

# XML-Benchmarks

*Zheng Wei*  
[wei@rhrk.uni-kl.de](mailto:wei@rhrk.uni-kl.de)

Universität Kaiserslautern  
Fachbereich Informatik  
AG DBIS

**Seminar XML**  
**WS2002/03**

## 1. Motivation

Zum Vergleich der Leistungsfähigkeit von XML-Datenbanksystemen sind Benchmarks erforderlich, welche den Spezifika der XML-Datenverarbeitung Rechnung tragen. Das Papier beschreibt die wesentlichen Anforderungen an geeignete Benchmarks, drei konkrete Benchmarks werden vorgestellt und miteinander verglichen: XMach-1, Xmark und X007.

### 1.1 Einführung XML-DB Benchmark

Es gibt eine Reihe unterschiedlicher XML-Datenbanksysteme. Der Bedarf dafür entsteht zum einen aus der zunehmenden Verbreitung von XML als Datenaustauschformat, u.a. in e-Business Anwendungen. Zum anderen speichern immer mehr Anwendungsprogramme und Systemprogramme ihre Daten in XML ab. Deswegen erhöht sich der Bedarf an XML-Datenbanksystemen ständig.

Die verfügbaren XML-Datenbanksysteme weisen Unterschiede bezüglich der zugrunde liegenden Architektur, Funktionalität und interner Realisierung auf. Bei der Auswahl eines System für eine bestimmte Anwendung sind diese Aspekte und die daraus resultierende Leistungsfähigkeit wesentliche Kriterien. Zur Leistungsbewertung von System und Datenbanksystemen existieren zahlreiche Benchmarks, um die Leistungsfähigkeit zu überprüfen.

Benchmarks sind wichtige Werkzeuge, um die Leistung eines Systems zu analysieren und potentielle Flaschenhälse zu entdecken.

1

Ziel von XML-Datenbankbenchmarks ist es, die Leistungsfähigkeit unterschiedlicher XML-Datenbanksysteme umfassend und realitätsnah zu bewerten und einen Leistungsvergleich zwischen einzelnen Systemen zu ermöglichen.

Z.Zt gibt es allerdings noch keinen „commonly agreed“ Standard für die Definition von XML-Datenbanken also werden wir hier eine „frequently used“ Definition anwenden, d.h., ein System um XML-Dokumente zu speichern und warten.

## 1.2 Historische Studien DB-Benchmarks

Vor eine genaueren Analyse betrachten wir zunächst die Entwicklungsgeschichte von relationalen Systemen und Benchmarks, um rationale Kriterien zu identifizieren.

Es gibt schon zahlreiche Benchmarks für konventionelle relationale Datenbanken, z.B : TPC, SAP und Peoplesoft.

Frühere Benchmarks waren nur entwerfen, um Funktionalität von Datenbanken auf einer sehr allgemeinen Ebene zu überprüfen. Der Wisconsin Benchmark hat beispielsweise eine Anzahl Queries, die die Systemleistung über join-Operationen testet. Obwohl es tatsächlich wertvoll ist, die Flaschenhälse in join-Prozessen festzustellen hat dieser Benchmark jedoch keine bestimmten Anwendungsszenarios.

Mit der Entwicklung von Datenbankanwendungen wird das Szenario jedoch wichtig. Einer der bedeutendsten Industrie-Benchmarks, TPC-H/R, wurde aufgrund der allgemeinen Anforderungen einer Data-Warehousing-Anwendung implementiert.

Die neueste Entwicklung sind anwendungsspezifische Benchmarks, wie z.B. die SAP und Peoplesoft Benchmark-Serien, die die Leistungscharakteristik einer spezifischen Anwendung evaluieren. Im Vergleich mit den früheren Allzweck-Benchmarks evaluieren solche „Applications Benchmarks“ die Datenbank nicht mehr als eine isolierte Einheit, sondern als „integriertes Backend“ in ein Anwendungsszenario. Solche „Application Benchmarks“ orientieren sich auch meistens an Customer-purchase-Vorgängen.

