigkeit (Performance) wesentliche Kriterien. Zur Leistungsbewertung von Computersystemen und Datenbanksystemen existieren zahlreiche Benchmarks. Ziel von XML-Datenbankbenchmarks ist es, in analoger Weise die Leistungsfähigkeit (Performance) unterschiedlicher XML-Datenbanksysteme umfassend und realitätsnah zu bewerten und einen Leistungsvergleich zwischen einzelnen Systemen zu ermöglichen.

Eine Reihe von Arbeiten zur Speicherung von XML in Datenbanken enthalten Geschwindigkeitsmessungen auf Basis selbst definierter Benchmarks [FIKo99] [FIKM00]. Diesen Messungen ist gemein, dass sie nicht detailliert beschrieben sind, wenige spezifische, auf die jeweilige Arbeit zugeschnittene Operationen enthalten und sich damit nicht zum allgemeinen Vergleich eignen. Dem wachsenden Bedarf nach wohldefinierten Benchmarks für XML-DBS tragen drei im Jahr 2001 spezifizierte Benchmarks Rechnung, die in diesem Kapitel beschrieben und gegen-

über gestellt werden sollen. Informationen zu einem erst kürzlich veröffentlichten Benchmark mit Namen »Michigan Benchmark« finden sich in [RPJA02].

Im Folgenden untersuchen wir nach einer allgemeinen Betrachtung der Eigenschaften von Benchmarks die Anforderungen an XML-spezifische Benchmarks. Im Anschluss daran werden die drei XML-Benchmarks in der Reihenfolge ihrer Veröffentlichung beschrieben: XMach-1, Xmark und XOO7. Die Beschreibung erfolgt in einheitlicher Weise und berücksichtigt die jeweilige Anwendungsdomäne, Datenmerkmale, Operationen sowie die Metriken. Danach erfolgt ein Vergleich zwischen den Benchmarks, und abschließend werden Beobachtungen zu bisher vorliegenden Ergebnissen mit dem Benchmark XMach-1 dargelegt.

## 14.2 Allgemeine Richtlinien für Benchmarks

Die Definition eines Benchmarks resultiert aus dem Bedarf nach einem Werkzeug zum objektiven Leistungsvergleich von Systemen bezüglich eines bestimmten Anwendungsbereichs. Ein aussagekräftiger Benchmark zur generellen Bestimmung der Leistungsfähigkeit von Computersystemen für alle Anwendungsgebiete ist nicht möglich, da je nach Anwendung stark unterschiedliche Anforderungen bestehen und einzelne Hardware- und Software-Komponenten unterschiedlich stark beansprucht werden. Aus diesem Grund existiert eine Vielzahl von Benchmarks, die spezifische Eigenschaften der Computersysteme oder einzelner Komponenten testen, wie z.B. SPEC CPU2000<sup>1</sup> zur Ermittlung der Integer- und Fließkommaleistung von Prozessoren oder BAPCO<sup>2</sup> zur Bestimmung der Leistungsfähigkeit bei der Ausführung von Bürosoftware. Auch im Datenbank-Umfeld können nicht alle Anwendungsbereiche durch einen Benchmark abgedeckt werden. Aus diesem Grund hat u.a. das Herstellergremium TPC (Transaction Processing Performance Council) verschiedene spezialisierte Benchmarks entwickelt, z.B. TPC-C zur Leistungsbewertung klassischer OLTP-Anwendungen (Online Transaction Processing), TPC-H für komplexe Decision-Support-Anfragen und TPC-W für den Einsatz relationaler Datenbanken im Web-Umfeld (E-Commerce)<sup>3</sup>. Es ist also festzuhalten, dass ein »guter« Benchmark auf einen spezifischen Aufgabenbereich (Domäne) fokussiert ist.

Wie in [Gray93] ausgeführt, sollte ein Benchmark neben der Fokussierung auf eine Domäne noch folgende Eigenschaften besitzen:

- **Relevanz:** Der Benchmark sollte die für den Anwendungsbereich wesentlichen Merkmale und Operationen abdecken, um aussagekräftige Ergebnisse zu erzielen.
- **Portabilität:** Der Benchmark sollte einfach auf allen relevanten Rechner-Plattformen implementiert werden können.

---

1. <http://www.specbench.org>  
2. <http://www.bapco.com>  
3. <http://www.tpc.org>

- *Skalierbarkeit*: Der Benchmark sollte für unterschiedlich große Konfigurationen eine Leistungsbewertung ermöglichen, von einem einzelnen PC bis hin zu Großrechner-Konfigurationen, für zentralisierte und verteilte Architekturen sowie für unterschiedlich große Datenmengen. Eine solche Skalierbarkeit ist auch wesentlich, um den Benchmark über einen längeren Zeitraum bei starken Verbesserungen in der CPU-Geschwindigkeit, Speichergrößen etc. als Maßstab einsetzen zu können. Damit die Ergebnisse vergleichbar bleiben, sind geeignete Skalierungsfaktoren zu berücksichtigen.
- *Einfachheit*: Damit ein Benchmark akzeptiert wird, muss seine Architektur und Funktionsweise so einfach sein, dass er leicht verständlich, nachvollziehbar und implementierbar ist. Durch diese Eigenschaft sind die Ergebnisse überprüfbar, wodurch sich ihre Glaubwürdigkeit erhöht.

Neben der Leistungsfähigkeit spielt auch die Bewertung der Kosteneffektivität bei der Auswahl eines Systems eine große Rolle. In TPC-Benchmarks wird die Kosteneffektivität durch einen Quotienten aus den Gesamtkosten einer Konfiguration und dem erzielten Leistungswert (z.B. Transaktionen pro Sekunde) bestimmt.

### 14.3 Betrachtungen zu XML-Datenbankbenchmarks

Nachdem im vorangegangenen Abschnitt allgemeine Kriterien für Benchmarks diskutiert wurden, geht es jetzt um die speziellen Anforderungen und Probleme von XML-Datenbankbenchmarks. Zunächst wird die Bedeutung der Domänen im XML-Umfeld betrachtet. Anschließend werden Kriterien für den XML-Datenbankbereich besprochen, die den Benchmark grundlegend bestimmen. Am Ende dieses Abschnitts wird auf Problembereiche, die sich aus dem aktuellen Entwicklungsstand von XML ergeben, eingegangen.

#### 14.3.1 Domänen für XML-Datenverwaltung

Ebenso wie es keinen universellen Datenbankbenchmark gibt, kann es auch für die XML-Datenverwaltung keinen alle Anwendungsbereiche abdeckenden, universellen Benchmark geben. Aufgrund des breiten Einsatzspektrums von XML und der hohen Bandbreite unterschiedlicher XML-Daten und den damit verbundenen Operationen gilt dies hier sogar in verstärktem Maße. XML-Datenbankbenchmarks sind daher auf einen bestimmten Anwendungsschwerpunkt auszurichten. Für diese Domänen-Fokussierung wesentlich sind vor allem die Art der XML-Daten (dokumentorientiert, datenorientiert oder Mischformen) sowie die jeweils benötigten Operationen. Im Folgenden diskutieren wir die für die Benchmarks relevanten Merkmale / Anforderungen, die sich aus den drei wesentlichen Arten von XML-Daten ergeben.

### Dokumentenkollektionen

Ein wesentliches Anwendungsgebiet für XML-Datenbanken ist die Verwaltung von *dokumentorientierten* XML-Daten (vgl. Kapitel 5) im Rahmen von Dokumentenkollektionen, wie sie in Firmen-Intranets oder im Internet auftreten. Die Elementhierarchie in den Dokumenten ist typischerweise bis zu 10 Stufen geschichtet, und die Dokumentgrößen reichen von einigen Kilobyte bis zu wenigen Megabyte.

Für diese Domäne sind folgende Operationstypen relevant:

- Hierarchische und sequenzielle Navigation
- Volltextsuche auf Elementinhalten
- Abrufen von vollständigen Dokumenten
- Extraktion von Dokumentfragmenten
- Erstellung von Verzeichnissen ausgewählter Elemente (z.B. Inhaltsverzeichnis)
- Einfügen und Löschen von Dokumenten
- Einfügen und Löschen von Dokumentfragmenten in einem Dokument

Aufgrund der unregelmäßigen und heterogenen Strukturen der Dokumente treten bei den Anfrageoperationen typischerweise auch Wildcards in den Pfadausdrücken auf.

### Strukturierte Daten

Im Gegensatz zur vorangegangenen Domäne sind *datenorientierte* oder strukturierte XML-Daten zu verwalten, wie sie u.a. in zahlreichen E-Business-Anwendungen dominieren. Die Elementhierarchie ist typischerweise flach und beträgt selten mehr als fünf Stufen. Da die XML-Datenobjekte (Dokumente) vergleichbar mit den Sätzen relationaler Datenbanken sind, liegt ihre Größe meist im Bereich von einigen hundert Byte bis zu wenigen Kilobyte.

