Computer Science and Application 计算机科学与应用, 2016, 6(3), 126-131
Published Online March 2016 in Hans. http://www.hanspub.org/journal/csa
http://dx.doi.org/10.12677/csa.2016.63016

Hans汉斯

Design and Implementation of Heterogeneous Database Engine for Enterprise Application System

Zhe Zhao

Xi'an No.1 Middle School, Xi'an Shaanxi
Email: 1140826296@qq.com

Received: Feb. 9th, 2016; accepted: Feb. 27th, 2016; published: Mar. 2nd, 2016

Abstract

With the continuous development of enterprise information management, many industries, unit or organization, internal departments have gradually realized the business, network management. How simple and effective access to and integrate heterogeneous database system, reduce duplication of code, shorten the software development cycle, and improve software maintenance is particularly important. In order to solve this problem, we design and implement a heterogeneous database engine by O/R Mapping technology to solve the problem of cross platform database access based on the O/R model. The engine we design supports UNIX, windows and other operating systems, and also integrates with Oracle, DB2, Server, My SQL, SQL and other database. After the actual application, the performance is stable and reliable. The desired goal is achieved.

Keywords

Heterogeneous Database Engine, Database Access, O/R Mapping, ADO.NET, SQL Server, My SQL

面向企业应用系统的异构数据库引擎设计与实现

赵　哲

西安市第一中学，陕西　西安

文章引用: 赵哲. 面向企业应用系统的异构数据库引擎设计与实现[J]. 计算机科学与应用, 2016, 6(3): 126-131.
http://dx.doi.org/10.12677/csa.2016.63016

Email: 1140826296@qq.com

收稿日期：2016年2月9日；录用日期：2016年2月27日；发布日期：2016年3月2日

摘　要

随着企业信息化管理不断发展，许多行业、单位或机构、部门内部已逐步实现了业务、信息的网络化管理，如何简洁有效地访问和集成异构数据库系统，减少重复编码、缩短软件的开发周期、提高软件的可维护性尤为重要。为解决这一问题，本文利用O/R Mapping技术基于.NET平台设计并实现了异构数据库引擎，较好地解决了基于对象/关系模型的跨平台数据库访问问题。该引擎支持Unix、Windows等操作系统，而且可与Oracle、DB2、SQL Server、My SQL等数据库进行集成。经实际应用，性能稳定可靠，达到了预期的目标，取得了良好的应用效果。

关键词

异构数据库引擎，数据库访问，O/R Mapping，ADO.NET，SQL Server，My SQL

1. 引言

计算机技术和信息化网络技术飞速发展，企业各种应用系统逐渐增多，这些应用系统的数据库信息对于企业的管理非常重要，这些信息往往具有多样性、异构性和复杂性，非常不利于进一步运用而获得更大的企业综合效益。对于分散异构的信息资源进行集成，使得原有的各种异构数据库数据能够整合到一起，提供给用户透明统一的接口，用户就像是在使用一个数据库一样而不必去考虑其中各种数据的差异[1] [2]。在信息系统的集成和构建过程中，存在大量的异构数据库系统，给企业软件系统的整合应用带来了一定的困难。如何简洁有效地访问以及将异构数据库系统集成化，以减少重复编码、缩短软件的开发周期、提高软件的可维护性已成为信息系统开发的重要课题，所以异构数据库访问引擎的开发非常必要。

异构数据库的访问是数据库集成的重要环节[2]。2002 年 Maurizio Lenzerini 提出用<G,S,M>这一三元的向量来描述数据集成。关于异构数据库的访问，华中科技大学开发了 DM3 系统，通过提供的数据库转换工具和 API 接口来实现数据库转换和数据透明访问。

以关系数据库为数据存储方式开发面向对象的软件时，由于关系概念与面向对象概念的迥然差异，使它们之间产生了访问方式不匹配的问题。为了解决这个问题，出现了许多新技术，在软件开发领域中最为炙热的对象关系映射(O/R Mapping)技术。本系统利用 O/R Mapping 技术开发了基于 ADO.NET 的异构数据库访问引擎，该引擎支持各种开发工具及数据库系统，可采用面向对象的方法开发平台无关的数据库应用程序[3] [4]。

