

基于 TPC-C 的 XML 数据库测试方案

司冠南, 许 静

(南开大学信息技术科学学院, 天津 300071)

摘 要: 为实现 XML 数据库的性能评测, 提出基于 TPC-C 的 XML 数据库测试方案。针对 XML 数据库特性, 对其数据结构、查询事务语句进行定制, 将原有 9 张表映射成 5 个 XML Schema 文件, 按照 SQL/XML 标准重写负载事务。应用该方案对 SQL Server 2005 数据库进行测试, 结果表明显示的各项事务特征均与 TPC-C 基准相同。

关键词: TPC-C 测试基准; 可扩展标记语言数据库; 对象关系

XML Database Test Scheme Based on TPC-C

SI Guan-nan, XU Jing

(College of Information Technical Science, Nankai University, Tianjin 300071, China)

【Abstract】 In order to realize the performance evaluation for Extensible Markup Language(XML) database, this paper presents a XML database test scheme based on TPC-C test benchmark. Aiming at the trait of XML database, it specializes in data structure and query event sentence. The original nine tables are mapped to five new XML Schema documents, it rewrites the workload according to the SQL/XML standard and uses the scheme for testing a SQL Server 2005 database. Results show that the characteristics of various affairs equal to TPC-C test benchmark.

【Key words】 TPC-C test benchmark; Extensible Markup Language(XML) database; object relation

1 概述

XML 技术飞速发展, 在许多方面有广泛应用, 并将占有越来越重要的地位。目前, 主要使用 XML 数据库对 XML 文档进行存储和处理。正确的性能评测对开发人员了解系统瓶颈、改善系统以及用户正确选择系统有重要作用。本文研究将 TPC-C^[1]用于 XML 数据库测试, 并使用该方案对真实系统进行测试。

2 相关工作

在 IT 行业中, 基准测试被公认为是衡量计算机性能的方法和标准。目前, 许多研究机构都提出了针对 XML 数据库的基准测试, 如 XMach-1^[2]、XMark^[3]、XPathMark、XOO7^[4]、XBench^[5]、MBench、MemBeR、TPoX^[6]等, 其中, XMark、XOO7、XBench 更侧重于通过大规模的查询语句覆盖尽量多的查询功能, 以检测数据库功能方面的全面性, 并没有提出具体的性能指标。XMach-1、TPoX 着眼于性能测试, 但没有得到大规模应用的检测, 且不太完善。而如 XPathMark、MBench、MemBeR 属于小规模测试, 不适用于整个系统的评测。因此, 采用更完善的 TPC-C, 并将其特别定制, 着眼于数据库软件交互过程^[7], 以适用于 XML 数据库的测试。

3 测试方案研究

3.1 数据结构方面的定制

3.1.1 定制原则

将 TPC-C 测试基准中数据结构映射为 XML Schema, 主要在元素类型、属性和文本上进行, 分为基于表格的映射和对象-关系映射。

基于表格的映射方式是将 XML 文件看作一个表格, 将表格的字段存储为子元素或属性。这种方式虽然简单, 但无法表示表之间的引用, 割裂了数据模型中各对象之间的相互

关系。对象-关系映射方式将 XML 文件中的数据视为特定的对象树模型, 其中, 元素及其类型、元素内容或复合元素类型通常被视为类, 而简单元素类型、属性被当作简单属性。通过传统的对象-关系映射技术完成数据库表与 XML 文件的映射。这种映射方式保持了对象间关系, 同时在部分 XML 文档更新及子文件级别上的并发控制方面都有一定优势。

