文章编号:1000-582X(2008)09-1054-05

扩展 OGSA-DAI 数据访问与集成框架的关键技术分析

李献礼
(长江师范学院 网络信息中心,重庆 涪陵 408000)

摘 要:基于 OGSA-DAI 数据访问模式,以网络服务方式提供对异构数据资源统一访问,更方便地满足特殊的数据源访问、集成。通过编写相应的 Activity 类操纵数据,创建服务器端 Activity 负责具体数据处理任务,创建客户端 Activity 用于生成执行文档和执行数据转换任务,扩展 OGSA-DAI 以支持异构数据源的集成和访问。

关键词:数据访问与集成;OGSA-DAI;数据源访问类;异构
中图分类号:TP393.09 文献标志码:A

Key techniques to extend data access and integration frame for OGSA-DAI

LI Xian-li
(Computer Network Information Center, Yangtze Normal University, Fuling 408000, P. R. China)

Abstract: OGSA-DAI is a middleware that provides universal data access for heterogeneous data sources with network services. It unites access and control of scattered and heterogeneous data to one logic data source and shields the database driver, data formats and communication protocols of the heterogeneous systems for users. Thus, the heterogeneous data source may be accessed and controlled as a single logic source, making it more convenient to access and integrate the special data source. To extend OGSA-DAI to support heterogeneous data access and integration services, new activity classes were designed to manipulate the data. Server-side activities were created to process the data, while client-side activities were created to build execution documents and data conversion based on XML Schema.

Key words: data access and integration; OGSA-DAI; data resource accessories; heterogeneous

随着网格技术的发展和成熟,支持大规模分布式数据访问与集成共享的数据网格越来越受到重视。面向服务的架构(SOA)和开放网格服务架构(OGSA)逐渐成为技术发展的主流。以网格服务方式对数据资源进行封装,然后提供统一的访问界面,是当前这一领域中较为普遍的作法[1]。但网格环境中的数据集成与访问主要面临的问题有:1)网格建立在服务的基础上,在网格中存取数据库必须符合网格的标准,即数据库应该成为网格中的一种资源并且提供相应的服务;2)数据库有不同的种类,并且属于同一种类的数据库产品在功能和接口上也有很大的不同,在集成各种数据库到网格中时必须减少重复工作,同时又要尽可能保留被集成的数据库的全部功能;3)网格鼓励数据共享,但数据来自于不同的研究者和组织机构,有着各自的数据库模式和数据库设计;4)网格中不仅包含结构化的数据,也包含半结构化的数据和无结构的数据[2]。因此,在数据网格环境中,针对数据的集成与访问提出了不少新的、特定的要求。其中最基本,也是最主要的是把多个相关的数据资源集成并封装为一个虚拟数据资

收稿日期:2008-04-01
基金项目:重庆市教委科学研究项目(KJ071306)
作者简介:李献礼(1960-),男,长江师范学院副教授,主要从事计算机网络应用、数据挖掘方向研究,(Tel)13372771011;(E-mai)lixl@yznu.edu.cn。

源，以标准的网格数据服务的形式向用户提供服务。

1　数据网格

数据网格要求通过提供一组服务来支持资源和信息发现，通过存储资源代理使计算可以在异构的存储资源上进行，因此，数据网格至少应该提供以下几种服务[3]：1）元数据目录服务，实现元数据目录并提供访问 API，通过 API 可以插入、更新、删除、查询目录中的数据；2）注册与发布；支持新实体的注册和利用元数据目录进行元据及相关数据的发布，利用注册服务来记录已经注册实体间的相互约束和相互联系，利用发布服务来控制对元数据及其它数据的访问级别；3）信息发现，为支持信息发现提供必要的工具；4）存储资源代理服务。该服务提供存储资源代理，用于将存储、检索数据集等高层用户的请求映射为异构分布式存储环境中的底层存储操作，并能够有效管理数据副本，存储资源代理利用存储在元数据目录中的信息来实现这一功能。

