第34卷 第22期 计 算 机 工 程 2008年11月
Vol.34 No.22 Computer Engineering November 2008
·软件技术与数据库· 文章编号：1000—3428(2008)22—0052—03 文献标识码：A 中图分类号：TP311

网格环境下基于XML的异构数据集成系统

郑 荣，马世龙
(北京航空航天大学软件开发环境国家重点实验室，北京 100083)

摘 要：分析地震、地质行业的数据资源特点，在数据网格中间件OGSA-DAI基础上提出一种基于XML的分布异构数据访问与集成框架，实现数据的透明访问和联合查询。系统以XML作为公共数据模型，使用三层模式集成机制，以XQuery同时作为XML模式之间的映射语言及全局查询语言，简化全局视图的构造和系统的查询处理。

关键词：异构数据源；数据集成；数据网格；模式映射；全局查询

XML-based Heterogeneous Data Integration System in Grid Environment

ZHENG Rong, MA Shi-long
(National Key Lab of Software Development Environment, Beijing University of Aeronautics & Astronautics, Beijing 100083)

【Abstract】 The characteristics of data resources in seismological and geological industries are analyzed. Utilizing the data grid middleware OGSA-DAI, an XML-based heterogeneous data access and integration framework is presented, and it can provide data transparent access and fusion query. The system chooses XML as a common data model, utilizes the three-layer schema integration mechanism. XQuery is used as both the mapping language among XML schemas and the global query language, so that the global view construction and the query processing of the system are simplified.

【Key words】 heterogeneous data resource; data integration; data grid; schema mapping; global query

1 概述

在地震、地质等数据密集型行业内实现数据的充分共享、互连互通，通过数据资源的聚合与协同满足应用的需求，是当前信息化建设的重要方面。由于信息化程度差异，这些领域的数据资源以各种形式存在(如数据库、文件等)，并且按行政区域存储在各级单位，形成Internet下的数据孤岛。

网格技术[1]的发展，很好地解决了分布环境下数据资源共享的问题。英国e-Science核心项目OGSA-DAI(Open Grid Service Architecture Data Access and Integration)[2]是一个通过网格提供对各种独立数据源访问和集成的中间件。OGSA-DAI允许用户通过统一的接口访问网格上不同类型的数据资源。但OGSA-DAI没有对各种异构数据模式进行统一描述，因而无法实现真正意义上的数据透明访问和数据集成。

XML具有平台无关性、自描述性、可扩展性、简单易于处理等优点，其相关技术的成熟使之成为Internet下数据表示和交换的标准。XML可在数据集成中扮演全局模式的角色，并采用XML查询语言——XQuery作为全局查询语言[3]。本文提出了一种网格环境下基于XML的异构数据访问和集成框架，扩展了OGSA-DAI，在OGSA-DAI提供数据共享和统一访问接口的基础上，通过XML公共数据模型和虚拟全局视图实现了数据的透明访问和联合查询。

2 异构数据访问与集成系统体系结构

数据访问与集成中间件复用在多个应用中，为上层应用提供全局视图，屏蔽底层数据的分布性、异构性。下文将从两个角度来阐述系统的结构。从数据的角度看，系统采用XML作为公共数据模型，通过面向应用的全局视图实现数据的透明访问和协同汇聚，如图1所示。数据访问与集成中间件分为数据访问层与数据汇聚层。

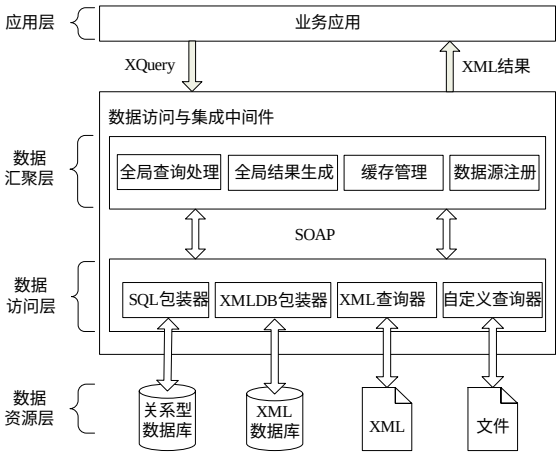

图1 异构数据访问与集成系统结构

(1)数据访问层：此层接受针对导出视图的XQuery查询，实现单一异构数据源的统一访问。对于有查询能力的数据源(如关系数据库、XML数据库)，包装器将XQuery转换为数据源能执行的查询语句并对查询结果进行转换。对于没有查