Bei den Operationen ergeben sich im Gegensatz zur vorher beschriebenen Domäne folgende Schwerpunkte:

- Suche auf Attributwerten
- Anfragen mit Sortierung, Aggregation und Verbundanweisungen
- Erzeugung neuer Ergebnisdokumente basierend auf Anfragedaten
- Einfügen und Löschen von Dokumenten
- Änderung von Attributwerten

Die XML-Daten sind in der Regel gemäß einem oder wenigen vordefinierten Schemata strukturiert. Damit besteht bei der Definition der Anfragepfade kaum Bedarf für Wildcards, wodurch die Anfragen ein höheres Maß an Selektivität aufweisen. In Abhängigkeit von der verwendeten Schemasprache und den von der Datenbank beim Speichern unterstützten Datentypen sind Datentyptransformationen (Casting) bei der Anfrageverarbeitung notwendig. Diese aufwändigen Umwandlungen sollten in eigenen Operationstypen untergebracht werden, damit die Ergebnisse der anderen Operationstypen keine Verzerrung erfahren.

### Gemischt strukturierte Daten

Die beiden beschriebenen Ansätze stellen Extreme beim Einsatz von XML dar – dokumentorientiert vs. datenorientiert. Viele Anwendungsgebiete erfordern jedoch Mischformen von XML-Daten. Zum Beispiel sind in Online-Katalogen neben strukturierten Produktdaten auch textuelle bzw. multimediale Zusatzinformationen zu verwalten.

Für solche Anwendungsbereiche kann die Kombination der verschiedenen Eigenschaften von XML-Daten unterschiedlich realisiert werden. Eine Möglichkeit besteht darin, dass die Elementhierarchie der XML-Daten durch ein Schema wohldefiniert ist, wobei es jedoch einzelne Elemente gibt, deren Inhalt nicht beschränkt ist und die damit die dokumentorientierten Daten aufnehmen können. In einer anderen Variante werden datenorientierte und dokumentorientierte XML-Daten in separaten Dokumenten/Objekten gespeichert und durch Verweise wird die Verbindung zwischen Dokumenten der beiden Klassen hergestellt.

Die für diese Domäne relevanten Operationstypen sind ebenfalls eine Mischung der Operationen für datenorientierte und dokumentorientierte XML-Daten. Ein Benchmark muss dabei das Verhältnis der Operationstypen entsprechend dem simulierten Anwendungsfall wählen. Eine Übernahme aller Operationstypen würde zu einem sehr umfangreichen Benchmark führen, der das Kriterium der Einfachheit verletzt. Zudem sind nicht alle Operationen für eine Domäne gleichermaßen bedeutsam.

#### 14.3.2 Spezifische Kriterien für einen XML-Datenbankbenchmark

Neben der Festlegung der Domäne sind bei einem XML-Datenbankbenchmark weitere Zielsetzungen festzulegen, in denen sich die bisherigen Vorschläge auch unterscheiden:

- *Skopus: Bewertung des gesamten DBS oder einzelner Komponenten*  
Aus Sicht der Nutzer eines XML-DBS ist das Leistungsverhalten des Gesamtsystems ausschlaggebend, das sich aus dem Zusammenwirken aller Komponenten ergibt. Die Bewertung wichtiger Einzelkomponenten, insbesondere des Anfrageprozessors, kann jedoch auch schon aussagekräftige Leistungskennzahlen ergeben. Da bei der Bewertung des Anfrageprozessors mehr die Effizienz bezüglich bestimmter Operationen im Vordergrund steht, ist die gewählte Anwendungsdomäne von geringerer Bedeutung.
- *Ein- vs. Mehrbenutzerbetrieb*  
Datenbanksysteme sind generell für die gleichzeitige Nutzung durch mehrere Benutzer ausgelegt, so dass auch Benchmarks für XML-Datenbanksysteme Leistungsaussagen für den Mehrbenutzerbetrieb liefern sollten. Wenn es jedoch um die Evaluierung einzelner Komponenten geht, z.B. des Anfrageprozessors, so kann die Beschränkung auf den Einzelbenutzerbetrieb sinnvoll

sein. In diesem Fall sind nur Antwortzeitbewertungen möglich, jedoch keine Messung der Durchsatzleistung.

■ *Operationen-Mix*

Die Bestimmung der zu messenden Last erfordert die Festlegung einer bestimmten Anzahl von Operationen unterschiedlicher Art und Komplexität. Die Anzahl der Operationen sollte eher gering sein, um einen überschaubaren Vergleich der Systeme zu ermöglichen. Versucht man dennoch, möglichst alle relevanten Operationstypen abzudecken, führt dies zu relativ komplexen Operationen. Je mehr Funktionalität in einer Operation verwendet wird, um so geringer ist die Aussagekraft der Laufzeitmessung, da z.B. Rückschlüsse auf die einzelnen Anfrageprozessorkomponenten nicht mehr möglich sind. Ein optimales Verhältnis dieser beiden Einflussgrößen, Anzahl und Komplexität, kann nur in Abhängigkeit von den intendierten Zielen des Benchmarks bestimmt werden.

### 14.3.3 Problembereiche von XML-Datenbankbenchmarks

Das frühe Entwicklungsstadium der XML-Datenbanksysteme, die unterschiedlichen Architekturen und unterstützten Funktionalitäten können zu Problemen führen, die bei der Spezifikation der Benchmarks beachtet werden müssen.

Das Hauptproblem ist das Fehlen einer standardisierten Anfragesprache. Obwohl mit XQuery eine solche in der Spezifikationsphase ist, wird es noch längere Zeit dauern, bis diese durch den Großteil der XML-Datenbanksysteme unterstützt wird. Als kleinster gemeinsamer Nenner steht in den meisten Fällen XPath 1.0 für Anfrageoperationen zur Verfügung, womit jedoch nur ein Teil der benötigten Operationen umgesetzt werden kann. Auch die proprietären Anfragesprachen der verfügbaren XML-Datenbanksysteme sind nur selten mächtiger als XPath. Komplexere Operationen müssen deshalb durch API-Zugriffe auf die Daten, z.B. mittels DOM, realisiert werden. Bei der Definition eines Benchmarks ist man deshalb gezwungen, die Operationen als beschreibenden Text oder beispielhaft in der aktuellen Entwurfsfassung von XQuery anzugeben.

Gibt es für die Anfragesprache mit XPath wenigstens eine allgemein akzeptierte Grundlage, so existiert für die Datenmanipulation nichts Vergleichbares. Selbst in XQuery sind bisher keine ändernden Operationen vorgesehen. Mit XUpdate<sup>4</sup> liegt zwar ein Arbeitspapier der XML:DB-Initiative vor, jedoch wird dieser Vorschlag bisher noch von keinem am Markt befindlichen XML-DBS umgesetzt. Ein weiteres Problem mit Änderungsoperationen ist, dass einige XML-Datenbanken nicht in der Lage sind, einzelne Fragmente von XML-Daten zu manipulieren. Um Änderungen durchzuführen, müssen dort die Dokumente vollständig ausgelesen, entfernt, geändert und wieder eingefügt werden. Zur Wahrung der Portabilitätseigenschaft ist es deshalb notwendig zu entscheiden, ob der

4. <http://www.xmldb.org/xupdate/index.html>

Benchmark Änderungsoperationen enthalten soll und welchen maximalen Umfang die zu ändernden Dokumente besitzen dürfen.

Die Portabilität muss auch bei der Zusammenstellung der Datenbankanfragen beachtet werden, da sich der angebotene Funktionsumfang der einzelnen Datenbanken derzeit noch beträchtlich unterscheidet. Portabilität und Relevanz arbeiten in diesem Fall also gegenläufig. Die Spezifikation zahlreicher Operationen erfolgt auf Kosten der Portabilität; umgekehrt kann ein hochgradig portabler Benchmark nur einen kleinen Teil der relevanten Operationen enthalten. Um kein System auszuschließen, kann der Benchmark alle Operationen einzeln und unabhängig voneinander testen und besteht demzufolge aus einer Anzahl von Sub-Benchmarks. Dies führt zu zahlreichen Einzelergebnissen, welche die Vergleichbarkeit zwischen Systemen erschwert. Alternativ kann der Benchmark vorsehen, fehlende Funktionen im zu testenden System zu ergänzen (ähnlich wie beim Einsatz des Systems in der Praxis, wobei fehlende Funktionen anwendungsseitig realisiert werden müssten).