2　网格环境下数据访问与集成的基本方法分析

OGSA-DAI（open grid services architecture-data access and integration）项目是 DAIS 工作组制定的网格数据库服务标准草案的一个参考实现，其目标是为大量有访问共享数据需求的应用提供一种通用的中间件，能够支持以服务方式实现对数据资源的统一访问[4]。OGSA-DAI 用网格服务的形式为用户提供数据访问和管理服务（如图 1 所示）。OGSA-DAI 的容器（globus toolkit or tomcat）启动成功后，网格数据服务工厂（grid data service factory，GDSF）便随之启动，该 GDSF 被配置为用特定的网格数据服务组注册器（data access and integration service group registry，DAISGR）对其进行注册，并连接到底层的数据资源上，数据源可以是结构化数据（关系数据库），也可以是半结构化（XML 数据库）或者是非结构化数据（普通文件）。GDSF 中包括：可用的活动、模式以及实现类；资源的元数据，它可以有 2 种形式：静态（写在配置文件中）或通过实现类 MetaDataExtractor 获取；驱动信息，其中包括数据资源的实现类，JDBC 驱动（或与之等价的对象）；资源的 URI。

一旦选定合适的 GDSF，客户机就请求其创建一个 GDS 实例，以访问特定的数据资源。GDSF 根据客户的请求创建网格数据服务（grid data service，GDS），一个 GDS 提供了对网格上数据资源的访问点。GDSF 能创建不同类型的 GDS，可以是暂时的也可以是永久。一个永久的 GDS 可以被看成是为一个数据库或其它的数据资源提供一个永久的“网格接口”，暂时的 GDS 可以在请求数据访问的单个用户连接或者在临时数据复制时产生。当 GDS 创建出来时，系统根据一些上下文信息和 GDS 的配置（最初是从工厂的配置中来的）将引擎实例化。当 GDS 接收到一份执行文档时，它将这份文档与一些上下文信息一起传送到引擎中。然后，引擎对执行文档依次进行如下的处理：解析并验证执行文档；识别执行文档中指定的活动；创建这些活动的实现实例；创建这些活动之间管道的实例；创建活动处理器的实例并启动；用处理器处理这些活动，将产生的数据通过管道传递；将输出信息结合在一起，形成一份响应文档；将这份响应文档返回给 GDS，由 GDS 返回客户机[5-6]。

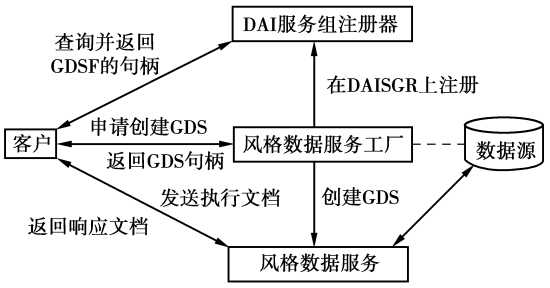

图 1　基于 **OGSA-DAI** 的典型交互模式

3　基于 OGSA-DAI 的数据网格访问与集成分析

OGSA-DAI 项目是基于 Java 实现的网格环境下数据访问与封装的中间件，为访问目前流行关系数据库（通过 JDBC）和 XML 数据库（通过 XMLDB）提供了大量的接口，能以多种不同的方式查询、操作、格式化数据集以及传送数据且屏蔽各种数据源的差异。OGSA-DAI 被设计成易于用户扩展，使用户能方便的为特殊的数据源开发满足需要的访问集成方法及其他的附加功能，并通过编写相应的 Activity 类来操纵数据[7]。

3.1　OGSA-DAI 项目中数据访问与集成的实现

OGSA-DAI-WSRF2.2 中，与数据源相关的配置信息被存储在/WEB-INF/etc/目录中的 dataResourceConfig.xml 和 DatabaseRoles.xml 文件中。dataResourceConfig.xml 文件中定义了有关数据源连接的一些信息，例如数据源驱动、URI 等。

DatabaseRoles.xml文件定义了有关用户名,密码的信息。下面分别列出了这2个配置文件的主要内容:

dataResourceConfig.xml文件主要内容(以访问MySQL数据库为例)

```
<!—数据源的元数据信息,用于描述数据源的产品信息—>
<metaData>
<productInfo>
    …
</productInfo>
<relationalMetaData>
<databaseSchema callback="uk.org.ogsadai.dataresource.MySQLMetaDataExtractor"/>
</relationalMetaData>
</metaData>
<!——指定角色映射文件——>
<roleMa Pname="Name"
implementation="uk.org.ogsadai.common.rolemap.SimpleFileRoleMapper" configuration="d:/tomcat—4.1.3/webapps/wsrf/WEB—INF/etc/ogsadai_wsrf/MySQLResource/DatabaseRoles.xml"/>
<!——指定数据源连接相关信息——>
<dataResource>
<driver implementation="com.mysql.jdbc.Driver">
<uri>jdbc:mysql://localhost:3306/ogsadai</uri>
</driver>
</dataResource>
```