## 1.3 Allgemeine Richtlinien für Benchmarks

(a)Relevanz

Der Benchmark muss die für den Anwendungsbereich wesentlichen Merkmale und Operationen abdecken, um aussagekräftige Ergebnisse zu erzielen.

(b)Portabilität

Der Benchmark muss einfach auf allen relevanten Rechner-Plattformen implementiert werden können.

(c)Skalierbarkeit

Der Benchmark muss 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.

(d)Einfachheit

Damit ein Benchmark akzeptiert wird, müssen seine Architektur und Funktionsweise so einfach sein, dass er leicht verständlich, nachvollziehbar und implementierbar ist. Durch diese Eigenschaften sind die Ergebnisse überprüfbar, wodurch sich ihre Glaubwürdigkeit erhöht.

## 1.4 Domänen für XML-Datenverwaltung

(a)Dokumentensammlungen

Die Verwaltung von Dokument orientierten XML-Daten im Rahmen von Dokumentensammlungen ist ein wesentliches Anwendungsgebiet. Die Elementhierarchie in den Dokumenten ist typischerweise bis zu 10 Stufen geschichtet, und die Dokumentgrößen reichen von einigen KB bis zu wenigen MB.

Für diese Domäne sind folgende Operationstypen relevant:

-Hierarchische und sequenzielle Navigation

-Volltextsuche auf Elementinhalten

-Abrufen von vollständigen-Extraktion Dokumentensammlungen von Dokumentfragmenten

-Erstellung von Verzeichnissen ausgewählter Elemente

3

-Einfügen und Löschen von Dokumenten

-Einfügen und Löschen von Dokumentfragmenten in einem Dokument

(b)Strukturierte Daten

Datenorientierte oder strukturierte XML-Daten zu verwalten, tritt auch in zahlreichen Anwendungsbereichen auf, z.B. e-Shop. Die Elementhierarchie ist typischerweise flach und beträgt selten mehr als 5 Stufen. Da die XML-Daten Objekte vergleichbar mit den Sätzen relationaler Datenbank sind, liegt ihre Größe meist im Bereich von einigen hundert B bytes bis zu wenigen KB.

Für diese Domäne sind folgende Operationstypen relevant:

-Suche auf Attribut Werten

-Anfragen mit Sortierung, Aggregation und Verbund Anweisungen

-Erzeugung neuer Ergebnisdokumente basierend auf Anfragedaten

-Einfügen und Löschen von Dokumentensammlungen

-Änderung von Attribut Werten

(c)Gemischt strukturierte Daten

Viele Anwendungsbereiche, z.B., e-Aktienmarkt, erfordern eine Mischform von XML-Daten. Für solche Anwendungsbereiche kann die Kombination der verschiedenen Eigenschaften von XML-Daten unterschiedlich realisiert werden. Eine Möglichkeit ist, dass die Elementhierarchie der XML-Daten durch ein Schema zwar wohldefiniert ist, es jedoch einzelne Elemente gibt, deren Inhalt nicht beschränkt ist und diese somit Dokument-orientierte Daten aufnehmen können. In einer anderen Variante werden datenorientierte und dokumente-orientierte XML-Daten in separaten Dokumenten/Objekten gespeichert und die Verbindung zwischen Dokumenten der beiden Klassen durch Verweise hergestellt.

Die relevanten Operationstypen sind ebenfalls eine Mischung der Operationen für datenorientierte und dokumentorientierte XML-Daten.

## 1.5 Spezifische Kriterien für einen XML-Datenbankbenchmark

### (a) Gültigkeitsbereich

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.

### (b) 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 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, aber keine Messung der Durchsatzleistung.

### (c) 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.

## 2. XMach-1

XMach-1 entstand an der Universität Leipzig im Jahr 2000, um die Leistung von XML-DBMS zu evaluieren. XMach-1 basiert auf einer Web-Anwendung und hat verschiedene Typen von XML-Daten.

### 2.1 Domäne

Der Benchmark simuliert eine Web-Anwendung zur Verwaltung unterschiedlicher textorientierter Dokumente, Das Testsystem kann beliebig viele Datenbanken umfassen und soll dabei anfallende Mapping- und Transformationsaufgaben übernehmen können. 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.