## 14.4 XMach-1<sup>5</sup>

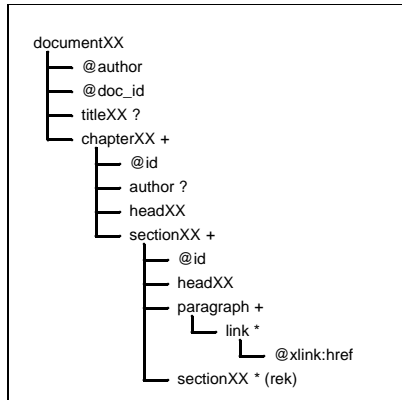
An der Universität Leipzig wurde im Jahr 2000 mit der Entwicklung von XMach-1 (XML Data Management benchmark) begonnen, dessen Spezifikation Anfang 2001 veröffentlicht wurde [BöRa01]. Es war der erste XML-Datenbankbenchmark mit Anspruch auf allgemeine Anwendbarkeit. Die wesentlichen Zielsetzungen von XMach-1 sind Skalierbarkeit, Mehrbenutzerbewertung und die Evaluierung des gesamten Datenverwaltungssystems. Im Gegensatz zu den anderen XML-Benchmarks enthält er eine genaue Beschreibung der Testumgebung sowie Vorgaben zur Ausführung der Operationen.

### Domäne

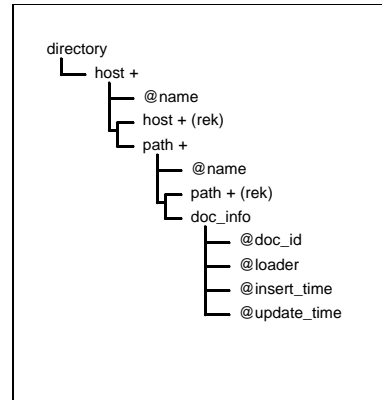
Der Benchmark simuliert eine Web-Anwendung zur Verwaltung unterschiedlicher textorientierter Dokumente. Das Testsystem kann beliebig viele Datenbank- sowie Applikations-Server umfassen, wobei letztere die generischen Anfragen der Clients in Anfragen auf die Datenbank abbilden und dabei anfallende Mapping- und Transformationsaufgaben übernehmen können. In großen Teilen entspricht dieses Szenario der in Kapitel 14.3.1 beschriebenen Domäne »Dokumentenkollektion«. Durch das Verwalten von Metadaten der Kollektion in einem strukturierten Dokument enthält der Benchmark jedoch auch Aspekte des Typs »Strukturierte Daten« und ist demzufolge der Domäne »Gemischt strukturierte Daten« mit Schwerpunkt auf dem dokumentorientierten Aspekt zuzuordnen.

---

5. Homepage: <http://dbs.uni-leipzig.de/de/projekte/XML/XmlBenchmarking.html>



**Abb. 14-1:** DOM-Struktur eines Textdokumentes in XMach-1



**Abb. 14-2:** DOM-Struktur des Metadattendokumentes

## Daten

Der Benchmark enthält zwei Dokumenttypen. Der größte Teil der Daten besteht aus Dokumenten, die in ihrer Struktur und Inhalt Textdokumenten wie Büchern, Aufsätzen o.ä. nachempfunden sind und damit vom Typ dokumentorientiert sind. In Abb. 14-1 ist die DOM-Hierarchie der Elemente und Attribute dargestellt. Diese Dokumente werden durch einen parametrisierbaren Generator synthetisch erzeugt. Um realitätsnahe Ergebnisse beim Speichern und Anfragen der Textinhalte zu erreichen, bestehen diese aus den 10.000 häufigsten englischen Wörtern, wobei ihre Verteilung der in natürlich sprachigem Text entspricht. Die Größe der Dokumente variiert zwischen 2 und über 100 Kilobyte mit einem Mittelwert von ca. 16 Kilobyte, so dass typische Dokumentgrößen ohne grafische oder multimediale Inhalte von dem Benchmark abgedeckt werden. Die Variabilität der Dokumentgröße wird durch die freie Anzahl von *chapter*- und *section*-Elementen sowie die rekursive Definition von *section* erreicht. Durch geeignete Parametrisierung bei der Generierung der Dokumente werden sowohl flache Dokumente (viele *chapter*-Elemente, wenige geschachtelte *section*-Elementen) als auch tiefe Dokumente (hohe Anzahl geschachtelter *section*-Elemente) erzeugt.

Den gemischt strukturierten Charakter erhält der Benchmark durch ein Dokument, das Metadaten über alle enthaltenen Dokumente enthält, wie z.B. URL, Name, Einfüge- und Änderungszeit. Abb. 14-2 zeigt die DOM-Struktur für dieses Dokument. In ihm sind alle Informationen in den Attributen enthalten und die Reihenfolge von Geschwisterelementen ist beliebig, so dass es vom Typ datenorientiert ist. Im Unterschied zu festen relationalen Strukturen besitzt es auch semistrukturierte Eigenschaften, die sich aus der variablen Pfadtiefe und der Optionalität von Attributen ergeben. Die Größe des Dokumentes ist abhängig von der Anzahl der Textdokumente. Jeder Eintrag benötigt ca. 160 Byte.



Der Benchmark unterstützt sowohl eine schemalose Speicherung der Dokumente als auch eine Variante mit Schemanutzung. Ein wesentliches Merkmal ist dabei die unbegrenzte Zahl von Tag-Namen, um zu testen, wie gut XML-Datenbanksysteme in der Lage sind, viele verschiedene Strukturen bzw. Schemata zu verwalten. Diesem Aspekt wird im Benchmark durch die Verwendung eines Zählers an bestimmten Tag-Namen (siehe Abb. 14-1) Rechnung getragen. Damit werden pro Schema durchschnittlich 20 Dokumente generiert.

Die Skalierbarkeit des Benchmarks bzgl. des Datenvolumens wird durch die Erhöhung der Anzahl der Dokumente in Vielfachen von 10 erreicht. Dadurch wächst auch das Metadatendokument proportional. Das Verhältnis von Dokumenten pro Schema, Anzahl von unterschiedlichen Autoren pro Dokumentanzahl u.a. wird durch die Dokumentanzahl nicht beeinflusst, so dass die Benchmarkresultate vergleichbar bleiben.

### Operationen

XMach-1 definiert acht Anfrage- und drei Änderungsoperationen, die in Tabelle 14-1 aufgelistet sind. Die Operationen reichen vom Abfragen komplexer vollständiger Dokumente (Q1) über Textsuche (Q2) bis hin zu Abfragen mit Sortier- und Verbundoperatoren, d.h., es liegt auch hier eine Mischung von dokument- und datenorientierten Anfragen vor. Die Änderungsoperationen beinhalten das Einfügen und Löschen von Textdokumenten (M1, M2) sowie die Änderung von Attributwerten des Metadatendokuments (M3). Für die Operationen liegt neben der verbalen Beschreibung auch eine Spezifikation in XQuery vor. Zur Illustration der unterschiedlichen Anfragekomplexität sind in Tabelle 14-1 für die Operationen Q3 und Q4 die XQuery-Ausdrücke angegeben. Während in Q4 die Anfrage im Wesentlichen aus einem regulären Pfadausdruck besteht, muss zur Formulierung von Q3 eine rekursive benutzerdefinierte Funktion eingesetzt werden.

Die Operationen werden gemeinsam in einem Mix ausgeführt, wobei die prozentualen Anteile der einzelnen Operationen festgelegt sind. Entsprechend der simulierten Domäne besitzt die Anfrage Q1 mit 30% das größte Gewicht. Dagegen kommt den Änderungsoperationen mit teilweise weniger als 1% nur geringe Bedeutung zu. Sie haben jedoch einen nicht unerheblichen Einfluss auf die Ausführung der Anfrageoperationen, da das Cache- und Transaktionsmanagement für die Aktualität der Daten sorgen muss.

Aufgrund der unterschiedlichen Schemata der Dokumente müssen einige Anfragen mit Wildcards arbeiten, was besondere Anforderungen an den Anfrageoptimierer stellt. Wegen der Operation M3 kann der Benchmark nur vollständig implementiert werden, wenn das Datenbanksystem die Änderung von Dokumentfragmenten erlaubt.