角色映射文件DatabaseRoles.xml文件主要内容如下:

```
<!——指定访问数据源的用户名与密码——>
<Database name="jdbc:mysql://localhost:3306/ogsadai">
<User dn="*" userid="root" password="123456" />
</Database>
```

另外,该目录下还有一个dataResourceClassCon-fig.xml文件,其主要内容是

```
<dataResourceClass implementation="uk.org.ogsadai.dataresource.JDBCDataResource"/>
```

这表明在与特定数据交互的过程中,GDS并不是孤立地管理它与数据库的交互。与此相反,会存在一个对象,用来管理或缓存连接,对于JDBC连接,GDS使用JDBCDataResource类来管理,用户可通过把此处的值修改成自己提供的新类的类名,从而达到实现对数据连接的不同管理措施或者是为特殊数据源提供连接访问方法。数据集成和访问涉及的类可以划分为3组[8]:

1)JDBCDataResource类,它通过实现DataResource和JDBCConnectionProvider接口,提供getJDBCConnection(String userCredentials)、releaseJDBCConnection(Connection connection)、getJDBCDriverClassName()等诸多方法的实现来负责处理JDBC连接、驱动程序、角色映射以及访问DBMS。OGSA-DAI架构可以支持连接池,方法是创建继承自JDBCDataResource类的新类JDBCPoolingDataResource,并覆盖getJDBCConnection()、releaseJDBCConnection()。一旦编写好这个新类,就可替换dataResourceClassConfig.xml配置文件中implementation的值为新创建的JDBPoolingDataResource,从而让某个数据资源使用这个新的管理类[9]。

2)SQL活动类,负责接受SQL,并用JDBC连接将这个SQL传递给DBMS,然后对返回的结果进行格式化。所有具体的SQL活动类(比如:SQLUpdateStatementActivity、SQLQueryStatementActivity、SQLBulkLoadRowSetActivity等)都要直接或间接扩展AbstractSQLActivity类。AbstractSQLActivity抽象类中包含受保护的方法,可以用这些方法从活动中获取输入和输出信息。输入可以是值(由执行文档提供),也可以是引用(从其他活动中流入),也可以作为SQL参数、SQL表达式以及存储过程名称等的响应。输出是ResultSet,或者根据活动的类型不同,也可能是被更新的记录数目[10]。

3)MetaDataExtractor类,发布与DataResource有关的服务数据,例如:数据库的逻辑和物理名字,数据库的内容,组织结构,各字段的属性,以便在注册器或工厂服务中选择正确的数据资源。

3.2 扩展OGSA-DAI访问与集成异构数据源

基于OGSA-DAI开发新的数据源访问及集成方法可以分为3个主要步骤完成。

步骤定义新的数据源访问实现类,基本方法是:①定义包含获取数据连接、释放数据连接等方法名的数据源访问接口(比如:DSConnectionProvider);②编写配置文件,所有相关的数据源配置信息都存储在dataResourceConfig.xml、DatabaseRoles.xml和dataResourceClassConfig.xml文件中,其中定义

与访问数据源相关的一些初始化参数；③定义资源属性，以引用配置文件中的相关参数；④开发具体的数据源访问实现类，该类必须实现 OGSA-DAI 提供的 uk. org. ogsadai. dataresource. DataResource 接口和前面自定义的该数据源访问接口 DSConnectionProvider[11]。