## 2.2 Daten

Der XML Datenbank enthält zwei Dokumenttypen: strukturierte Daten und Textdokumente. Das Directory enthält Metadaten auf alle Textdokumenten.

### 2.2.1 Directory document

Das Directory Document genügt folgender DTD:

```
<!ELEMENT directory (host+)>
<!ELEMENT host (host+ | path+)>
<!ATTLIST host
  name CDATA #REQUIRED>
<!ELEMENT path (path+ | doc_info)>
<!ATTLIST path
  name CDATA #REQUIRED>
<!ELEMENT doc_info EMPTY>
<!ATTLIST doc_info
  doc_id ID #REQUIRED
  loader CDATA #REQUIRED
  insert_time NMTOKEN #REQUIRED
  update_time NMTOKEN #IMPLIED>
```

Die werden beispielsweise durch dieses zwei URL <http://www.test-company.com/products/overview.xml> und <http://support.test-company.com/help.xml>

```
<directory>
  <host name="com">
    <host name="test-company">
      <host name="www">
        <path name="products">
          <path name="overview.xml">
            <doc_info doc_id="2"
              loader="robot1"
              insert_time="20000110152530"/>
          </path>
        </path>
      </host>
    <host>
      <host name="support">
        <path name="help.xml">
```

```

    <doc_info doc_id="3"
      loader="robot1"
      insert_time="20000612094430"/>
  </path>
</host>
</host>
</host>
</host>
</directory>

```

### 2.2.2 Managed documents

In Xmach-1 enthält ein Textdokument Kapitel, Abschnitte und Paragraphen. Weiterhin gibt es noch für Autoren und Querenweise Tags, um spezifische Queries auszuführen.

Mit nur einer DTD alle Dokumenten zu simulieren, ist jedoch unrealistisch für große Sammlungen von Dokumenten aus verschiedenen Quellen. Um multiple DTDs zu simulieren, kann der Generator eine Integer Zahl erzeugen und alle Tagnamen sowie „author“ und „link“ als Variante der Generic DTD eingefügt. Z.B. hat die 17. DTD als Elementnamen sowie document17, chapter17, author, section17, etc.

Die Generic DTD sieht folgende aus:

```

<!ELEMENT document (title, chapter+)>
<!ATTLIST document
  author CDATA #IMPLIED
  doc_id ID #IMPLIED>
<!ELEMENT author (#PCDATA)>
<!ELEMENT title (#PCDATA)>
<!ELEMENT chapter (author?, head, section+)>
<!ATTLIST chapter
  id ID #REQUIRED>
<!ELEMENT section (head, paragraph+, section*)>
<!ATTLIST section
  id ID #REQUIRED/>
<!ELEMENT head (#PCDATA)>
<!ELEMENT paragraph (#PCDATA | LINK)*>
<!ELEMENT link EMPTY>
<!ATTLIST linke
  xlink:type (simple) #FIXED "simple"
  xlink:href CDATA #REQUIRED>

```

## 2.3 Datenbank population und Skalierung

Die XML-Datenbank wird vor der Ausführung mit einer initialen Menge von Dokumenten des Benchmarks gefüllt. Um Skalierung zu verschiedenen Systemen zu unterstützen, gibt es vier mögliche Datenbankgrößen mit einer initialen Dokumentanzahl: 10,000 , 100,000 , 1,000,000 or 10,000,000. Durch Insert-Operationen wird die Anzahl von Dokumenten während der Ausführung des Benchmarks erhöht.

Die DB-Population besteht aus zwei Teilen: Hinzufügen generierter Dokumente zur Datenbank und entsprechender Entities ins Directory-Dokument. Der Generierungsprozess benutzt einen Zufallzahlgenerator, um Dokumente zu erzeugen.

### 2.3.1 URL Generierung

Jedes Dokument in dieser Datenbank hat eine eindeutige(unique) URL und eine Dokument-Id. Ein URL enthält 3 Hostelemente und 1 bis 4 Pfadelementen, das letzte Pfadelement ist der Dokumentname.