ID	Operation	Kommentar
Q1	Liefere das Dokument zu URL X.	Ausgabe eines vollständigen Dokumentes. Selektion des Dokumentes mittels Verbund mit Metadatendokument.
Q2	Liefere doc_id der Dokumente, die die Phrase X in einem paragraph-Element enthalten.	Testet Leistungsfähigkeit der Volltextsuche.
Q3	Navigiere beginnend beim ersten chapter-Element über jedes erste section-Element. Gib das letzte section-Element zurück.	Simuliert Navigation über den Dokumentbaum mittels Sequenzoperatoren.
	<pre> DEFINE FUNCTION deepestFirstSection(ELEMENT \$e) RETURNS ELEMENT {   IF (empty(\$e/section10)) THEN \$e   ELSE deepestFirstSection(\$e/section10[1]) } deepestFirstSection(/document10[@doc_id="d1"]/chapter10[1]/section10[1]) </pre>	
Q4	Liefere eine flache Liste der head-Elemente aller section-Elemente des durch doc_id X selektierten Dokumentes.	Rekonstruktion von Teilen des Dokumentes. Die Erstellung eines Inhaltsverzeichnis wird simuliert.
	<pre> FOR \$a IN /document10[@doc_id="d1"]/section10/head10 RETURN \$a </pre>	
Q5	Liefere alle Dokumentnamen (name-Attribut des letzten path-Elementes), die unterhalb einer gegebenen URL liegen.	Hier kann der Einfluss der Elementordnung geprüft werden.
Q6	Finde alle author-Elemente mit Inhalt X und liefere id-Attribut des Elternelementes.	Selektion über Elementinhalt.
Q7	Liefere doc_id von allen Dokumenten, die wenigstens X-mal referenziert werden.	Gruppierung und Zählerfunktionalität werden hier getestet.
Q8	Liefere doc_id von den 100 zuletzt geänderten Dokumenten, die ein author-Attribut besitzen.	Benötigt eine Reihe von datenorientierten Funktionen: Zähler, Sortierung, Verbund, Existenzoperator.
M1	Füge neues Dokument ein.	Testet die Einfügeleistung komplexer Dokumente mit aktiven Indexen.
M2	Lösche Dokument mit doc_id X.	Testet das Löschen eines komplexen Dokumentes mit aktiven Indexen.
M3	Ändere Namen und update_time-Attribut des Dokumentes mit doc_id X.	Testet die Effizienz von Änderungsoperationen auf Attributwerten.

Tab. 14-1: Operationen von XMach-1 mit ausgewählten XQuery-Beispielen

### Metriken

Als Mehrbenutzer-Benchmark steht die Messung des Durchsatzes im Vordergrund. In XMach-1 wird diese in *Xqps* (*XML queries per second*) gemessen und entspricht der Anzahl der durchgeführten Q1-Anfragen. XMach-1 sieht eine Un-

terscheidung in eine schemaunterstützte und eine schemalose Variante vor und bietet dafür zwei verschiedene Einheiten an. Neben dem Durchsatz werden auch die Antwortzeiten der einzelnen Operationen gemessen. Durch Vorgeben der maximalen Antwortzeiten für jeweils 90% der Anfragen wird erreicht, dass die Anzahl der Clients für einen maximalen Durchsatz nicht zu Lasten der Antwortzeiten optimiert werden kann. Auf der anderen Seite wird durch Denkzeiten auf der Seite der Clients erreicht, dass zur Erhöhung des Durchsatzes die Anzahl der Clients steigen muss, so dass ein Mehrbenutzerbetrieb erzwungen wird. Die Skalierung auf leistungsfähigere Systeme kann demzufolge sowohl durch die Anzahl der Dokumente als auch durch die Anzahl der Clients erfolgen.

Die Erzielung eines einzelnen Ergebniswertes pro System gestattet die Bestimmung der Kosteneffektivität (\$ bzw. Euro pro Xqps) analog zu TPC-Benchmarks. Selbstverständlich können mit dem Benchmark jedoch auch Einbenutzerbewertungen vorgenommen werden, insbesondere zu den Antwortzeiten der einzelnen Operationen. Außerdem werden Angaben bestimmt zur Ladezeit der Datenbank und zur Größe der Datenbank und von Indexstrukturen.

## 14.5 Xmark<sup>6</sup>

Dieser Benchmark wurde am National Research Institute for Mathematics and Computer Science (CWI) der Niederlande entwickelt und Mitte 2001 in der Version 1.0 veröffentlicht [SWKF01]. Hauptziel des Benchmarks ist die Untersuchung der Performance des Anfrageprozessors, welches sich in der großen Anzahl der Anfragen und in der Beschränkung auf ein lokales Computersystem (1 Rechner) widerspiegelt. Die dem Benchmark zugrunde liegenden Richtlinien werden in [SWKF01b] ausführlich diskutiert.

### Domäne

Wie schon bei XMach-1 liegt auch diesem Benchmark ein konkretes Anwendungsszenario zugrunde. Modelliert wird die Datenbank eines Internet-Auktionsportals. Diese, als ein großes XML-Dokument gestaltete Datenbank, enthält eine Anzahl von Fakten, die durch ihre feste Struktur im Wesentlichen datenorientierte Eigenschaften aufweisen. Durch die Aufnahme von Beschreibungstexten werden jedoch auch dokumentorientierte Aspekte eingebracht, d.h. auch dieser Benchmark gehört zur Domäne »Gemischt strukturierte Daten«, hier jedoch mit Schwerpunkt auf dem datenorientierten Aspekt.

### Daten

Der Benchmark verwaltet alle Daten innerhalb eines einzigen Dokumentes, dessen DOM-Struktur in Abb. 14-3 wiedergegeben ist. Es enthält die Auktionsdaten als

6. Homepage: <http://monetdb.cwi.nl/xml/index.html>

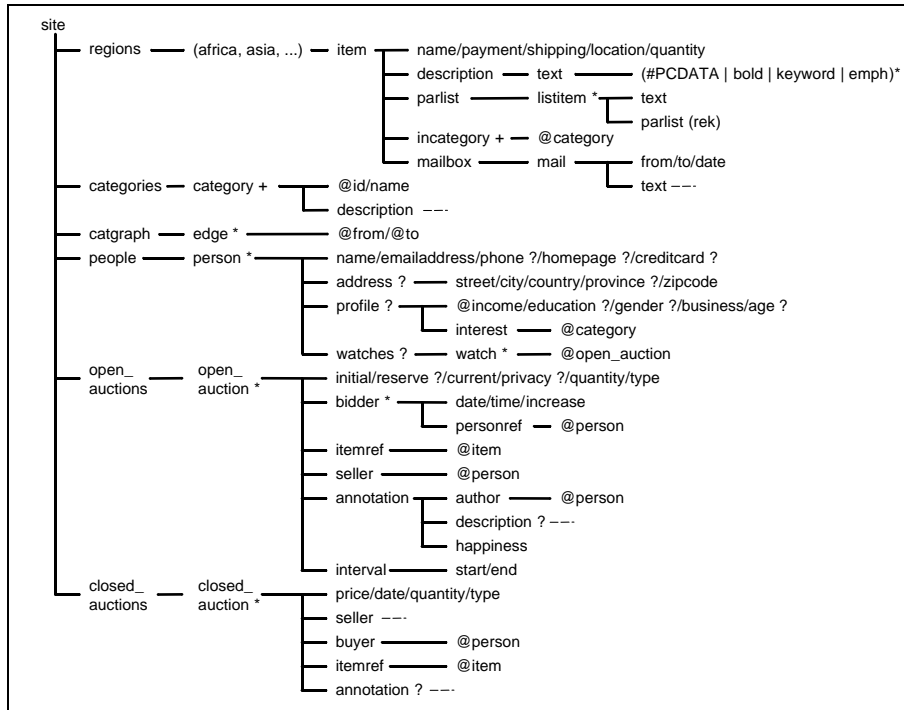


Abb. 14-3: DOM-Struktur des Xmark-Dokumentes

strukturierte Fakten, welche in den Hauptinhalten (Kindelemente des Wurzelementes) Gegenstände (*item*) – unterteilt nach Regionen –, Kategorien (*categories*), Personen (*people*), offene und geschlossene Auktionen (*open\_auctions* / *closed\_auctions*) gespeichert sind. Die sich daraus ergebenden Teilbäume sind durch zahlreiche Verweise miteinander verbunden. Die Elementreihenfolge von Geschwisterelementen ist größtenteils frei. Nur bei einzelnen Elementen, wie z.B. der Bieterhistorie (*bidder*) oder der Beschreibung der Gegenstände (*description*) ist die Reihenfolge relevant.

Die Größe des Dokumentes kann über einen Skalierungsfaktor von 10 Megabyte bis 10 Gigabyte eingestellt werden. Skaliert wird über die Anzahl von Objekten ausgewählter Mengen, z.B. Gegenstände und Personen, indem deren Basisanzahl mit dem Skalierungsfaktor multipliziert wird. Dieses hat jedoch zur Folge, dass sich das Verhältnis bestimmter Elemente zueinander ändert.

Das Dokument weist eine recht komplexe, jedoch auch feste Struktur auf. Die Tatsache, dass sich bei der Skalierung zwar die Anzahl der Elemente, jedoch nicht die Anzahl der Elementtypen erhöht, unterstreicht den stark ausgeprägten datenorientierten Skopus des Benchmarks.

## Operationen

Entsprechend seiner Ausrichtung auf den Anfrageprozessor definiert Xmark eine große Anzahl von Anfragen (20), die jeweils verschiedene Aspekte der Anfrageverarbeitung testen. Die Anfragen sind in Tabelle 14-2 aufgelistet. Neben der Formulierung der Anfragen in textueller Form liegen diese auch in XQuery-Syntax vor. Für ein besseres Verständnis der Operationsbeschreibungen sind in der Tabelle für die Operationen Q3 und Q20 auch die XQuery-Umsetzungen angegeben. Da ein entsprechender Standard zur Datenmanipulation noch nicht vorliegt, verzichtet Xmark auf die Evaluierung von Änderungsoperationen.