第二个步骤为新数据源访问实现类创建 Activity来执行相关的数据源操作，其中服务器端的 Activity实现具体的数据处理任务，而客户端的 Activity通常用于生成执行文档以及执行数据转换任务，涉及的所有的输入/输出参数都必须定义在 XML schema 文件中。具体方法是：①提供一个 XML Schema 文件描述 Activity，这个 XML Schema 被用来验证那些传入 Activity 构建函数的 XML 元素，该文件主要包含 2 个部分，分别是输入部分和输出部分；②编写服务器端的 Activity 类代码，必须实现 uk. org. ogsadai. activity. Activity 接口，所有关于数据源访问的操作都被实现在 processBlock() 方法中[12]。可以从一个 ActivityContext 对象中得到一个相应的数据源连接对象，输出的结果也必须被塞入到这个 ActivityContext 对象中。而输入参数可以从构建方法中的 Element 对象中得到；③编写相应的客户端 Activity 类，利用 generateXML() 方法来构建一个 XML 文档（执行文档）来传送需要的输入和输出参数，而代表输入和输出参数的 XML 元素的语法必须满足已经定义的 XML schema 文件中的定义[13]。

第三个步骤，布署数据源访问实现类和相应的 Activity 类。①OGSA-DAI-WSRF2. 2 中，使用

```
ant deployPlugInResourceTomcat
-Dtomcat. dir=    //tomcat 的安装目录
-Ddai. resource. id=    //数据资源 id 编号
-Ddai. resource. impl=    //数据资源访问实现类名
```

执行 build. xml 文件中的 deployPlugInResourceTomcat 任务创建与数据源访问相关的配置文件（dataResourceClassConfig. xml、activityConfig. xml 等）并布署至 tomcat 容器中；

②编辑 activityConfig. xml 文件，新增一个 Activity：

```
<activity name="NewActivity"
  implementation="my. uk. org. ogsadai. activity. NewActivity"
  schema="NewActivity. xsd"/>
```

③最后，把数据源访问的实现类和相应的 Activity类打包为 jar 文件后拷贝到 tomcat 安装目录下的 webapps/wsrf/WEB－INF/lib 中[14]。

3.3　实现客户端对数据的访问

应用程序使用 OGSA-DAI 客户端工具集提供的多种不同 Activity（也可自己新建 Activity）来分别执行 SQL 操作、XSL 数据转换以及数据传输等任务，也可汇集多个请求后再向数据源服务对象提交执行[15]。下面的例子演示了在 tomcat＋GT4＋OGSA-DAI-WSRF 环境下怎样从数据源获取数据以及将获取的数据转换为指定格式的方法：

```
/* XSLTransformWithDelivery. java */
  // 设置 Data Service 的 URL 和 resourceID
String handle = "http://localhost:8080/wsrf/services/ogsadai/DataService";
String id = "MySQLResource";
  // 创建一个 GDS 服务实例
DataService service = GenericServiceFetcher. getInstance(). getDataService(handle, id);
// 获取 XSL 转换格式文档，其中定义了数据元素的转换方式
String url = "http://localhost:8080/tutorial/transformRowSet. xsl";
DeliverFromURL deliver = new DeliverFromURL( url );
  //构造 sql 查询 Activity
SQLQuery query = new SQLQuery( "select * from books where id<=100");
WebRowSet rowset = new WebRowSet(query. getOutput());
XSLTransform transform = new XSLTransform();  //构造数据转换 Activity
transform. setXMLInput( rowset. getOutput( ) );
//指定数据来源
transform. setXSLTInput( deliver. getOutput( ) );
//指定转换格式
  //构造请求并把多个 activites 绑定
ActivityRequest request = new ActivityRequest();
request. add(deliver);
request. add(query);
request. add(rowset);
request. add(transform);
service. perform( request ); //执行请求
System. out. println( "Results:\n" + transform. getOutput(). getData());  //查看数据转换
```

后的结果。

4 结 语

OGSA-DAI提供了一个整合已有数据存储产品和基础网格架构从而形成数据网格的集成解决方案,利用OGSA-DAI提供的接口,把对分散、异构数据的访问和控制统一在单一逻辑数据源上,为用户屏蔽掉异构系统上不同数据库驱动、数据格式和通讯协议等运行机制,从而使各种异构的数据源可以被当作单一的逻辑资源来进行存取和控制操作。网格层在数据网格服务中充当数据层的代理,客户应用程序通过网格服务来操纵数据层的数据。

参考文献:

[1] ANTONIOLETTI M, ATKINSON M, BAXTER R, et al. The design and implementation of grid database services in OGSA-DAI [J]. Concurrency and Computation: Practice and Experience, 2005, 17(2): 357-376.