Die reguläre Expressionssyntax der generierten URL ist:

```
ahost{1-3}.bhost{1-(N/199)}.chost{1-5}/[apath{1-3}/[bpath{1-3}/[cpath{1-3}]]]NAME
```

*NAME=DOC\_ID+LOADER\_ID+'.xml' .*

### 2.3.2 Daten-Generierung für Text (#PCDATA)

Um realistische Leistungsaussagen über generierten Text zu bekommen, muss das „managed document“ ähnliche Eigenschaften wie ein „natural language text(NLT)“ haben. Der Generator benutzt ein Wörterbuch, der etwa 10,000 englische Alltagswörter enthält, aus denen zufällig Wörter ausgewählt werden.

### 2.3.3 Document Generierung

Der Generator benutzt einen Dokumentbaum-Algorithmus mit dem Parametern „depth“ und „size“ zum generieren:

*number of sections per document: 5-150*  
*number of paragraphs per section: 1-15*



*number of sentences per paragraph:2-30*  
*number of words per sentence:3-30*  
*probability of having an author attribute/element:0.5*  
*number of words per head or title element:2-12*  
*probability of having a phrase within a sentence:0.01*  
*probability of having a link element within a paragraph:0.05*  
*number of documents per DTD:2-100*

8

### **2.3.4 Phrase list generation**

XMach-1 enthält eine „search for phrase in document“ Funktion. Um geeignete Resultate zu bekommen, ist es jedoch nötig zu wissen, welche Phrase man suchen kann, um entsprechende „Hits“-Statistiken zu erzeugen.

Deswegen wird eine Liste von „search Phrases“ generiert, wobei jede Phrase eine 3-Wort-Tupel ist. Die Anzahl von Phrasen ist 1/10 der initialen Anzahl von Dokumenten.

## **2.4 Operationen**

Xmach-1 enthält 8 Query Typen und 3 Update Typen.

### **2.4.1 Queries**

*Q1: Get document with given URL*

*Q2: Get doc\_id and URL from documents containing a give phrase in paragraph elements*

*Q3: Start with first element which name starts with "chapter" and recursively follow first element which name starts with "section". Return each of the "section" elements*

*Q4: Return flat list of head elements which are children of elements whose names start with "section" from a document given by doc\_id*

*Q5: Get document name from all documents which are below a given URL fragment*

*Q6: Get doc\_id and id of parent element of author element with given content*

*Q7: Get doc\_id from documents which are referenced by at least four other documents*

*Q8: Get doc\_id from the last 100 updated documents haing an author attribute*

## 2.4.2 Data manipulation operations

*M1: Insert document with given URL*

*M2: Delete a document with given doc\_id*

*M3: Update URL and update\_time for a given doc\_id*

## 2.4.3 Operation mix

*Q1 <=30%*

*Q2 >=20%*

*Q3 >=10%*

*Q4 >=10%*

*Q5 >=10%*

*Q6 >=10%*

*Q7 >=4%*

*Q8 >=4%*

*M1 >=0.75%*

*M2 >=0.25%*

*M3 >=1%*

*Summe=100%*

## 2.5 Metriken

Als Mehrbenutzer-Benchmark steht die Messung des Durchsatzes im Vordergrund. In Xmach-1 wird diese in *Xqps (XML queries per second)* gemessen.

Offensichtlich muss die Antwortzeit beschränkt werden. Alle Query-Typen und die DMO (Daten Manipulation Operation) M3, also 90% aller Operationen, müssen innerhalb von 3s ausgeführt werden. Wegen des Festplattenzugriff, sind die Insert- and Delete-Operationen M1 and M2 nicht Zeit-kritisch, deswegen beträgt das Zeitlimit für Insert und Delete 20s inklusive der Ausführungszeit für Datenbank und Application Server.

Um einen echten Mehrbenutzerbetrieb mit einer realistischen Anzahl von konkurrenten Clients zu simulieren, ruft ein Browser nicht mehr als eine Operation gleichzeitig auf. Nach erfolgreicher Ausführung Operation gibt es noch eine Wartezeit zwischen 1s und 10s, bevor die nächste Operation durchgeführt wird.

