目录型元数据在科学数据库系统平台中的应用

沈志宏　王龙潇

（中国科学院计算机网络信息中心，北京 100080）

摘　要　在科学数据库系统平台数据访问系统中，目录型元数据是实现上层数据集成以及底层数据访问的必要前提。本文介绍了目录型元数据的概念、结构与存储，并详细介绍了目录型元数据在数据访问模型中的重要作用。最后，本文着重介绍了目录型元数据在数据访问系统中的主要实现技术。

关键词　科学数据库 元数据 目录型元数据 数据访问 系统平台

1. 引　言

科学数据库系统平台是科学院“十五”信息化重大项目，旨在通过数据网格技术、WEB服务技术、分布式数据库技术等，构造统一的数据库系统平台，以实现科学数据库内数据资源的共享与集成。

科学数据库系统平台软件由数据访问子系统、信息服务子系统以及安全体系三部分组成。作为整个系统平台软件的核心部分，数据访问子系统面向数据集提供对各类数据资源（主要是关系型、可关系型）的管理、支持对数据的高级集成，并提供统一的访问接口支持对数据集的访问。

数据访问子系统如何访问底层的物理数据库？如何屏蔽掉各数据库之间的差异性，向上提供开放、统一的数据资源？这就必须依赖于目录型元数据。目录型元数据由系统平台中的信息服务子系统提供，它属于由科学数据库核心元数据标准（Scientific Database Core Metadata，简称 SDBCM）定义的结构描述信息。

本文将就目录型元数据的概念、结构与存储、来源与访问逐步作一阐述，并详细介绍目录型元数据在数据访问模型中的重要作用。最后，本文着重介绍在数据访问子系统的实现中针对目录型元数据采取的主要技术。

2. 目录型元数据的概念

2.1 元数据

在科学数据库标准规范体系中，元数据标准和技术是实现系统平台的主要手段之一。系统平台通过元数据，从而达到数据标准化以及数据共享、交换和整合。

根据科学数据库核心元数据标准，元数据主体一般包括数据集描述信息、数据集分发信息、元数据参考信息、服务描述信息以及结构描述信息。

2.2 结构描述信息

作为元数据主体信息之一，结构描述信息主要描述数据集所包含实体的结构信息，包括实体基本信息、属性信息、约束条件等内容。这里的实体结构信息主要是指关系数据库中的数据表及其属性信息，包含了关系型数据库数据字典中的大部分信息。

结构描述信息对终端数据用户是屏蔽的，主要提供给程序开发人员使用。

2.3 目录型元数据

目录型元数据主要来源于元数据标准中的结构描述信息，各种元数据之间彼此相互关联，并形成一定的层次结构，即程序可理解的目录结构。

根据元数据来源和含义的不同，目录型元数据可以分成物理型元数据和逻辑型元数据。物理型元数据主要关注数据的物理存储，这部分数据来源于对物理存储结构的提取和扩展；逻辑型元数据则主要关注数据的外在描述和组织形式，它在物理型元数据的基础上进行归纳整合，屏蔽内部物理存储的分布式和异构性，从而向上提供统一的逻辑数据接口，以达到最终的数据集成和共享。

物理型元数据主要包括：数据库元数据（Database）、数据表元数据（Table）、字段级元数据（Field）；逻辑型元数据则包括：数据集元数据 （Dataset）、属性元数据（ Attribute），此外，逻辑型元数据还向上扩展，形成联合数据集（Union Dataset），以支持更高级、更统一的数据集成。如：跨库检索就是一个应用联合数据集的例子。