Die in der Tabelle angegebenen Bereiche entsprechen der Einordnung der Operationen durch die Autoren des Benchmarks. Man erkennt daran den Skopus auf die Bewertung bestimmter Aspekte der Anfrageverarbeitung. Die relativ große Zahl der Operationen resultiert auch aus der Aufnahme mehrerer funktional ähnlicher Anfragen (z.B. Verbundanfragen Q8/Q9, Q11/Q12), um die Unterstützung bestimmter Optimierungsmöglichkeiten zu bewerten. Einige Anfragen wie z.B. Q10 sind nur für kleinere Datenbestände sinnvoll. Eine alternative Einteilung der Operationen nach »dokumentorientiert« vs. »datenorientiert« ist prinzipiell möglich, jedoch gibt es Überlappungen. Anfragen, die primär zum dokumentorientierten Bereich gehören, sind z.B. Q2-Q4 (Elementreihenfolge), Q13 (Rekonstruktion, Elemente mit gemischtem Inhalt) und Q14 (Textsuche). Dem datenorientierten Bereich sind besonders die Anfragen Q1, Q11, Q12 und Q20 zuzuordnen, da diese hauptsächlich auf Attributwerten arbeiten. Die anderen Anfragen enthalten Aspekte, die für beide Bereiche relevant sind.

## Metrik

Die Ausführung der Anfragen erfolgt im Einbenutzerbetrieb, wobei jede Operation einzeln betrachtet wird. Als Resultat liefert der Benchmark die Ausführungszeiten für die einzelnen Operationen. Bei Operationen mit sehr kurzen Ausführungszeiten wird ein Wiederholungsfaktor angegeben, der bestimmt, wie oft die Operation nacheinander ausgeführt wird. In diesem Fall enthält die Ausführungszeit alle Wiederholungen. Ein Problem hierbei ist, dass durch ein von einigen Systemen vorgenommenes Caching von Anfrage-Antwort-Paaren die durchschnittliche Antwortzeit vom Ausmaß der Caching-Effekte beeinflusst wird.

Bereich	ID	Operation	Kommentar
Genauere Übereinstimmung	Q1	Liefere den Namen des Gegenstandes, registriert in Nordamerika, mit der ID 'item20748'.	Einfache Anfrage mit kompletter Pfadangabe und Zeichenkettensuche.
Elementreihenfolge	Q2	Liefere das erste Gebot für alle offenen Auktionen.	Testet Kosten für Zugriff unter Beachtung der Elementreihenfolge.
	Q3	Liefere das erste und das aktuelle Gebot aller offenen Auktionen, bei denen das aktuelle Gebot wenigstens doppelt so hoch wie das erste ist.	Komplexere Anwendung des Zugriffs auf bestimmte Elementpositionen.
		<pre>FOR \$b IN document("auction.xml")/site/open_auctions/open_auction WHERE \$b/bidder[0]/increase/text()*2 &lt;=       \$b/bidder[last()]/increase/text() RETURN &lt;increasefirst=\$b/bidder[0]/increase/text()       last=\$b/bidder[last()]/increase/text()</pre>	
	Q4	Liste die Mindestgebote aller offenen Auktionen, bei denen eine bestimmte Person A vor einer anderen Person B ein Gebot abgegeben hat.	Testet Aufwand zur Feststellung der Tag-Reihenfolge.
Casting	Q5	Wie viele verkaufte Gegenstände kosten mehr als 40?	Zur Abarbeitung der Anfrage muss der als Zeichenkette gespeicherte Preis in eine Zahl umgewandelt werden.
Reguläre Pfadausdrücke	Q6	Wie viele Gegenstände sind für alle Regionen gelistet?	Testet einfache Pfadausdrücke mit Wildcards.
	Q7	Wie viele Texte (description-, mail- und email-Elemente) gibt es?	Testet den Optimierer für Pfadausdrücke mit Wildcards. Ein kompletter Baumdurchlauf sollte nicht durchgeführt werden.
Referenzverfolgung	Q8	Liste die Namen aller Personen und die Anzahl der von ihnen gekauften Gegenstände.	Stellt mittels Verweisen einen Verbund zwischen Personen und geschlossenen Auktionen her.
	Q9	Liste die Namen aller Personen und die Namen der Gegenstände, die sie in Europa gekauft haben.	Zusätzlich zu Q8 werden noch Gegenstände über Verweise in den Verbund aufgenommen.
Komplexe Ergebnisse	Q10	Liste alle Personen nach ihren Interessen.	Erstellt umfangreich strukturiertes Ergebnisdokument.
Verbund über Attributwerte	Q11	Liste für jede Person die Anzahl der derzeit angebotenen Gegenstände, deren Preis unter 0,02% des Einkommens der Person ist.	Der Verbund über Attributwerte ist umfangreicher als bei dem Verbund mittels Referenzen.
	Q12	Liste für jede Person mit einem Einkommen über 50.000 die Anzahl der zum Verkauf angebotenen Gegenstände auf, deren Preis kleiner als 0,02% des Einkommens ist.	Ähnlich zu Q11 mit zusätzlicher Selektion der Personen. Beide Operationen bieten Optimierern viele Angriffspunkte.

Tab. 14-2: Operationen von Xmark

Bereich	ID	Operation	Kommentar
Rekonstruktion	Q13	Liste den Namen und die Beschreibung aller für Australien angebotenen Gegenstände auf.	Hier müssen größere Teile des Dokumentes wiederhergestellt werden.
Volltextsuche	Q14	Liefere die Namen aller Gegenstände, deren Beschreibung das Wort 'gold' enthält.	Kombiniert strukturelle Suche mit Volltextsuche.
Pfadnavigation	Q15	Liefere die hervorgehobenen Schlüsselwörter in Anmerkungen von geschlossenen Auktionen.	Evaluiert lange Pfadausdrücke ohne Wildcards.
	Q16	Liefere die IDs der Verkäufer von jenen geschlossenen Auktionen, die wenigstens ein hervorgehobenes Schlüsselwort in den Anmerkungen haben.	Test auf Existenz mit langen Pfadausdrücken ohne Wildcards.
Fehlende Elemente	Q17	Welche Personen haben keine Homepage?	Sucht nach fehlenden Elementen.
Funktionsaufruf	Q18	Konvertiere die Währung der Mindestgebote aller offenen Auktionen.	Testet die Ausführung benutzerdefinierter Funktionen (UDF).
Sortierung	Q19	Liefere eine alphabetisch sortierte Liste aller Gegenstände zusammen mit der Ortsangabe.	Sortierung einer großen Anzahl von Zeichenketten.
Aggregation	Q20	Gruppieren alle Kunden nach ihrem Einkommen in 4 Gruppen und gib die Kardinalität jeder Gruppe aus.	Pro Gruppe wird eine Selektion über dem income-Attribut vorgenommen. Diese Wiederholung der Operation kann für Optimierungen genutzt werden.
		<pre> &lt;result&gt;   &lt;preferred&gt;     COUNT (document("auction.xml")/site/people/person/       profile[@income &gt;= 100000])   &lt;/preferred&gt;,   &lt;standard&gt;     COUNT (document("auction.xml")/site/people/person/       profile[@income &lt; 100000 and @income &gt;= 30000])   &lt;/standard&gt;,   &lt;challenge&gt;     COUNT (document("auction.xml")/site/people/person/       profile[@income &lt; 30000])   &lt;/challenge&gt;,   &lt;na&gt;     COUNT (FOR\$p in document("auction.xml")/site/people/person       WHEREEMPTY(\$p/@income)       RETURN\$p)   &lt;/na&gt;, &lt;/result&gt; </pre>	

Tab. 14-2: Operationen von Xmark

## 14.6 XOO7<sup>7</sup>

Ebenfalls Mitte 2001 wurde der Benchmark XOO7 vorgestellt, welcher an der National University of Singapore entwickelt wurde. Im Gegensatz zu den vorangegangenen Benchmarks ist er keine komplette Neuentwicklung, sondern basiert auf dem Benchmark OO7 [CaDN93], der zur Evaluierung objektorientierter Datenbanken spezifiziert wurde. Obwohl die Autoren des Benchmarks auf Parallelen des Objektmodells und des XML-Datenmodells hinweisen, zeigt es sich, dass die Datenstruktur und auch die Operationen von OO7 nur einen Teil der Aufgaben eines XML-Datenbanksystems abdecken. Diesen Mangel versuchen die Autoren durch eine Ergänzung der Datenstruktur und zusätzliche Anfragen zu kompensieren.

### Domäne

Im Unterschied zu den beiden anderen Benchmarks liegt XOO7 kein konkretes Anwendungsszenario zugrunde. Es werden vielmehr generische Beschreibungen komplex aufgebauter Module mit im Wesentlichen »Teil-von«-Beziehungen modelliert. Dies geschieht in einer festen und regelmäßigen Struktur, bei der alle Daten in den Attributen liegen, womit das Dokument datenorientierte Eigenschaften besitzt. Durch die Aufnahme von textuellen Beschreibungen und entsprechenden Operationen darauf, erhält der Benchmark auch dokumentorientierte Aspekte und ist demnach, wie schon die vorangegangenen Benchmarks, der Domäne »Gemischt strukturierte Daten« zuzuordnen, wobei der datenorientierte Anteil dominiert.