2. 异构数据库引擎框架设计

异构数据库引擎的总体设计目标是：为应用软件开发人员提供一个通用数据库访问和存取接口，使其能够方便地对异构数据库进行操作而不需要太多地了解底层数据访问的技术，并能够按照面向对象方法进行数据库无关的软件设计开发。

ADO.NET 异构数据库访问技术.NET Framework 中有四种提供程序，每一种数据提供程序都包含各自的对象，若要使用这种数据库访问技术，无法实现数据库平台的通用性访问[5] [6]。

应用程序需要对数据库进行访问的时候，不需要直接调用 ADO.NET 的一系列对象方法，而只需调

用异构数据库引擎提供的接口，由异构数据库引擎去响应应用程序的请求，并自动生成相关的SQL语句，然后根据请求调用 ADO.NET 相关的方法，最后把结果反馈给应用程序。在整个访问操作过程中，ADO.NET 及底层数据库访问技术对程序员来讲是透明的，不必掌握 ADO.NET 相关知识，即可正常进行数据库应用系统的开发[7]。

异构数据库引擎设计采用了对象关系映射 O/R Mapping (Object Relation Mapping)技术、软件分层和复用的思想。O/R Mapping 技术通过在关系数据库之上加入对象关系映射引擎，让现有的关系数据库得到最大程度上的“可持续发展”，既保护了用户的已有投资，又使得熟悉关系技术的开发群体能够平滑过渡到新的阶段[8]。分层是一种解决复杂问题非常有效的方法，通过分层能够逐渐地将复杂问题分解，给每一层赋予的功能单一，层与层之间的交互信息单一，从而使得系统框架清晰，易于实现、可扩展、可维护和可复用。异构数据库引擎总体结构设计如图 1 所示。

- 对象表示层：表示应用系统中业务对象与数据库中关系表的映射，该层设计的优劣直接关系到上层应用能否方便地使用业务对象，进行业务逻辑的处理以及下层持久化操作能否高效便捷地运行。
- 对象控制层：控制对象表示层的实体类，定义操作实体类的相关运算，包括将数据实体的状态保存或更新到数据库中。
- 数据访问层：负责封装多种数据访问对象，目的是当底层的数据库发生改变的时候，高层应用系统不会受影响，进而提高异构数据库引擎的通用性。

3. 异构数据库引擎设计

本系统研究的异构数据库访问引擎在数据库底层上基于 ADO.NET 和其组件技术来实现，提供了各种主流数据库系统，如 Oracle、SQL Serve 等的通用访问接口，并支持 ODBC 通用数据源，以实现 Sybase、DB2、Informix 等数据库系统的访问。异构数据库访问引擎对外提供了统一的接口与访问方式，可以实现了快速、简洁的数据库面向对象访问技术，支持跨平台数据库系统，从而简化了数据库应用软件的开发，也大幅度地提高了软件开发效率。

3.1. O/R Mapping 技术

很多企业应用都使用面向对象语言开发，而用关系数据库作为底层数据库。由于对象范例和关系范例结构上的不同，导致了对象模型和关系模型之间的阻抗不匹配。O/R Mapping (Object-Relational Mapping)指的是在对象模型/关系模型两种途径之间和支持这两种途径的系统之间的一个转换,即把对象的属性保存在关系数据库中，把对数据库的访问封装起来。我们在操作数据库时，不再和复杂的 SQL 语句打交道，只要像操作对象一样操作它就可以了，是解决对象和关系之间阻抗不匹配问题的一种有效方法。在采用关系数据库的面向对象程序开发中，用户界面层和业务逻辑层是面向对象的，而当对象状态发生变化的