逻辑型元数据主要面向数据集开发人员；而物理型元数据则主要面向于数据访问子系统的开发人员，供系统内部使用。

3. 目录型元数据的结构与存储

3.1 目录型元数据的层次结构

不同类别的目录型元数据在结构上还具有完整的层次，其结构如图 1 所示：

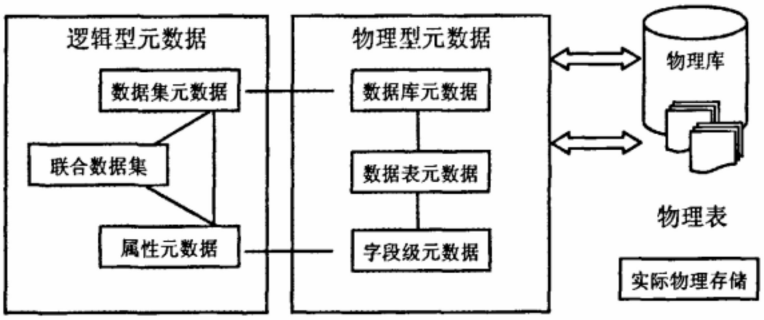

图 1 目录型元数据的结构

其中各种目录型元数据所包含的内容列举如下：

1. 数据集元数据（Dataset）：关于数据集的描述信息，主要包括：数据集 URI、数据库所在位置、端口号、用户名、用户密码、数据库类型、实例名称、数据库描述、数据库版本、有效标志、数据集检索点等；
2. 属性元数据（ Attribute）：关于数据集属性的描述信息，主要包括：属性中文标识名、属性名称、属性类型、属性单位、描述、属性来源、对应字段等；
3. 联合数据集元数据（Union Dataset）：关于多个具有相似属性的数据集的描述信息，主要包括：数据集 URI、各子数据集的 URI 等；
4. 数据库元数据（Database）：关于数据库的描述信息，主要包括：所在位置、端口号、用户名、用户密码、数据库类型、数据库驱动类型、JDBC URL、实例名称、库管理员、描述、数据库版本等；

5. 表级元数据（Table）：关于数据表的描述信息，主要包括：表中文标识名、表名称、表类型、表描述、表操作员、表创建者、表创建日期、主键字段、更新时间字段等；
6. 字段级元数据（Field）：关于字段的描述信息，主要包括：字段中文标识名、字段名称、字段类型、字段长度、字段顺序、字段描述、是否空、缺省值、字段辅助内容等；

除了层次关系外，各元数据之间还存在着复杂的关联关系，如：联合数据集与数据集之间的一对多关系，数据表与数据字段之间的一对多关系，等等。此外，在物理型元数据和逻辑型元数据之间，也保持着映射关系。如：数据集与数据库之间存在着多对一的映射，而字段和属性之间则存在着依赖于数据集的一一映射关系，等等。

3.2 目录型元数据的存储

目录型元数据采用统一的元数据存储方式，即系统中所有的元数据都存放在基于 LDAP 的目录中，进行统一管理。

需要指出的是，在实际存储中，并不需要保存数据库元数据。数据库元数据只是运行时元数据实体（Runtime Object），但它又是必须的，因为它是物理型元数据与逻辑型元数据之间的交界点，数据表与数据集都与之发生关联。由于数据集的元数据信息其实包含了对应数据库的大部分结构信息，在数据访问的具体实现中，程序内部只需要从数据集元数据中抽取出部分的信息来构造数据库元数据。

前文已经提过，数据集与数据库之间存在着多对一的映射，对于多个数据集对应着的同一个数据库，系统将仅仅只构造一个数据库实体，以消除数据的冗余。

4. 目录型元数据的来源与访问

4.1 目录型元数据的录入

元数据录入工具是元数据录入的主要用户界面，大部分元数据信息需要用户遵循标准格式的 Schema 手动输入，但是对于目录型元数据，大部分数据可以做到程序自动提取、扩展。根据目录型元数据类型的不同，一般完成目录型元数据的录入可以分解为如下几个步骤：