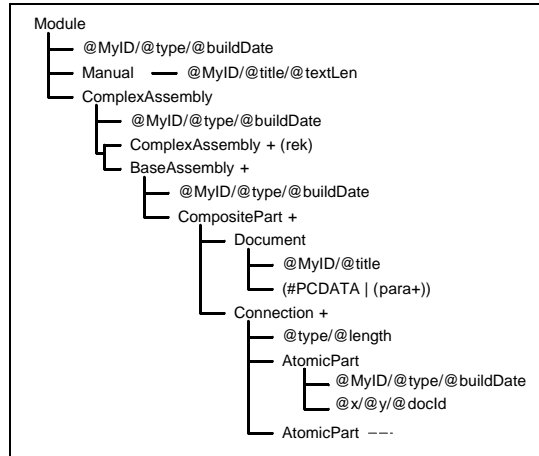
### Daten

Wie in Xmark gibt es auch bei XOO7 nur ein einziges Dokument, welches alle Daten umfasst. In Abb. 14-4 sind die Elementhierarchie sowie die zugehörigen Attribute dargestellt. Betrachtet man das Schema des zugrunde liegenden objektorientierten Benchmarks, wird dessen Ausrichtung auf Vererbung und Objektreferenzierung ersichtlich. Während das erste Konzept in XML nicht vorhanden ist, und deshalb für die Elemente die Attribute dupliziert wurden, wird auch die Möglichkeit der Referenzierung via ID und IDREF nicht genutzt, so dass die einzelnen »Objekte« redundant in den Daten vorliegen. Die enthaltenen Querverweise können somit nicht durch das Datenbanksystem verwaltet werden.

Die Daten können mittels eines Generators erzeugt werden, der durch Vorgabe mehrerer Parameter zur Rekursionstiefe und Anzahl von Kindelementen Dokumente in verschiedenen Größen erzeugen kann. Um eine Vergleichbarkeit von Ergebnissen zu ermöglichen, gibt es Vorgaben zu Parameterkonfigurationen, die als Small, Medium und Large bezeichnet sind und zu Dokumentgrößen von wenigen Megabyte bis zu einem Gigabyte reichen. Die Generierung der Zeichen-

7. Homepage: <http://www.comp.nus.edu.sg/~ebh/XOO7.html>





**Abb. 14-4:** DOM-Struktur des XOO7-Dokumentes

kettenwerte und Texte basiert dabei auf Zählern oder der Wiederholung von kurzen Zeichenketten, was die Ergebnisse beim Speichern und Anfragen dieser Werte im Vergleich zu realen Anwendungen verzerren kann.

Das den Daten zugrunde liegende Schema ist fest und regelmäßig, d.h., es gibt weder optionale Attribute oder Elemente noch freien Elementinhalt. Diese Eigenschaft, kombiniert mit der geringen Anzahl unterschiedlicher Elemente, die auch unabhängig von der Dokumentgröße ist, unterstreicht den datenorientierten Charakter des Benchmarks.

### Operationen

Da die acht Anfragen des OO7-Benchmarks nur unzureichend das typische Anfragespektrum von XML-Anwendungen abdecken, insbesondere fehlen Navigation und dokumentorientierte Operationen, wurden wichtige Operationstypen ergänzt. Insgesamt umfasst der XOO7-Benchmark 18 Anfragen, die in Tabelle 14-3 aufgeführt sind. Wie schon Xmark kennt auch XOO7 keine Operationen zur Datenmanipulation.

ID	Operation	Kommentar
Q1	Liefere AtomicPart-Elemente, deren MyID mit 5 zufällig erzeugten Werten übereinstimmt.	Einfache Selektion über numerischen Attributwert.
Q2	Liefere jeweils das erste para-Element von 5 Dokumenten, die über den Titel ausgewählt wurden.	Selektion über Zeichenkettenattribut. Elementreihenfolge wird benötigt.
Q3	Selektiere 5% der AtomicPart-Elemente mittels buildDate.	Selektion eines Bereiches über numerischen Attributwert.

**Tab. 14-3:** Operationen von XOO7

ID	Operation	Kommentar
Q4	Liefere AtomicParts-Elemente zusammen mit den zugehörigen Dokumenten.	Verbundoperation über Attributwerte.
Q5	Liefere alle Dokumente, die 2 vorgegebene Phrasen enthalten.	Volltextsuche über Document-Elemente.
Q6	Wiederhole Q1 und ersetze mehrfach auftretende Elemente durch ihre ID.	Testet Fähigkeit zum Erzeugen neuer Ergebnisstrukturen.
Q7	Zähle für jedes BaseAssembly-Element die Anzahl der Dokumente.	Testet Aggregatfunktion.
Q8	Liefere eine absteigend sortierte Liste der CompositePart-Elemente deren buildDate innerhalb eines Jahres vom aktuellen Datum aus liegt.	Testet Sortierung über Attributwerte und den Zugriff auf Umgebungsinformationen.
Q9	Liefere alle BaseAssembly-Elemente eines bestimmten Typs ohne ihre Kindelemente.	Testet Fähigkeit zur Beschränkung der Ergebnismenge auf eine Elementebene.
Q10	Liefere CompositePart-Elemente, deren buildDate-Wert nach dem eines enthaltenen BaseAssembly-Elements liegt.	Testet Effizienz des Zugriffs auf Eltern-Kind-Beziehungen.
Q11	Liefere alle BaseAssembly-Elemente eines Dokumentes, zu denen es BaseAssembly-Elemente gleichen Typs und späteren buildDate in einem anderen Dokument gibt.	Anfrage über mehrere Dokumente mit Vergleichen über Zeichenketten- und numerischen Attributen.
Q12	Liefere alle AtomicPart-Elemente mit ihren CompositePart-Elementen als Kindelemente.	Restrukturierung der Daten durch Vertauschen von Eltern-Kind-Beziehungen.
Q13	Liefere alle ComplexAssembly-Elemente eines bestimmten Typs.	Pfadausdruck mit Wildcards, da nach rekursiv definiertem Element gefragt wird.
Q14	Liefere alle BaseAssembly-Elemente, die nicht von einem gegebenen Typ sind.	Testet Robustheit bei Negation.
Q15	Liefere alle Connection-Elemente innerhalb eines CompositePart-Elementes, deren length-Wert größer als der Durchschnitt ist. Kindelemente sollen nicht mitgeliefert werden.	Gruppierungs- und Durchschnittsfunktion werden benötigt.
Q16	Liefere für ein bestimmtes CompositePart-Element die IDs des Elementes und der zugehörigen Document-Elemente in einem Result-Element zurück.	Testet Fähigkeit zur Erzeugung neuer Ergebnisstrukturen.
Q17	Liefere die Connection-Elemente, die jeweils unter den ersten 5 eines CompositePart-Elementes sind und deren length-Wert über einem vorgegebenen Wert liegt.	Benötigt Informationen zur Elementreihenfolge.
Q18	Liefere für jedes CompositePart-Element die ersten 5 Connection-Elemente, deren length-Wert über einem vorgegebenen Wert liegt.	Ähnlich zu Q18. Hiermit soll getestet werden, ob das Datenbanksystem Optimierungen durchführt.

**Tab 4-3** (Fortsetzung): Operationen von XOO7

Von den Autoren von XOO7 werden die Operationen in die Gruppen »Traditionelle Datenbankanfragen« (Q1-Q9), »Navigation« (Q10-Q16) und »Dokument-

anfragen« (Q17-Q18) eingeteilt. Allerdings spielen auch in der ersten Gruppe Navigation und dokumentorientierte Aspekte eine Rolle; umgekehrt zeigen die Anfragen der beiden anderen Gruppen ebenfalls Eigenschaften traditioneller Datenbankabfragen.

### **Metrik**

Analog zu Xmark beschränkt auch XOO7 die Testumgebung auf einen einzelnen Computer. Die Daten werden lokal gespeichert und die Anfragen im Einbenutzerbetrieb ausgeführt. Grundlegende Metrik sind demzufolge die Antwortzeiten der einzelnen Anfragen, die unabhängig voneinander ermittelt werden. In [BLLL01] wurden in einer Studie die einzelnen Operationen in die o.g. drei Gruppen aufgeteilt, jeweils 10 mal ausgeführt und pro Gruppe der Mittelwert gebildet. Diese Vermischung von Antwortzeiten unterschiedlicher Anfragen ermöglicht keine aussagekräftige Leistungsbewertung, da es starke Unterschiede zwischen Anfragetypen geben kann und die Durchschnittswerte einseitig von den zeitaufwändigen Anfragen bestimmt werden. Außerdem leidet die Vergleichbarkeit der Ergebnisse aufgrund möglicher Verzerrungen bei wiederholtem Ausführen der Anfragen.

