基于 RDBMS 的 XML 数据存储技术①

王振辉[1]，刘 军[2]，王振铎[3]

[1](西安翻译学院 信息工程学院，西安 710105)
[2](西安电力高等专科学校 计算机工程系，西安 710032)
[3](西安思源学院 电子信息工程学院，西安 710038)

摘 要：XML 已成为互联网事实上的数据表示标准，在数据交换和数据仓库中广泛应用。但 XML 文件特征不能保障数据的安全性和并发访问，而 RDBMS 严谨的关系理论和技术的成熟性可以弥补 XML 技术的不足。结合 XML 与 RDBMS 的优点，提出一个基于 RDBMS 的 XML 数据存取方案，用于简化 XML 数据的管理和数据仓库的构建。利用 Oracle XML DB 技术实现 XML 在关系数据库中存储、更新和检索操作，使用户能透明地通过 RDBMS 来管理 XML 数据，相对于映射策略的数据转储方式，明显提高了 XML 数据存储效率。

关键词：关系数据库；XML；数据存取； Oracle XML DB；数据仓库

XML Data Storage Technology Based on RDBMS

WANG Zhen-Hui[1], LIU Jun[2], WANG Zhen-Duo[3]

[1](College of Information Engineering, Xi'an Fan-yi University, Xi'an 710105, China)
[2](Department of Computer Engineering, Xi'an Electric Power College, Xi'an 710032, China)
[3](College of Electronics and Information Engineering, Xi'an Si-Yuan University, Xi'an 710038, China)

Abstract: XML has become the data representation standard on Internet, and widely used in data exchange and data warehouse. But the file characteristics of XML cannot guarantee the security of data and concurrent access and the precision of RDBMS theory and maturity of RDBMS can be achieved fast and concurrent access to data. The paper combines the advantages of XML and RDBMS and comes up with a XML storage resolution based on RDBMS to simplify the management of XML data and data warehouse building. It allows users to manage XML data based RDBMS transparently by using Oracle XML DB, relative to the middleware method mapping strategy. It can significantly improve the efficiency of the XML data storage.

Keywords: RDBMS; XML; data storage; Oracle XML DB; data warehouse

1 引言

近年来，一些信息化应用水平较高的企业为了整合内部的异构数据源，实现信息资源的再利用，纷纷构建数据仓库以实现企业商业智能和数据挖掘。数据仓库的核心技术就是将多种异构数据源转换为同一种大型关系数据库，然后进行主题检索。异构数据源相互转化方法主要有四种，一是利用数据转换程序，对数据格式进行转换，从而能被其它的系统接收[1]；二是利用前台开发工具提供的数据转换工具，将一种数据转存为另一种数据格式[2]；三是用动态 SQL 语句编程处理[3]；四是专用中间文件的数据交换方式[4]。

专用中间文件的数据交换方式是中间件思想的体现，目前中间文件多以 XML 文档为主。XML 是 Internet 公用数据格式定义语言。因为它的可扩展性和自描述性，使得异构数据交换变得容易起来，数据交换是 XML 最重要的用途之一。来自企业内部的多种异构数据源，通过 XML 技术集成后，为数据仓库提供了一个统一格式的数据，简化了数据仓库获取数据和转化

① 收稿时间:2010-07-10;收到修改稿时间:2010-08-10

数据的过程。目前，XML 数据交换技术主要采用模式映射的方法，将 XML 文档用 DTD 或 Schema 映射为关系数据库中的表[5,6]，而在数据交换过程中产生了大量的 XML 文档。随着数据量的增加，这些 XML 文档的统一管理和 XML 文档作为数据存储的缺陷—查询性能低下的特点暴露无疑，并且由于文件系统的约束，文件大小、安全性和并发性都不能满足用户的要求。