1. 指定数据集的连接参数：输入数据集的目录型元数据，即数据集的结构服务信息，如连接参数（数据库所在主机 ip、数据库管理系统 DBMS 类型、版本以及数据库名、用户名及密码）。用户在填写数据集连接参数之后，可以同时指定数据集的其它信息，如：检索点（主表）等；
2. 完成物理型元数据的自动导入：数据集管理人员输入数据集的参数之后，程序就会引导用户逐步建立相应的数据表、数据字段的元数据。这一过程是自上而下的：数据集（→数据库）→数据表→数据字段；
3. 完成属性元数据的定义：属性完成对字段的映射，用户可以指定需要映射的字段，程序就会自动将其中的字段元数据设置成属性元数据（如：自动将属性的显示长度设定为字段的长度），用户只需要在此基础上做一修改完善。这一过程是自上而下的：数据集→数据集属性；
4. 完成联合数据集的定义：将几个相同范畴的数据集统一成联合数据集，并指定其共有的属性。这一过程是自下而上的：数据集→联合数据集。

整个过程中，程序的自动导入功能体现在很多地方，而需要手动设置的则很少，这也是目录型元数据与其他类型的元数据的重要区别之一。

4.2 目录型元数据的访问

对目录型元数据的访问符合统一的元数据访问模式。程序开发人员通过使用统一的元数据访问 SDK，即可达到对目录型元数据的访问。

5. 目录型元数据在数据访问模型中的应用

数据访问模型是数据访问子系统的层次模型，作为结构描述信息，目录型元数据的应用贯穿于数据访问模型中的各个层次。

5.1 数据访问模型

从整个系统的观点看，数据访问从功能逻辑上可以分为以下三层：

1. 接口层：提供访问数据资源的 API，供应用程序调用。
2. 中间层：提供数据资源的定位与数据的集成。
3. 数据层：提供可管理的数据资源，也可以称为网格化的数据集。

数据访问模型如图 2 所示：

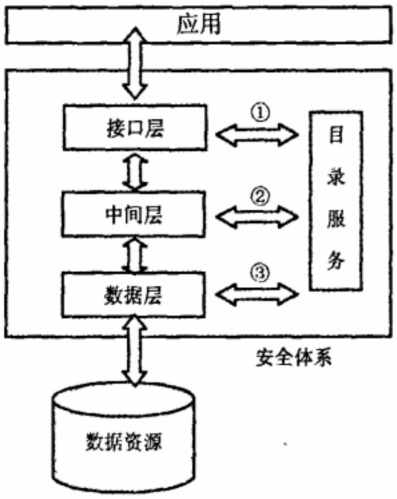

图 2 数据访问模型

5.2 目录型元数据在数据访问模型中的应用

从图 2 中的①②③标注可以看出，目录型元数据在不同层次中都发挥着关键的作用，这些作用具体体现在：

1.接口层：接口层向上提供访问数据资源的 API，目录型元数据在该层次提供简单的信息，以完成对外部用户访问请求的合法性检测和过滤，以决定是否有必要将该请求发送至中间层；

2.中间层：中间层提供数据资源的定位与下层数据的集成，目录型元数据在该层次提供数据集在各数据节点的定位信息以及联合数据集的属性信息，中间层根据这些信息将访问请求定位到合适的数据节点上去；

3.数据层：数据层在数据节点提供可管理的数据集，目录型元数据在该层次提供数据集的信息以及具体的物理连接信息。数据库访问内核采用 JDBC3.0，结合元数据，即可达到对 Oracle/MySQL/SQL Server/Access 等多种数据类型的统一访问。