基金项目：国家“973”计划基金资助项目(2005B321905)；国家地震网络计算应用系统基金资助项目(2004DKA50740)
作者简介：郑 荣(1979－)，女，硕士研究生，主研方向：数据网格，数据集成；马世龙，教授、博士生导师
收稿日期：2007-12-20 **E-mail：**zhengrong@nlsde.buaa.edu.cn

询能力的数据源(如文件)，查询器将执行 XQuery 查询提取结果。

(2)数据汇聚层：此层接受针对全局视图的 XQuery 查询语句，并返回全局模式的 XML 结果。为了提高系统的性能，在此层需要提供数据缓存管理功能。

从网格服务的角度来看，系统将数据访问层的包装器和查询器都作为一种新的数据访问器集成在 OGSA-DAI 中，通过 OGSA-DAI 的网格数据服务来访问数据。数据汇聚层的全局查询器也封装成网格服务。如图 2 所示，全局查询处理服务接受查询请求，将请求分解为单个数据源的子查询，调用相应的数据网格服务执行，合并网格数据服务返回的查询结果。

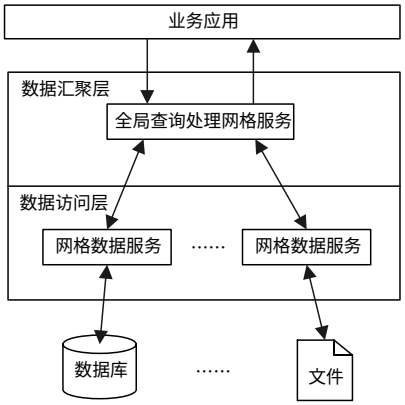

图 2　数据访问与集成网格服务框架

3　系统实现的关键技术

3.1　模式映射及集成

数据集成的首要问题是通过模式映射和集成解决数据资源的模式异构和语义异构，以提供统一视图。本文使用一种基于 XML 的三层模式集成机制，如图 3 所示，三层模式分别为：本地模式(local schema)，导出模式(export schema)，面向应用的全局模式(application-oriented global schema，简称为全局模式)。从本地模式到导出模式的映射称为局部映射，主要解决数据模式上的异构，通过定义映射的规则可实现该过程的自动化。导出模式与全局模式都是基于 XML 模式的，从导出模式到全局模式的映射称为全局映射，用来解决数据语义的异构以及数据的汇聚，需要数据管理员(或二次开发人员)手工定义。这种三层模式集成机制易于扩展，当出现新的数据源时，只需要扩展该数据源的局部映射，而全局映射无需扩展。

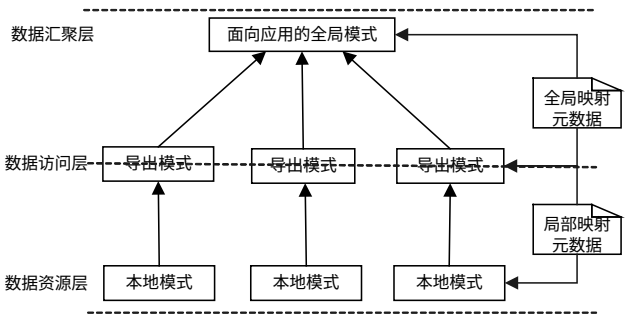

图 3　模式集成结构

3.1.1　局部映射

局部映射解决数据模式异构的问题，即如何将非XML类型的数据模型映射为XML数据模型。局部映射可用三元组(E, L, M_{LE})表示。E表示导出模式，L表示本地模式，E, L一一对应。M_{LE}表示L和E之间的映射，由局部映射规则来描述。

XML的模式描述语言目前主要有DTD(Document Type Definition)和XML Schema两种。XML Schema比DTD具有明显的优势[4]，因此本系统采用XML Schema描述导出模式。