Die Kosteneffektivitätsbewertung funktioniert analog zum TPC-Benchmark, gemessen wird in (\$ bzw. € pro Xqps).

10

## **2.2 Xmark**

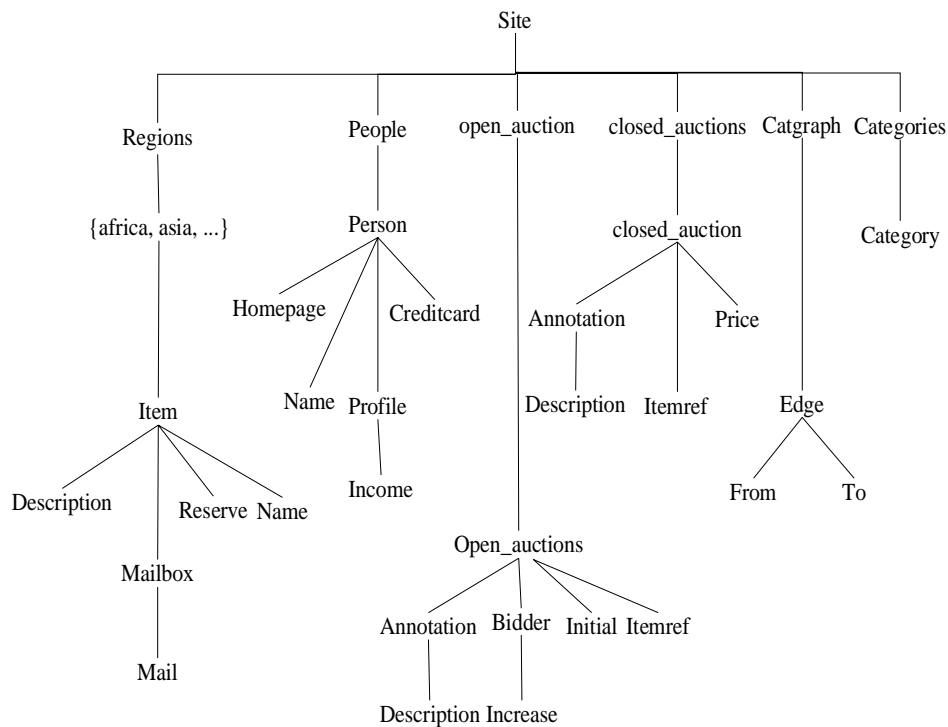
Dieser Benchmark wurde am National Research Institute for Mathematics and Computer Science (CWI) der Niederlande entwickelt und Mitte 2001 veröffentlicht.

### **2.2.1 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 wird jedoch auch der dokumentorientierte Aspekt eingebracht, somit gehört auch dieser Benchmark zur Domäne „Gemischt strukturierte Daten“, jedoch mit Schwerpunkt auf dem datenorientierten Aspekt.

### **2.2.2 Daten**

Der Benchmark verwaltet alle Daten innerhalb eines einzigen Dokumentes, dessen Element Relationships-Diagramm unten wiedergegeben ist. Es enthält die Auktionsdaten als strukturierte Fakten, welche Gegenstände in die Hauptinhalte nach Regionen, Kategorien, Personen, offene und geschlossene Auktionen unterteilt.



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

### 2.2.3 Operationen

Entsprechend seiner Ausrichtung auf den Anfrageprozessor definiert Xmark eine große Anzahl von Anfragen, die jeweils verschiedene Aspekte der Anfrageverarbeitung testen.