关系数据库有着严格的理论基础，丰富的查询语言和广泛的应用。同时，能够保证数据的完整性，安全性和并发性的要求。包括 Oracle、微软、IBM 在内的数据库厂商都在走从传统关系数据库中支持 XML 的技术路线。例如，微软公司的 SQL Server 2005 提供一种新的 XML 数据类型，使在 SQL Server 数据库中存储 XML 片段或文件成为可能，Oracle10G 提供了 XML Type 数据类型来构建 XML DB,对 XML 文档进行统一存储[7,8]。本文利用关系数据库的优点，介绍如何通过 Oracle XML DB 技术管理 XML 文档，解决 XML 的并发访问与更新问题，同时简化数据仓库的构建过程。

2 技术框架

传统的基于 XML 数据仓库系统技术框架如图 1 所示[9]。异构数据源数据经 DB-XML 转换程序，转化为中间 XML 文档，该文档经 XM-DB 程序，将多个异构数据源中的数据按数据仓库主题要求抽取、转换、合并后存入数据仓库的表中。其中 DB-XML 和 XML-DB 程序在数据集成中称作 Wrapper 和 Dewrapper，采用的技术是模式映射技术。

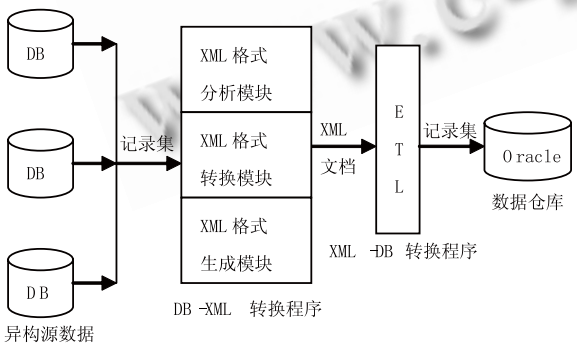

图 1 基于 XML 数据仓库系统技术框架图

本文总体设计思想是用支持 XML 存储的 Oracle10G 中 XML DB 技术取代复杂的 XML-DB 转换程序，由于存储 XML 数据的数据库系统与目标数据仓库类型一致，存取数据更加快捷、便利。图 2 是本文的基于关系数据库的 XML 数据仓库系统技术框架图。

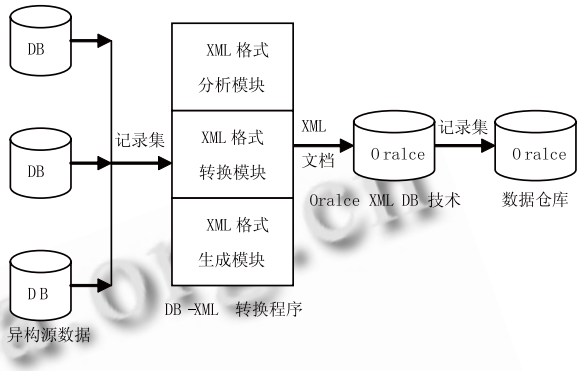

图 2 改进的基于 XML 数据仓库系统技术框架图

3 Oracle XML DB技术

XML DB 是利用 Oracle10G 数据库管理 XML 文档的一个特性。结合关系数据库技术和 XML 技术的优势为数据交换和数据的高效检索提供了新的方法。XML DB 可用于存储、查询、更新、转换或处理 XML，并使用 SQL 查询访问相同的 XML 数据[10]，相对 XML 查询技术 Xquery 和 Xpath 要高效很多。下面从 XML 数据存储、更新、查询三个方面说明 XML DB 实现技术。

3.1 XML 数据存储

Oracle10G 中使用 XMLTYPE 数据类型保存 XML 数据，具体存储步骤如下。

(1) 建立包含 XMLTYPE 字段的表

在 SCOTT 用户下建立表 tbl_xml_data，用于存放 XML 数据。

```
SQL>Create table tbl_xml_data
( xml_filename varchar(20) primary key,
      xml_content xmltype);
```