## **14.7 Vergleich der XML-Benchmarks**

In diesem Abschnitt wird ein vergleichender Überblick über die vorgestellten Benchmarks gegeben. Dieser kann als Entscheidungshilfe für die Wahl eines geeigneten Benchmarks genutzt werden. Die hierbei diskutierten Kriterien sind als Übersicht in Tabelle 14-4 zusammengefasst.

Allen vorgestellten Benchmarks ist gemein, dass sie ein möglichst breites Spektrum der XML-Datenverarbeitung abdecken wollen, wodurch sie als Ganzes nur bedingt zur Evaluierung der zu erwartenden Leistungsfähigkeit von reinen dokument- oder datenorientierten Anwendungsszenarios geeignet sind. Trotzdem lassen sich bei den Benchmarks klare Tendenzen zu dem einen oder anderen Typ erkennen. XMach-1 mit seinen verschiedenen Schemata und großem Textanteil bedient schwerpunktmäßig den dokumentorientierten Aspekt, während Xmark und XOO7 hauptsächlich datenorientierte Eigenschaften aufweisen.

Der wesentliche Unterschied liegt jedoch im Skopus der Benchmarks. Nur XMach-1 definiert als Ziel die praxisnahe Evaluierung des gesamten Datenbanksystems im Mehrbenutzerbetrieb. Die beiden anderen Benchmarks beschränken sich auf eine ausführliche Evaluierung des Anfrageprozessors, was sich in der Zahl der Anfragen sowie der Beschränkung auf Einbenutzerbetrieb und lokalen Rechner widerspiegelt.

	<b>XMach-1</b>	<b>Xmark</b>	<b>XOO7</b>
Domäne	Gemischt strukturierte Daten	Gemischt strukturierte Daten	Gemischt strukturierte Daten
Schwerpunkt	dokumentorientiert	datenorientiert	datenorientiert
Skopus	DBMS	Anfrageprozessor	Anfrageprozessor
# Benutzer	Mehrbenutzer	Einbenutzer	Einbenutzer
# Rechner	≥1	1	1
# Schemata	# Dokumente/20	1	1
# Dokumente	10 <sup>n</sup> (n ≥ 3)	1	1
Größe der Daten	16KB*10 <sup>n</sup>	10MB-10GB	ca. 4MB-1GB
# Anfragen	8	20	18
# Änderungsoperationen	3	0	0

**Tab. 14-4:** Benchmarkvergleich

Ein weiterer Unterschied besteht in der Aufteilung der Datenbasis in Dokumente. Während XMach-1 die Datenbank auf viele kleinere Dokumente verteilt, erzeugen Xmark und XOO7 jeweils nur ein einziges Dokument, was bei XML-Datenbanken, die ein Dokument als kleinstes Anfragegranulat behandeln, zu Problemen führen kann. Weiterhin kann damit auch nicht der Einfluss verschiedener Schemata getestet werden.

Die Größe der Datenbank selbst lässt sich in allen Benchmarks über Parameter in weiten Grenzen einstellen. Die in der Tabelle angegebenen Obergrenzen resultieren aus den zur Vergleichbarkeit angegebenen Parameterkonfigurationen und nicht aus einer prinzipiellen Beschränkung der Datengeneratoren.

Als einziger der hier vorgestellten Benchmarks unterstützt XMach-1 auch Datenmanipulationsoperationen. Dies wird durch die Spezifikation der Testumgebung unter Einbeziehung eines Applikations-Servers ermöglicht, der einen Ausgleich fehlender oder nicht standardisierter Funktionalität bietet.

Zusammenfassend lässt sich sagen, dass als entscheidendes Kriterium für die Wahl eines der vorgestellten Benchmarks der Skopus betrachtet werden muss. Eine praxisnahe Bewertung des Gesamtsystems ist derzeit nur mit XMach-1 möglich. Sind jedoch Ausführungszeiten und Optimierungsstrategien für einzelne Anfragen das Ziel der Untersuchung, so sind Xmark und XOO7 die erste Wahl. Letztere sind sich in weiten Teilen sehr ähnlich, so dass ein Blick auf die reinen Parameter wenig zur Auswahl beitragen kann. Im Allgemeinen gilt, dass XOO7 mit seinem Rückgriff auf einen objektorientierten Benchmark ein sehr einfaches Datenschema mit wenigen XML-spezifischen Eigenschaften aufweist, so dass in den meisten Fällen Xmark die bessere Wahl darstellen dürfte.

## 14.8 Erfahrungen und Ergebnisse mit XMach-1

In diesem Abschnitt werden Erfahrungen, die bei der Umsetzung von XMach-1 für XML-Datenbanksysteme gesammelt wurden, sowie Ergebnisse, die einen Eindruck von der derzeitigen Leistungsfähigkeit dieser Systeme vermitteln sollen, vorgestellt.

Die ersten Systeme, für die der Benchmark implementiert wurde, waren native XML-Datenbanken, da diese alle wesentlichen Funktionen, die der Benchmark benötigte, enthielten und somit einen minimalen Anpassungsaufwand versprachen. Dem stand jedoch entgegen, dass diese Datenbanken als Neuentwicklungen noch recht jung am Markt waren und somit Einschränkungen und Fehler aufwiesen, die teilweise eine vollständige Implementierung verhinderten.

Die hauptsächliche Problemquelle bei vielen Systemen lag in der Indexierung – vor allem dem Volltextindex – und dem Mehrbenutzerbetrieb. Im Mehrbenutzerbetrieb waren sowohl Synchronisationsprobleme als auch überproportional gestiegene Antwortzeiten von Leseoperationen während paralleler Schreiboperationen zu verzeichnen. Letzteres lässt sich zum einen auf Sperrgranulate auf Dokumentenebene zurückführen, durch die die Abarbeitung der Anfragen blockiert wird, zum anderen scheinen sich die notwendigen Änderungen an den Indizes beim Einfügen bzw. Löschen von Dokumenten blockierend auf die anderen Anfragen auszuwirken. Bei einer größeren Anzahl von Anfrageprozessen (>20) zeigten einige Systeme Instabilitäten, in deren Folge es zum Abbruch der Anfrageverarbeitung kam. Weitere Problembereiche mit den ersten XML-Datenbankversionen sind in [BöRa01b] aufgeführt.

Operation	NXD 1 alte Version 90%-Antwortzeit (ms)	NXD 1 Optimierung + neue Version 90%-Antwortzeit (ms)
Q1	250	31
Q2	2684	100
Q3	120	11
Q4	80	60
Q5	30	20
Q6	2713	30
Q7	6709	20
Q8	370	11
M1	126069	9173
M2	32306	10114
M3	291	641

**Tab. 14-5:** Antwortzeitvergleich für alte und neue Version eines nativen XML-DBS

Viele dieser Probleme wurden im Laufe von neuen Versionen bereits behoben, was zu einer Steigerung der Anfrageeffizienz führte. Dies verdeutlicht auch Tabelle 14-5. Hier ist jeweils das Maximum der besten 90% der Antwortzeiten zu allen Operationen für ein natives XML-Datenbanksystem gegeben. Die teilweise drastischen Verbesserungen resultieren sowohl aus einer optimierten Implementierung als auch aus dem Einsatz einer neueren Version der Datenbank. Mit dem Versionswechsel wurden einige der Optimierungen erst möglich. Die vergleichsweise langen Antwortzeiten für die Änderungsoperationen M1 und M2 resultieren aus einer wenig effizienten Implementierung des Volltextindex, der bei beiden Operationen aktualisiert werden muss. Die Messungen wurden an einer Datenbank mit 1000 Dokumenten und einem Benutzer auf einem Intel-Pentium-III-Computer (800 MHz) mit einem Speicherausbau von 512 MB durchgeführt.

Ein großes Hemmnis ist jedoch weiterhin die beschränkte Mächtigkeit der Anfragesprachen. Während die nativen XML-Datenbanken wenigstens über vollwertige XPath-Implementierungen und einen direkten DOM-Zugriff verfügen, ist selbst diese Funktionalität bei relationalen Datenbanken mit XML-Erweiterung eingeschränkt und kann oftmals nicht auf Indexstrukturen zurückgreifen.

Die Ausführung von Anfragen über mehrere Schemata hinweg stellt für native XML-Datenbanken inzwischen kaum noch ein Hindernis dar. Ganz anders sieht es hier im relationalen Lager aus. Ein effizienter Zugriff auf die Daten kann hier meistens nur innerhalb eines Schemas erfolgen. Ein Vorteil der langjährigen Entwicklung relationaler Datenbanken wird bei der effizienten Ausführung von Volltextanfragen deutlich, die teilweise den nativen XML-Datenbanken noch Probleme bereiten. Letztere verlagern diese Funktionalität verstärkt auf Fremdprodukte, die über eine Schnittstelle an die Datenbank angebunden werden.