Genauere Übereinstimmung

*Q1: Return the name of the person with ID 'person0'*

Elementreihenfolge

*Q2: Return the initial increases of all open auctions*

*Q3: Return the first and current increases of all open auctions whose current increase is at least twice as high as the initial increase.*

*Q4: List the reserves of those open auctions where a certain person issued a bid*

*before another person.*

Casting

*Q5: How many sold items cost more than 40?*

Reguläre Pfadausdrücke

*Q6: How many items are listed on all continents?*

*Q7: How many pieces of prose are in our database?*

Referenzverfolgung

*Q8: List the names of persons and the number of items they bought.*

*Q9: List the names of persons and the number of items they bought in Europe.*

Komplexe Ergebnisse

*Q10: List all persons according to their interest; use French markup in the result.*

Verbund über Attributwerte

*Q11: For each person, list the number of items currently on sale whose price does not exceed 0.02% of the person's income. 12*

*Q12: For each person with an income of more than 50000, list the number of items currently on sale whose price does not exceed 0.02% of the person's income.*

Rekonstruktion

*Q13: List the names of items registered in Australia along with their descriptions.*

Volltextsuche

*Q14: Return the names of all items whose description contains the word 'gold'.*

Pfadnavigation

*Q15: Print the keywords in emphasis in annotations of closed auctions.*

*Q16: Return the Ids of the sellers of those auctions that have one or more keywords in emphasis.*

Fehlende Elemente

*Q17: Which person do not have a homepage?*

Funktionsaufruf

*Q18: Convert the currency of the reserves of all open auctions to another currency.*

Sortierung

*Q19: Give an alphabetically ordered list of all items along with their location.*

Aggregation

*Q20: Group customers by their income and output the cardinality of each group.*

## 2.2.4 Metrik

Die Ausführung der Anfragen erfolgt im Einbenutzerbetrieb, wobei jede Operation einzeln betrachtet wird. Als Resultat liefert der Benchmark die Ausführungszeiten fuer 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 gibt es aber ein Problem: Caching-Effekte.

## 2.3 X007

Der Mitte 2001 vorgestellte X007 Benchmark wurde an der National University of Singapore entwickelt wurde. Dieser Benchmark ist keine komplette Neuentwicklung, sondern basiert auf dem Benchmark 007, der zur Evaluierung objektorientierter Datenbanken spezifiziert wurde.

13

### 2.3.1 Domäne

Im Unterschied zu den beiden anderen Benchmarks liegt X007 kein konkretes Anwendungsszenario zugrunde. Es werden vielmehr generische Beschreibungen komplex aufgebauter Module mit im wesentlichen „Teil-von“-Beziehungen modlliert. Dies geschieht in einer festen und regelmaessigen Struktur, bei der alle Daten in den Attributen liegen, womit das Dokument datenorientierte Eigenschaften besitzt.

### 2.3.2 Daten

Wie in Xmark gibt es auch bei X007 nur ein einziges Dokument, welches alle Daten umfasst.

Der DTD von X007 wird darunter gezeigt:

```

<!DOCTYPE      Module>
<!ELEMENT      Module      {Manual, ComplexAssembly}>
<!ATTLIST      Module      MyID          NMTOKEN #REQUIRED
                        type          CDATA      #REQUIRED
                        buildDate    NMTOKEN #REQUIRED>
<!ELEMENT      Manual      (#PCDATA)>
<!ATTLIST      Manual      MyID          NMTOKEN #REQUIRED
                        type          CDATA      #REQUIRED
                        buildDate    NMTOKEN #REQUIRED>
<!ELEMENT      ComplexAssembly {ComplexAssembly+ | BaseAssembly+}>

```

```

<!--ATTLIST
    ComplexAssembly
        MyID
            type
            buildDate
        NMTOKEN #REQUIRED
        CDATA #REQUIRED
        NMTOKEN #REQUIRED>
<!--ELEMENT
    BaseAssembly
<!--ATTLIST
    BaseAssembly
        MyID
            type
            buildDate
        NMTOKEN #REQUIRED
        CDATA #REQUIRED
        NMTOKEN #REQUIRED>
<!--ELEMENT
    CompositePart {Document, Connection+}
<!--ATTLIST
    CompositePart
        MyID
            type
            buildDate
        NMTOKEN #REQUIRED
        CDATA #REQUIRED
        NMTOKEN #REQUIRED>
<!--ELEMENT
    Document #PCDATA/(para+)>
<!--ATTLIST
    Document
        myID
            type
        NMTOKEN #REQUIRED
        CDATA #REQUIRED>
<!--ELEMENT
    para (#PCDATA)>
<!--ELEMENT
    Connection (AtomicPart, AtomicPart)>
<!--ATTLIST
    Connection
        type
        length
        CDATA #REQUIRED
        NMTOKEN #REQUIRED>
<!--ELEMENT
    AtomicPart EMPTY>
<!--ATTLIST
    AtomicPart
        myID
            type
            buildDate
            x
            y
            docID
        NMTOKEN #REQUIRED
        CDATA #REQUIRED
        NMTOKEN #REQUIRED
        NMTOKEN #REQUIRED
        NMTOKEN #REQUIRED>
    
```