(2) 在 XMLType 数据表中存放数据。

将 XML 文档存入 XMLType 字段中，分为以下三个步骤：

1) 构建 XML 物理目录和 Oracle10G 虚拟目录的映射关系

首先，在 Oracle10G 数据库所在的服务器主机上建立存放 XML 文档的目录，如：D:\Oracle\ora10G\xml_file。

然后，在 Oracle10G 数据库中创建一个虚拟目录指向新创建的目录。

SQL> CREATE OR REPLACE DIRECTORY XMLDIR AS ‘D:\Oracle\ora10G\xml_file’

最后，将访问此目录的权限授权给用户 SCOTT

SQL>GRANT READ ON DIRECTORY XMLDIR TO SCOTT

2) 建立 XML 数据读取函数 getClobDocument

getClobDocument 函数用于读取 XML 文件的内容，在 system 用户中已经建立了这个文件读取函数，如果不是使用 system 用户读写 XML DB,可以按照下面代码，在 SCOTT 用户下建立这个函数。

```
SQL>Create or replace function getClobDocument
        ( filename in varchar2,
          charset in varchar2 default NULL)
          return CLOB deterministic
      is
        file bfile:= bfilename('XMLDIR',filename);
        charContent CLOB := ' ';
        targetFile bfile;
                        lang_ctx number :=
DBMS_LOB.default_lang_ctx;
        charset_id number := 0;
        src_offset number := 1;
        dst_offset number: = 1;
        warning number;
      begin
        if charset is not null then
                                charset_id :=
NLS_CHARSET_ID(charset);
        end if;
        targetFile := file;
DBMS_LOB.fileopen(targetFile,
DBMS_LOB.file_readonly);
DBMS_LOB.LOADCLOBFROMFILE(charContent,
targetFile,DBMS_LOB.getLength(targetFile), src_offset,
dst_offset,
                charset_id,lang_ctx,warning);
        DBMS_LOB.fileclose(targetFile);
        return charContent;
    end;
```

3) 将 XML 数据存入 XML TYPE 字段中

SQL>insert into tbl_xml_data values('book.xml',XMLTYPE(getClobDocument ('book.xml')))

注：book.xml 文件存放在前面建立的 D:\Oracle\ora10G\xml_file 目录中。

也可以使用如下的 PL/SQL 过程来手工完成 XML 数据的存储。

```
SQL> declare
       XML_TEXT CLOB:='<书>
         <ISBN>7-5053-9255-7</ISBN>
         <书名>Oracle 实用教程</书名>
         <作者>郑阿奇</作者>
                        </书>';
      begin
     insert into tbl_xml_data
values('book.xml',XMLTYPE(XML_TEXT));
      end;
```

3.2 更新 XML 数据

更新 XML 数据即将 XMLType column 列中的值为替换新的 XML 数据。可以使用如下命令来实现。

(1) 更新数据

```
SQL>update tbl_xml_data
      set xml_content
=XMLTYPE(getClobDocument('book_new.xml'));
```

(2) 清除列中 XML 数据。

```
SQL>update tbl_xml_data set xml_content =null;
```

3.3 查询 XML 数据

在 Oracle XML DB 中用 extract(),extractValue()和 existsNode()等函数结合 Xpath 表达式来完成一些数据操作。

SQL/XML 运算符分为两类：第一类运算符允许以普通 SQL 操作查询和访问 XML 内容。第二类运算符提供了一个业界标准的方法，从 SQL Select 语句的结果集中生成 XML 文档。

(1) 返回一个节点值

```
SQL>select extractvalue(xml_content,'/书/书名') as
书名 from tbl_xml_data;
```

(2) 判断节点是否存在

```
SQL>SELECT existsNode(f02,'/书[书名="Oracle
实用教程"]')
```

```
FROM tbl_xml_data ;
SQL>SELECT count(*)
FROM tbl_xml_data
WHERE existsNode
(f02,'/书[书名="Oracle 实用教程"]') = 1;
```