由于目前使用最多的非XML数据源为关系数据库，因此本文重点讨论关系模式到XML模式的映射[5]。该系统中一个关系模式被映射成一个四层的XML模式，依次对应数据库、表、记录、字段。映射的规则如下：

(1)数据库的映射：映射为 XML 的根元素，元素名为数据库的名称。

(2)表的映射：映射为根元素的子元素(称为表元素)，元素名为表名，元素类型为复杂类型。

(3)记录的映射：在每个表元素下，生成一个名为“表名_tuple”的子元素(称为 tuple 元素)，代表表中的一行，元素类型为复杂类型，并设置其出现次数为零到多次。

(4)列的映射：映射为对应表的 tuple 元素下的子元素(称为列元素)，元素名为列名。

(5)数据类型的映射：XML Schema 有丰富的数据类型与关系数据库中的数据类型对应，这里不一一列举。数据类型映射到对应列元素的 type 属性；是否为空映射为 nillable 属性；默认值映射为 default 属性。

(6)主外键关系的映射：关系模式中的主外键分别映射成 XML Schema 中的 key 和 keyref 约束。

例如一个用来存储地震信息的关系数据库 EarthquakeInfo，其中有表 tab_earthquake(ID int primary key, Lat float, Long float, Occur_Date date, M float…)，ID, Lat, Long, Occur_Date, M 分别表示地震 ID、纬度、经度、发生时间、震级，经局部映射后的导出模式用 XML Schema 描述如下：

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified" attributeFormDefault="unqualified">
 <xs:element name="EarthquakeInfo">
  <xs:complexType>
  <xs:all>
   <xs:element name="tab_earthquake">
    <xs:complexType>
     <xs:sequence>
      <xs:element
name="Tab_earthquake_tuple" minOccurs="0"
 maxOccurs="unbounded">
       <xs:complexType>
        <xs:all>
         <xs:element name="ID" type="xs:int"/>
         <xs:element name="Lat" type="xs:float"/>
         <xs:element  name="Long" type="xs:float"/>
          <xs:element
name="Occur_Date" type="xs:Date"/>
 <xs:element name="M" type="xs:float"/>
        </xs:all>
       </xs:complexType>
      </xs:element>
     </xs:sequence>
     </xs:complexType>
```