So wie 007 hat auch der X007 Benchmark hat drei mögliche Konfiguration:

Parameters	Small	Medium	Large
NumAtomicPerComp	20	200	200
NumConnPerAtomic	3	6	9
DocumentSize(bytes)	500	1000	1000
ManualSize(bytes)	2000	2000	4000
NumCompPerModule	50	50	500
NumAssmPerAssm	3	3	3
NumAssmLevels	5	5	7
NumComPerAssm	3	3	3
NumModules	1	1	1

### 2.3.3 Operationen

Da die acht Anfragen des 007-Benchmarks nur unzureichend das typische Anfragespektrum von XML-Anwendungen abdecken, insbesondere fehlen Navigation und dokumentorientierte Operationen, wurden wichtige Operationstypen ergänzt. Insgesamt umfasst der X007-Benchmark 18 Anfragen. Wie Xmark besitzt der X007 auch keine Operationen zur Datenmanipulation.

#### Traditionelle Datenbankabfragen

*Q1: Randomly generate 5 numbers in the range of AtomicPart's MyID. Then return the AtomicPart according to the 5 numbers.*

15

*Q2: Randomly generate 5 titles for Documents then return the first paragraph of the Document by lookup on these titles.*

*Q3: Select 5% of AtomicParts via buildDate (in a certain period).*

*Q4: Join AtomicParts and Documents on AtomicParts docId and Documents MyID.*

*Q5: Randomly generate 2 phrases among all phrases in Documents. Select those documents containing the 2 phrases.*

*Q6: Repeat query 1 but replace duplicated elements using their IDREF.*

*Q7: For each BaseAssembly count the number of documents.*

*Q8: Sort CompositePart in descending order where buildDate is within a year from current date.*

*Q9: Return all BaseAssembly of type 'type008' without any child nodes.*

#### Navigation

*Q10: Find the CompositePart if it is later than BaseAssembly it is using (comparing the buildDate attribute).*

*Q11: Select all BaseAssemblies from one XML database where it has the same 'type' attributes as the BaseAssemblies in another database but with later buildDate.*



*Q12: Select all AtomicParts with corresponding CompositeParts as their sub-elements.*

*Q13: Select all ComplexAssemblies with type 'type008'.*

*Q14: Find BaseAssembly of not type 'type008'.*

*Q15: Return all Connection elements with length greater than Avg(length) within the same composite part without child nodes.*

*Q16: For CompositePart of type 'type008', give 'Result' containing ID of CompositePart and Document.*

Dokumentanfragen

16

*Q17: Among the first 5 Connections of each CompositePart, select those with length greater than 'len'.*

*Q18: For each CompositePart, select the first 5 Connections with length greater than 'len'.*

### 2.3.4 Metrik

Analog zu Xmark beschränkt auch X007 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. 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.

## 2.4 Vergleich und Evaluation

	<i>XMach-1</i>	<i>Xmark</i>	<i>X007</i>
Domäne	Gemischt strukturierte Daten	Gemischt strukturierte Daten	Gemischt strukturierte Daten
Schwerpunkt	Dokumentorientiert	Datenorientiert	Datenorientiert
Skopus	DBMS	Anfrageprozessor	Anfrageprozessor
#Benutzer	Mehbenutzer	Einbenutzer	Einbenutzer

	<i>XMach-1</i>	<i>Xmark</i>	<i>X007</i>
#Rechner	1 oder mehr	1	1
#Schemata	#Dokumente/20	1	1
#Dokumente	Mehr als 1000	1	1
Größe der Daten	16KB*10^n (n>2)	10MB-10GB	ca. 4MB-1GB
#Anfragen	8	20	18
#Änderungsoperationen	3	0	0

Tabelle 1

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 Anwendungsszenariens geeignet sind. Trotzdem lassen sich bei den

17

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 X007 hauptsächlich datenorientierte Eigenschaften aufweisen.