（3）返回 XML 文档片断

```
SQL>SELECT extract(f02,'/书')
    FROM tbl_xml_data ;
```

4 总结

关系数据库仍然是目前主流的数据库存储技术，使用关系数据库存取 XML 数据实用性强，可以实现用户对 XML 数据的透明管理，同时提高了数据交换的效率。本文根据 XML 标准化思路，结合关系数据库的优势，提出了基于关系数据库管理 XML 文件的数据存储方案，通过使用 Oracle XML DB 技术构建了 XML 数据存取系统，大大简化了建立数据仓库的数据的获取，转化和处理过程，同时实现了 XML 数据的并发访问，提高了数据的完整性和安全性。随着这两种技术的相互借鉴和结合，数据的标准化工作和数据挖掘技术也会更加日臻完善。

参考文献

1 陈弦,陈松乔.基于数据仓库的通用 ETL 工具的设计与实现.计算机应用研究,2004,21(8):214－216.

2 张丽华.基于 PB 数据管道的异构数据库转换系统设计与实现.计算机系统应用,2006,15(11):73－76.

3 罗林球,孟琦,李晓,苏国平,张澄澈.异构数据库迁移的设计和实现.计算机应用研究,2006,23(12):233－238,

4 周红波,孙宇达,王继霞,王瑞,王志宝.基于 XML 的数据交换及其参照完整性研究.计算机工程与设计,2006,27(14):2611－2613.

5 顾天竺,沈洁,陈晓红.基于 XML 的异构数据集成模式的研究.计算机应用研究,2007,24(4):94－96.

6 文必龙,王守信.一个基于 XML Schema 的数据交换模型.大庆石油学院学报,2004,28(2):65－68,

7 屈正庚.利用 SQLXML 创建 XML 查询的方法.商洛学院学报,2006,(4):34－37.

8 江枫.Oracle XML DB 的发展历程.程序员,2007,(12):67－69.

9 仇丽青,赵庆祯,基于 XML 的数据仓库系统.计算机系统应用,2004,13(2):12－14.

10 兰小机,鲁小娟.基于 Oracle XML DB 的 XML 文档存储技术的研究.测绘科学,2008,33(5):201－203.

(上接第 172 页)

证了本文提出的 ParaClustalW 算法的并行处理具有很好的性能。

5 结论与展望

本文把桌面网格平台运用到生物信息学多序列比对领域，分析了 ClustalW 算法的任务划分策略，描述了 ParaClustalW 并行化策略，模拟实验表明，该方法具有较好的性能，对多序列比对算法的研究具有一定的参考价值。如何更好的根据桌面网格平台的特点，优化 ClustalW 算法的并行划分策略，实现自适应并行与负载均衡，充分提高因特网上工作机的空闲处理器周期是本文的下一步工作。

参考文献

1 Andrade N, Costa L, Germoglio G, Cirne W. Peer-to-peer grid computing with the ourgrid community. Proc. of the SBRC 2005-IV Salao de Ferramentas. 2005.

2 Anglano C, Canonico M, Guazzone M, Botta M, Rabellino S, Arena S, Girardi G. Peer-to-Peer Desktop Grids in the Real World: The ShareGrid Project. Proc. of 8th IEEE International Symp. on Cluster Computing and the Grid, 2008. CCGRID'08. 2008: 621－626.

3 Thompson JD, Higgins DG, Gibson TJ. CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. Nucleic Acids Research. 1994(22):4673－4680.

4 Zhu MJ, Hu GW, Zheng QL, et al. Multiple sequence alignment using minimum spanning tree. Proc. of 2005 International Conference on Machine Learning and Cybernetics, 2005(ICMLC 2005). Guangzhou, IEEE Computer Society, 2005: 3352－3356.

5 Edgar RC. Muscle: multiple sequence alignment with high accuracy and high throughput. Nucleic Acids Res, 2004 (23): 1792－1797.