3.1.2 定制方法

TPC-C 的对象关系如图 1 所示。

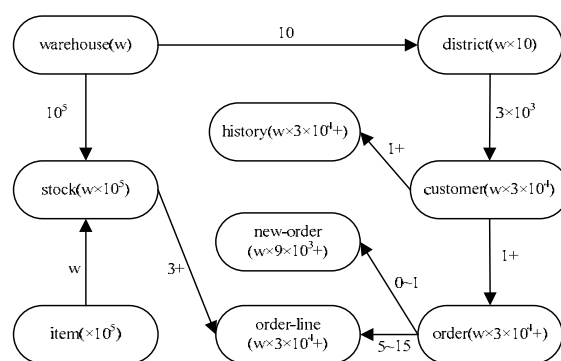


图 1 TPC-C 对象关系

本文采用对象-关系映射方式, 按照表之间的关联关系将原有的 9 张表转换为 5 个 XML Schema 文件, 其中, warehouse 与 district 之间以及 district 与 customer 之间都是一对多的关系, 因此可以将其合并为一个 XML Schema 文件, district 作

基金项目: 国家“863”计划基金资助项目(2009AA01Z152)

作者简介: 司冠南(1981-), 男, 博士研究生, 主研方向: 软件工程; 许 静, 教授、博士、博士生导师

收稿日期: 2010-04-07 **E-mail:** sign@mail.nankai.edu.cn

为 warehouse 的复杂子元素, customer 则作为 district 的复杂子元素。order 与 order-line 之间是一对多的关系,而 order 与 new-order 是一一对应的关系,而且只有订单状态为新订单时该记录才会在 new-order 表中出现,发货后就会从 new-order 中删除。因此,可将其合并为一个文件。另外,history、item、stock 作为统计性的对象实体存在,与其他对象的关系不太密切,各自形成一个单独的文件,这样有利于降低 XML Schema 文件的复杂度。

3.2 事务方面的定制

3.2.1 定制原则

TPC-C 定义了 5 类事务,包括更新客户账户余额、发货、查询客户最近交易的状态、查询仓库库存状况、提交新订单。通过执行由上述事务按比例混合的测试负载来模拟终端用户操作。执行测试负载时每分钟可以处理新订单的交易量作为主要评测指标 tpmC。

目前对 TPC-C 测试基准的应用都是通过 SQL 语句操作关系型数据库,在访问 XML 数据库时,需要采用符合 SQL/XML 标准的语句对其进行实现。目前,SQL Server、Oracle、DB2 等主流数据库都对 SQL/XML 标准进行了支持。

3.2.2 定制方法

本文将 TPC-C 事务中的数据库操作以符合 SQL/XML 标准的语句实现,以便对 XML 数据进行操作,仅以查询仓库库存状况事务为例。

该事务用于确定最近卖出并且库存量低于特定值的商品数量,包含 3 个步骤:

- (1)确定某地区下一个可用订单号;
 - (2)确定该地区最后 20 个订单中的所有商品;
 - (3)对于每个选定的商品,确定其库存是否低于门限值。
- 上述功能可用 2 个 SQL 语句实现,具体如下:

```
SELECT d_next_o_id FROM district
WHERE d_w_id=:w_id AND d_id=:d_id;
SELECT COUNT(DISTINCT (s_i_id))
FROM order_line, stock
WHERE ol_w_id=:w_id AND ol_d_id=:d_id
AND ol_o_id<=:o_id
AND ol_o_id>=:o_id-20
AND s_w_id=:w_id AND s_i_id=ol_i_id AND s_quantity
<:threshold
```

其中,“:w_id”为变量,在系统运行时自动赋值。

将此语句使用 SQL/XML 标准进行重写,由于各字段的值都是保存在 XML 元素中,因此,需要使用 XPath 表达式查询出这些值查询再进行操作,具体如下:

```
SELECT xquery('/warehouse/ district/@D_W_ID', xmldoc)
FROM warehouse
WHERE xquery('/warehouse/ district/@D_W_ID', xmldoc)=:w_id
AND xquery('/warehouse/district/@D_ID', xmldoc)=:d_id;
SELECT COUNT(DISTINCT(xpath('/order/order
_line/@OL_I_ID', xmldoc))) FROM orders, stock
WHERE xquery('/order/ order_line/@OL_W_ID', xmldoc)=:w_id
AND xquery('/order/ order_line/@OL_D_ID', xmldoc)=:d_id
AND xquery('/order/ order_line/@OL_O_ID', xmldoc)<=:o_id
AND xquery('/order/ order_line/@OL_O_ID', xmldoc)>=:o_id-20
AND xquery('/stock/@S_W_ID', xmldoc)=:w_id
AND xquery('/stock/@S_I_ID', xmldoc)=ol_i_id
AND xquery('/stock/@S_QUANTITY', xmldoc) < :threshold
```