<i>Id</i>	<i>Description</i>
R1	Query all data types and collections of possibly multiple XML documents.
R2	Allow data-oriented, document-oriented and mixed queries.
R3	Accepts streaming data.
R4	Supports operations on various datamodels.
R5	Allow conditions/constraints on text elements.
R6	Supports for hierarchical and sequence queries.
R7	Manipulate NULL values.
R8	Supports quantifiers (exists, for all, „non“) in queries
R9	Allow queries that combine different parts of document(s).
R10	Support for aggregation.
R11	Able to generate sorted results.
R12	Support composition of operations
R13	Allow navigation (reference traversals)
R14	Able to use environment information as part of queries e.g. current date, time etc.
R15	Able to support XML updates if data model allows
R16	Support for type coercion
R17	Preserve the structure of the documents
R18	Transform and create XML structures
R19	Support ID creation

<i>Id</i>	<i>Description</i>
R20	Structural recursion
R21	Element ordering

Tabelle 2

Der wesentliche Unterschied liegt jedoch im Geltungsbereich 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.

18

<i>Coverage</i>	<i>XMach-1</i>	<i>XMark</i>	<i>X007</i>
R1	Ja	Ja	Ja
R2	Ja	Ja	Ja
R3	Nein	Nein	Nein
R4	Ja	Ja	Ja
R5	Ja	Ja	Ja
R6	Nein	Nein	Nein
R7	Nein	Ja	Nein
R8	Nein	Ja	Ja
R9	Ja	Ja	Ja
R10	Ja	Ja	Ja
R11	Nein	Ja	Ja
R12	Nein	Nein	Nein
R13	Nein	Ja	Ja
R14	Nein	Nein	Ja
R15	Ja	Nein	Nein
R16	Nein	Nein	Nein
R17	Nein	Ja	Ja
R18	Nein	Ja	Ja
R19	Nein	Nein	Nein
R20	Nein	Nein	Ja
R21	Nein	Ja	Ja

Tabelle 3

Ein weiterer Unterschied besteht in der Aufteilung der Datenbasis in Dokumente. Während XMach-1 die Datenbank auf viele kleinere Dokumente verteilt, erzeugen Xmark und X007 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 1 angegebenen Obergrenzen resultieren aus den zur Vergleichbarkeit angegebenen Parameterkonfigurationen und nicht aus einer prinzipiellen Beschränkung der Datengeneratoren.

19

Als einziger der 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.

Wenn als entscheidendes Kriterium für die Wahl eines der drei Benchmarks der Gültungsbereich betrachtet werden muss, so ist XMach-1 ist der praxisnächste.

Sind Ausführungszeiten und Optimierungsstrategien für einzelne Anfragen das Ziel der Untersuchung sind Xmark und X007 am besten geeignet. Im Allgemeinen gilt, dass X007 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.

### 3. 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 X007 konzentrieren sich im Wesentlichen 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 Jahren signifikant gebessert haben. Trotzdem bestehen bei einigen XML-Datenbanksystemen immer noch Skalierungsprobleme auf sehr große

Datenmengen und Mehrbenutzerfähigkeit.

### Literatur

1. A. Schmidt, F.Waas, M.Kersten, Xmark: A Benchmark for XML Data Management, VLDB 2002
2. T.Böhme, E.Rahm, Xmach-1: A Benchmark for XML Data Management, <http://dbs.uni-leipzig.de>
3. S.Bressan, M. Lee, Y. Li, The X007 XML Management System Benchmark, N.U. Singapore
4. E.Rahm, G.Vossen, Web and Datenbanken, dpunkt.verlag GmbH 2003
4. W3C, XML, <http://www.w3c.org/XML>