总而言之，目录型元数据不仅是上层数据集成的核心基础，还为底层数据连接提供了关键信息。因此，目录型元数据是屏蔽数据之间的异构性、实现数据共享集成、完成数据底层

连接的必要前提。

6. 目录型元数据访问的实现技术

由于目录型元数据贯穿于数据访问的每一个环节中，所以必须采取一些有效的措施，来保证高效能的、完整的元数据访问。以下是现行系统中采取的实用机制：

6.1 对象 Cache 机制

在数据访问过程中，如何提供高性能的元数据对象访问和目录访问，一种很有效的途径就是使用对象池，即 Cache 机制。

如果每次都要从远程目录频繁地读取数据，不同层次、不同类别的元数据对象以及它们之间的关系的创建将是一个非常费时而且复杂的过程，对元数据的访问势必成为系统的瓶颈。这样的操作不仅会造成服务器的压力过大，而且也会严重地影响用户访问数据的效率。因此，必须引入元数据的 Cache 机制，在数据访问子系统开始启动的时候，系统自动读取远程目录服务器的数据，创建本地的元数据对象，并同时创建它们之间的关联，构造本地的目录结构（Local Catalog）。如上过程描述如图 3 所示：

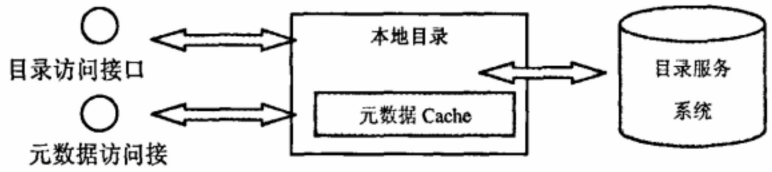

图 3 目录型元数据对象的 Cache 机制

这种 Cache 机制可以使元数据的访问性能达到最大化，大大减少与远程目录服务系统的交互次数。本机制的重点则主要在于保证 Cache 中的对象与实体之间的实时同步。

6.2 目录完整性机制

目录完整性也是目录服务系统中一个不容忽视的地方。如果出现关键元数据失效或者丢失，很有可能造成数据访问的终止，甚至造成整个数据集服务的瘫痪。

因此在数据访问子系统中引入目录完整性机制很有必要。系统在适当的时候可以在后台启动目录完整性程序，一次完整的完整性任务可能包含若干条任务项，这些任务项包括：连接可用性、属性映射字段可用性、数据集引用表的可用性等等。不同的任务项通过一定的方式部署至完整性任务中，目录完整性机制将会自动运行所有的任务项，以最终完成对整个结构目录系统的完整和修复。

7. 结束语

基于目录型元数据技术，我们目前已开发了通用数据访问工具1.0和2.0，并在各兄弟院所单位得到了成功的应用。该系统底层支持通用的数据库类型，如：Oracle、SQL Server、MySQL以及Access等；同时向上提供了统一的数据访问功能，包括：数据录入、数据维护、数据检索等功能。实践证明，通过目录型元数据技术进行各数据库应用的建立、配置和管理非常方便快捷，大大缩短了系统的开发时间。

数据访问子系统是科学数据库系统平台中的一个重要环节，如何有效地组织和管理各科学院所分布的数据并屏蔽其中的异构性以达到统一访问的要求，绝大部分就依赖于由元数据核心标准支持的各种元数据，其中目录型元数据则是整个访问过程的基础。另一方面，数据

访问子系统对目录型元数据提出了更高的要求，如何实现完整的目录层次结构（通用数据访问工具只完成了数据集以下层次的访问），并引入数据访问层次模型，进一步地统一和规范数据访问的流程，这些问题的研究是实现数据访问子系统中的关键内容。因此，研究目录型元数据及其实现技术对数据访问子系统甚至整个系统平台软件的研发有着至关重要的作用。

参 考 文 献

[1] 中国科学院科学数据库核心元数据标准（版本号：1.1），中国科学院计算机网络信息中心，科学数据库中心，2003 年 8 月

Catalog Metadata and Its Application in SDB System Platform

Shen Zhihong　Wang Longxiao

(Computer Network Information Center, Chinese Academy of Science, Beijing 100080)

Abstract　Catalog metadata play an important role in the Data Access Subsystem of SDB System Platform. This paper introduces the conception, structure and storage of catalog metadata, and points out the key function of catalog metadata in Data Access Model. The paper also gives emphasis on description of the key technologies of catalog metadata applied in Data Access Subsystem.

Key words　scientific database; metadata; catalog metadata; data access; system platform