其中,xquery()为数据库系统内置 XML 字段处理函数,用以

获取 XML 元素内容或属性值。第 1 个参数是获取数据的 XPath 表达式,第 2 个参数是 XML 字段,保存所需数据。

4 测试方案设计与实现

4.1 系统架构

按照 TPC-C 测试基准,建立如图 2 所示的评测系统架构。

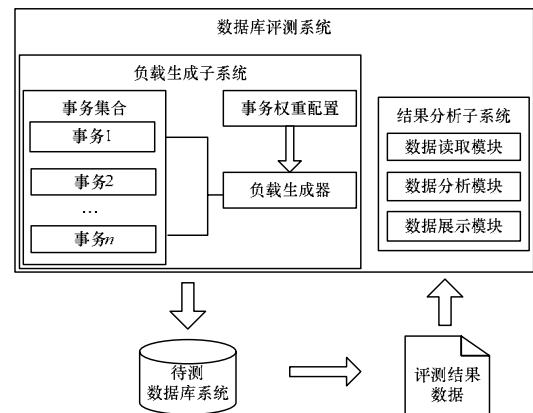


图 2 评测系统架构

在图 2 中,负载生成子系统根据事务权重配置生成负载,即由负载生成器按照各事务所占比重随机生成事务顺序、发生频率以及其中各 SQL 语句的数据。调度各事务并发操作数据库,并在各事务之间加入用户等待时间。在负载执行过程中实时生成评测结果数据并保存。在负载执行完成后,结果分析子系统对评测结果数据进行分析。数据读取模块获取已保存的数据,数据分析器统计分析评测指标,形成直观的统计信息,数据展示模块显示最终结果。

4.2 负载的生成

负载生成器可用测试工具 Loadrunner 实现,使用其 3 个主要模块 VUser、Controller 和 Analysis。

设计 2 个 VUser、VUser1 模拟 new-order 操作,VUser2 模拟其他操作。建立事务权重表,使 VUser 按照预定权重随机执行前述 5 个事务。既满足了 TPC-C 测试基准要求的交易执行机制,又消除了 CPU 性能、I/O 延迟等系统瓶颈对评测结果的影响。在执行负载时,通过 Controller 调度多个 VUser 模拟多用户并发环境,多个 VUser 也可以在同一个工作站运行以减少硬件需求。

4.3 SQL Server 2005 中的数据库设计

为简便起见,只测试一个 warehouse 场景。在数据库中建立 5 张表,每张表对应一个前述的对象,各表中都只有一个 XML 数据类型的列,用于存储 XML 文档。在生成 XML 数据时,将 warehouse 生成成为一个单独的文件,存为对应表中的一行。将其余对象各生成若干个文件,在对应表中存为单独一行。这样可以同时测试数据库处理单个大 XML 文件和多个小 XML 文件的能力。

5 测试结果分析

本文按照上述方案对一个实际系统进行了 8 h 以上的不间断测试。测试结果显示各事务具有如下特征:payment 执行频繁,响应时间短;delivery 执行不频繁,响应时间长;order-status 数量较少,响应快;stock-level 数量较少,new-order 执行较频繁。这些特征均与 TPC-C 测试基准的描述一致,说明本文测试方案是正确的。

通过计算各事务在每个时间点上的流量,可得出系统的

(下转第 41 页)