[2] 王威,刘卫东,宋佳兴.网络服务下的数据访问与集成规范架构[J].计算机应用,2005,25(10):2320-2321.
WANG WEI, LIU WEI-DONG, SONG JIA-XING. Research on the specification framework and key technology of WS-DAI [J]. Journal of Computer Applications, 2005, 25(10): 2320-2321.

[3] 刘小览,陈静.基于数据网格的异构信息共享平台的实现[J].计算机工程,2007,33(9):280.
LIU XIAO-LAN, GHEN JING. Implementation of heterogeneous information share platform based on data grid[J]. Computer Engineering, 2007, 33(9): 280.

[4] The University of Edinburgh. OGSA-DAI WSRF 2.2 User Guide[EB/OL]. (2006-04-12)[2006-08-20]http://www.ogsadai.org.uk/documentation/ogsadai-wsrf-2.2/doc/.

[5] The University of Edinburgh. Using OGSA-DAI behind an application-specific service[EB/OL]. (2005-06-10)[2006-07-12]. http://www.ogsadai.org.uk/documentation/scenarios/behind_application-specific_service/.

[6] The University of Edinburgh. Using OGSA-DAI with binary large objects (BLOBs)[EB/OL]. (2005-6-10)[2006-09-18] http://www.ogsadai.org.uk/documentation/scenarios/working_with_blobs/.

[7] NEIL HARDMAN, ANDREW BORLEY, JAMES MAGOWAN. OGSA-DAI: a look under the hood: architecture and database access[EB/OL]. (2004-06-10)[2005-03-23] http://www.128.ibm.com/developerworks/library/gr-ogsadai/.

[8] The Globus Project. GT4.0: Java WS Core [EB/OL]. (2005-04-29)[2006-01-12] http://www.globus.org/toolkit/docs/4.0/common/javawscore/.

[9] 周维,阎保平.网格数据访问中间件的设计与实现[J].计算机工程与应用,2005(34):90-91.
ZHOU WEI, YAN BAO-PING. Design and implementation of a data grid access middleware[J]. Computer Engineering and Applications, 2005(34): 90-91.

[10] 南凯,阎保平.扩展OGSA-DAI的数据集成框架及原型[J].计算机工程,2007,33(10):55.
NAN KAI, YAN BAO-PING. Data integration framework and prototype on OGSA-DAI extension[J]. Computer Engineering, 2007, 33(10): 55.

[11] 金宝轩,边馥苓.基于OGSA-DAI的空间数据访问和集成研究[J].测绘信息与工程,2005,30(3):21-22.
JIN BAO-XUAN, BIAN FU-LING. Spatial data access and integration based on OGSA-DAI[J]. Journal of Geomatics, 2005, 30(3): 21-22.

[12] 孙跃,陈春平,任江洪.基于OGSA的网格工作流管理系统研究与实现[J].重庆大学学报:自然科学版,2007,30(2):70.
SUN YUE, CHEN CHUN-PING, REN JIANG-HONG. Grid workflow management system research and the realization based on the OGSA[J]. Journal of Chongqing University: Natural Science Edition, 2007, 30(2): 70.

[13] 郭立文,杨扬,翟正利.基于OGSA的网格服务管理模型研究[J].计算机工程与应用,2007,43(1):9-10.
GUO LI-WEN, YANG YANG, ZHAI ZHENG-LI. A grid information service system ODIS based on OGSA[J]. Computer Engineering and Applications, 2007, 43(1): 9-10.

[14] CROMPTON S Y, MATTHEWS B M, GRAY W A, et al. OGSA-DAI and bioinformatics grids: challenges, experience and strategies[J]. Cluster Computing and the Grid, 2006(6): 8-10.

[15] ALPDEMIR N, MUKHERJEE A, GOUNARIS A, et al. A grid service for distributed querying on the grid [C] // Processdings of the 9th International Conference on Extending Database Technology. Bahia: [s.n.], 2004.

[16] ALTINTAS I, BIRNBAUM A, BALDR K K, et al. A framework for the design and reuse of grid workflows[C]// LectNotes Comput sc. Beijing: [s. n.], 2005.

(编辑 侯 湘)