Die Entwicklung der XML-Datenbanken lässt sich auch an der Datenbankgröße, die für eine bestimmte Rohdatengröße benötigt wird, ablesen. In Tabelle 14-6 ist für eine native XML-Datenbank über mehrere Versionen hinweg die Datenbankgröße aufgeteilt nach Daten und Index wiedergegeben. Grundlage dafür bildete die 1000-Dokumente Konfiguration des Benchmarks, die in Rohdatenform ca. 16 Megabyte benötigt. Es ist zu erkennen, dass Daten und Index in der letzten Version weniger als die Hälfte der ursprünglichen Größe belegen. Im Durchschnitt ist für die nativen XML-Datenbanken zu beobachten, dass die Datenbankgröße ohne Index gegenüber den Rohdaten um den Faktor 1,5 bis 3 zunimmt.

	Daten (MB)	Index (MB)
NXD 2 Version 1	64,4	72,4
NXD 2 Version 2	44,5	35,6
NXD 2 Version 3	25,9	31,9

**Tab. 14-6:** Änderung der benötigten Datenbankgröße eines nativen XML-DBS

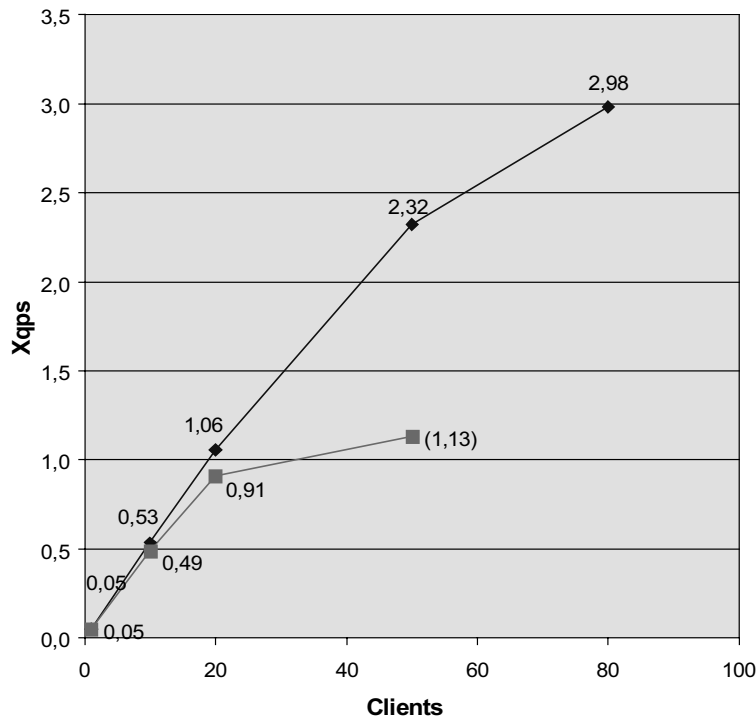


Abb. 14-5: Vergleich der Durchsatzleistung für zwei native XML-DBS (Juni 2002)

Das Laden der Datenbank mit 1000 Dokumenten kann je nach System zwischen 90 und 600 s betragen. In Kombination mit der Erstellung der Indexstrukturen reicht die Spanne von 200 bis 800 s für die Population der Datenbank. Bestimmender Faktor ist dabei die Erzeugung des Volltextindex.

Einen Eindruck von der Mehrbenutzer-Leistungsfähigkeit aktueller XML-Datenbanksysteme soll die Grafik in Abb. 14-5 vermitteln. Sie stellt die Durchsatzleistungen zweier nativer XML-DBS in Abhängigkeit von der Anzahl der Clients dar. Die Messungen erfolgten ebenfalls auf dem schon weiter oben erwähnten Intel-Pentium-Computer mit 1000 Dokumenten. Die geringe Datenmenge schließt Behinderungen für den Externspeicherzugriff weitgehend aus, so dass Durchsatzbegrenzungen primär durch CPU- und Sperrengpässe (verursacht durch die Änderungsoperationen) zu erwarten sind. Man erkennt, dass die beiden Systeme NXD 1 und NXD 3 nur einen relativ geringen Durchsatz von ca. 3 bzw. 0,9 Xqps erreichen (beim NXD 3-Wert von 1,13 wurden die Antwortzeitgrenzen überschritten). NXD 1 erzielt eine nahezu lineare Durchsatzsteigerung für bis zu 50-80 Clients, während NXD 3 schon bei 20 Clients seine Leistungsgrenze erreicht. Dieses spiegelt sich ebenfalls in den 90%-Antwortzeiten der einzelnen Operationen wider. Erreicht NXD 3 die vorgegebene 3-Sekunden-Grenze schon bei 20 Clients, ist dieses bei NXD 1 erst bei 80 Clients der Fall. Die im Benchmark nicht restringierten Antwortzeiten der Änderungsoperationen liegen auch im Mehrbenutzerbetrieb meist deutlich höher als für die Queries.

## 14.9 Zusammenfassung

Benchmarks sind ein wichtiges Instrument zum Vergleich der Leistungsfähigkeit von XML-Datenbanken. Aufgrund des breiten Anwendungsbereiches von XML muss bei der Wahl des Benchmarks die anvisierte Domäne berücksichtigt werden. Die drei vorgestellten Benchmarks berücksichtigen sowohl daten- als auch dokumentorientierte XML-Daten, jedoch mit unterschiedlicher Schwerpunktsetzung. Xmark und XOO7 konzentrieren sich im Wesentlichen auf die Bewertung des Anfrageprozessors im Einbenutzerbetrieb sowie auf eine aus einem XML-Dokument bestehende Datenbank. XMach-1 ist für die Bewertung des Gesamtleistungsverhaltens eines XML-DBS konzipiert und definiert mit Xqps (XML Queries per Second) eine Durchsatzmetrik für den Mehrbenutzerbetrieb. Die Anwendung der Benchmarks auf konkrete Systeme wird durch die derzeit noch geringe Funktionalität der Anfragesprachen erschwert. Die bisher vorliegenden Ergebnisse zeigen, dass sich die Funktionalität und Leistungsmerkmale einzelner Systeme in den letzten beiden Jahren signifikant gebessert haben. Trotzdem bestehen bei einigen XML-Datenbanksystemen immer noch Skalierungsprobleme auf sehr große Datenmengen und Mehrbenutzerfähigkeit.

## Literatur

- [BLLL01] Bressan, S.; Lee, M. L.; Li, Y. G.; Lacroix, Z.; Nambiar, U.: *The XOO7 XML Management System Benchmark*. National University of Singapore, CS Dept Technical Report TR 21/00, November 2001.
- [BöRa01] Böhme, T.; Rahm, E.: *XMach-1: A Benchmark for XML Data Management*. Proc. 9. GI-Fachtagung Datenbanksysteme in Büro, Technik und Wissenschaft, S. 264-273, Springer, Berlin, März 2001.
- [BöRa01b] Böhme, T.; Rahm, E.: *Benchmarking XML Database Systems – First Experiences*, Position Paper. Ninth International Workshop on High Performance Transaction Systems (HPTS), Pacific Grove, California, Oktober 2001.
- [CaDN93] Carey, M. J.; DeWitt, D. J.; Naughton, J. F.: *The OO7 Benchmark*. In Proc. of the ACM SIGMOD Int. Conference on Management of Data, S.12-21, Juni 1993.
- [FIKM00] Florescu, D.; Kossmann, D.; Manolescu, I.: *Integrating Keyword Search into XML Query Processing*. In: Proc. of the 9th WWW Conference, Amsterdam, Juni 2000.
- [FIKo99] Florescu, D.; Kossmann, D.: *Storing and Querying XML Data using an RDMBS*. In: IEEE Data Engineering Bulletin, Volume 22, Number 3, S. 27-34, September 1999.
- [Gray93] Gray, J. (Hrsg.): *The Benchmark Handbook for Database and Transaction Processing Systems*. Morgan Kaufmann Publishers, San Mateo, Kalifornien, 1993.
- [RPJA02] Runapongsa, K.; Patel, J. M.; Jagadish, H. V.; Al-Khalifa, S.: *The Michigan Benchmark: Towards XML Query Performance Diagnostics*. <http://www.eecs.umich.edu/db/mbench/paper.html>, Juni 2002.
- [SWKF01] Schmidt, A.; Waas, F.; Kersten, M. L.; Florescu, D.; Manolescu, I.; Carey, M. J.; Busse, R.: *The XML Benchmark Project*. Technical Report INS-R0103, CWI, Amsterdam, Niederlande, April 2001.
- [SWKF01b] Schmidt, A.; Waas, F.; Kersten, M. L.; Florescu, D.; Carey, M. J.; Manolescu, I.; Busse, R.: *Why And How To Benchmark XML Databases*. SIGMOD Record, Volume 30, Number 3, S. 27-32, September 2001