```
                    <xs:key name="EarthquakeID">
                        <xs:selector
xpath="./Tab_earthquake_tuple/ID"/>
                        <xs:field xpath="."/>
                    </xs:key>
                </xs:element>
            </xs:all>
        </xs:complexType>
    </xs:element>
</xs:schema>
```

3.1.2 全局映射

经过局部映射后，导出模式都是XML类型的，因而全局映射不再需要解决模式异构的问题，全局映射的目的是将导出模式重构合并形成面向应用的全局视图。全局映射可用(G, E, M_{GE})表示，G表示全局模式，E为一个集合(E_1, E_2,…,E_n)，其中，E_1到E_n分别表示不同的数据源的导出模式。M_{GE}表示G和E之间的映射关系。本文用XQuery来描述M_{GE}，在XQuery中定义标签来表示全局模式中的元素，通过路径表达式Xpath来表示局部模式中的元素。

采用 XQuery 同时作为 XML 的查询语言以及全局映射语言，不论对于用户还是对于系统的实现都具有很大的优势。对于系统的用户来说，定义视图将与构造查询一样方便，不用重新学习特有的映射语言。对于系统的实现来说，不需要专有的映射语言解释器，将全局查询语句与全局映射的 XQuery 语言合并就可完成查询的重写。

3.2 查询处理

上一节中的模式映射及集成为用户提供了一个面向应用的全局视图。在用户眼里，数据如同放在一个大“ XML 文档 ”中，用户基于全局视图构造 XQuery(本系统支持 FLWOR(For-Let-Where-Orderby-Return)语句)来访问数据并得到 XML 格式的结果。查询的处理过程如图 4 所示，分为数据汇聚层的全局查询和数据访问层的局部查询。

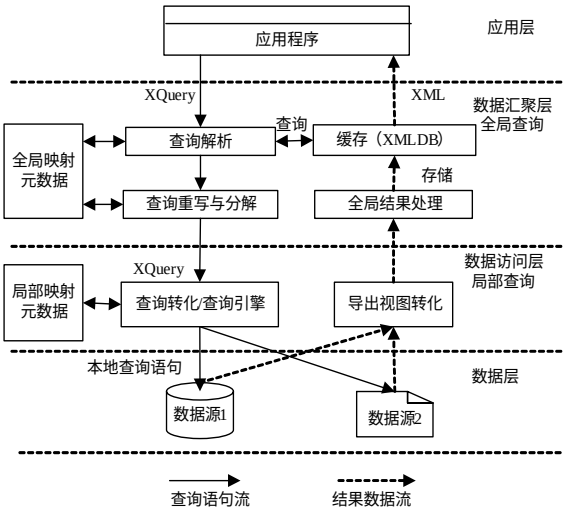

图 4 查询处理示意图

3.2.1 全局查询

全局查询的处理过程包括查询解析，查询重写与分解，全局结果处理。查询解析根据全局模式对查询语句作语法检查。如果语法正确，查询缓存(XMLDB)中是否有该查询结果，如果有则直接返回查询结果；如果语法不正确，返回错误提示。查询重写是将查询的 XQuery 语句与全局映射的 XQuery 语句进行合并以及标准化。查询分解是根据数据源的位置信息将重写后的查询请求分解为单个数据源可执行的子查询，并生成查询计划。依据查询计划调用相应的数据网格服务执行查询操作，待查询计划执行完毕，将结果合并，转化为全局视图格式返回给用户。

全局查询分解是全局查询中最重要的部分。一个全局查询分解过程可表示为Q_G =(Q_e, *Cond*)，Q_G表示全局查询，Q_e是一个集合，表示分解后的子查询Q_{e1}, Q_{e2},…, Q_{ei}，*Cond*表示子查询结果的约束集合，表示为*Cond* =(Q_{ei}, Q_{ej}, *Opt*)，*Opt*表示查询结果之间的关系，可为Union, Intersection, Minus。数据源通过网格服务来访问，同时这个过程也可以表示为服务的分解过程S_G=(S_e, *Cond*)，S_G表示全局查询网格服务，S_e是一个集合，表示包装对应数据源的网格服务S_{e1}, S_{e2},…, S_{ei}， *Cond* 的意义同上。

3.2.2 局部查询

对于有查询能力的数据源，局部查询将 FLWOR 语句转换为本地查询引擎能够执行的查询语句，由本地查询引擎执行查询操作后，再将结果转为导出视图返回。对于关系数据源，基于 3.1.1 节介绍的关系模式到 XML 模式的映射规则，FLWOR 语句被转化成 SFWO(Select-From-Where-Orderby)语句，由 SQL 引擎执行。对于 XML 数据库，一般直接支持 XQuery 查询。

对于没有查询能力的数据源，局部查询要实现 FLWOR 查询引擎。对于 XML 文件，目前已有成熟的 XQuery 引擎，本系统中采用 XQEngine，将其集成到系统即可。

4 结束语

本文提出了一种异构数据访问与集成框架，通过扩展 OGSA-DAI，实现了数据的充分共享、透明访问和联合查询。系统以 XML 作为公共数据模型，采用三层模式集成机制，并以 XQuery 同时作为 XML 模式之间的映射语言及全局查询语言，简化了全局视图的构造和系统的查询处理。

参考文献

[1] Foster I, Kesselma C. 网格计算[M]. 北京: 电子工业出版社, 2004.

[2] OGSA-DAI WSRF 2.2 User Guide[EB/OL]. (2006-10-20). http://www.ogsadai.org.uk/documentation/ogsadai-wsrf-2.2/doc/.

[3] Halevy A, Rajaraman A, Ordille J. Data Integration: the Teenage Years[C]//Proc. of International Conference on Very Large Data Bases. Seoul, Korea: VLDB Endowment, 2006: 9-16.

[4] XML Schema Part 0: Primer Second Edition[EB/OL]. (2004-10-20). http://www.w3.org/TR/xmlschema-0/.

[5] Bourret R. Mapping DTDs to Databases[EB/OL]. (2005-09-20). http://www.rpbourret.com/xml/DTDToDatabase.htm.