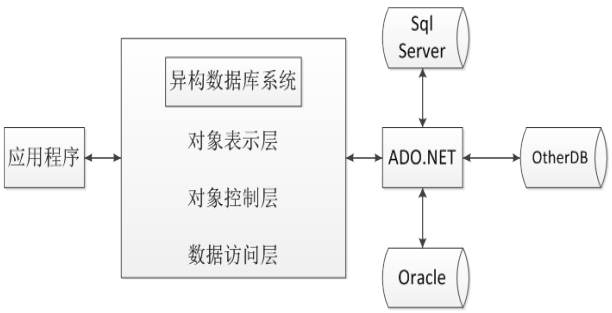

Figure 1. Frame of heterogeneous database engine
图 1. 异构数据库引擎框架图

时候，需要将对象的状态保存在关系数据库中[9]。在这种情况下，如果不使用 O/R Mapping 技术，就需要在程序的数据访问层写很多代码实现在关系数据库中保存、修改、删除对象信息，并且这些代码往往是重复的，存在很多近似的通用模式。如果采用 O/R Mapping 技术的话，就可以通过 O/R Mapping 异构数据库引擎来完成对象状态的持久化操作，通过 O/R Mapping 异构数据库引擎来保存、修改、删除和读取对象，O/R Mapping 异构数据库引擎还可以生成 SQL 语句等，开发者只需要关心对象即可。

3.2. ADO.NET 异构数据库访问技术

ADO.NET 是一种关系数据，根据单独的数据操作能够对已经离散的数据进行访问。在数据处理形式上，它构建了合理的数据处理环境。ADO.NET Framework 中包含四种数据提供程序，即 SQL Server .NET Framework、Oracle .NET Framework、OLE DB .NET Framework 和 ODBC .NET Framework。而每种数据提供程序又包含各自的对象。如果要利用 ADO.NET 进行数据库访问，则必须根据实际使用的数据库类型来选择.NET 数据提供程序，然后在应用程序中使用，写定了的程序代码只能适应预先确定的数据库系统，无法实现数据库平台的通用性访问。只有重复写代码才能适应每个特定数据库的访问方法，这不仅大大增加了编码人员的工作量，而且无法实现数据库平台无关的程序设计，更不易于系统维护。

3.3. 异构数据库引擎工作原理

本文运用了软件分层的思想来设计异构数据库引擎，主要包含对象表示层、对象控制层、数据访问层三层。对象表示层采用了的 ADO.NET 的 Data Set 类和基于 XML 的配置文件，直接影响用户的使用。Data Set 是 ADO.NET 中的数据对象，它可以包含多个数据表(Data Table)及其关系。Data Set 的结构可以与关系数据库进行很好地映射。在 XML 配置文件中定义了各个实体类与关系数据库之间的映射关系，即定义了特定于某个实体类的 Data Table，在运行时由相应的类构造方法根据 XML 配置文件生成特定的实体类[10]。对象控制层主要实现了 4 个持久化的核心方法，分别是 Insert()、Update()、Delete()和 Search()方法，实现对数据库的新增、修改、删除及按对象标识进行查找，它们的作用是自动生成 SQL 语句。数据访问层可以支持多种类型的数据库系统，在 ADO.NET 中访问不同类型的数据库时需要使用不同的命名空间及数据访问类，因此应利用简单的工厂模式来实现数据库类型的自动切换。

4. 异构数据库引擎实现[11] [12]

数据库引擎模块是一种主要的实现数据库访问的方法，它主要依赖于系统配置模块运行实现。系统的设计原理参考如图 2，其中数据库引擎 Web Service 接口用于扩展数据库引擎，实现跨应用平台的调用。

4.1. 系统配置模块

该模块的功能是配置异构数据库引擎的参数信息。如果用户开发的是 Web 程序，这些参数信息定义在 web.con fig 文件的<app Settings>配置节中，如果用户开发的 Windows 程序，则把这些参数信息定义在文件名与应用程序名相同、扩展名为.con fig 的 XML 文件的<app Settings>配置节中。

4.2. 异构数据库引擎数据访问层模块

客户端请求一般分为四大类：数据查询(Select)、数据定义(Create, Drop, Alert)、数据操纵(Insert, Update, Delete)、数据存储。根据客户的请求方式，该引擎有相应的处理方法：客户在每次发出请求时，只需调用该模块提供的方法，传入正确的参数，该模块就会把处理结果反馈给客户，完成此次请求。若执行错误，则返回错误代码。

实现该引擎的关键技术：

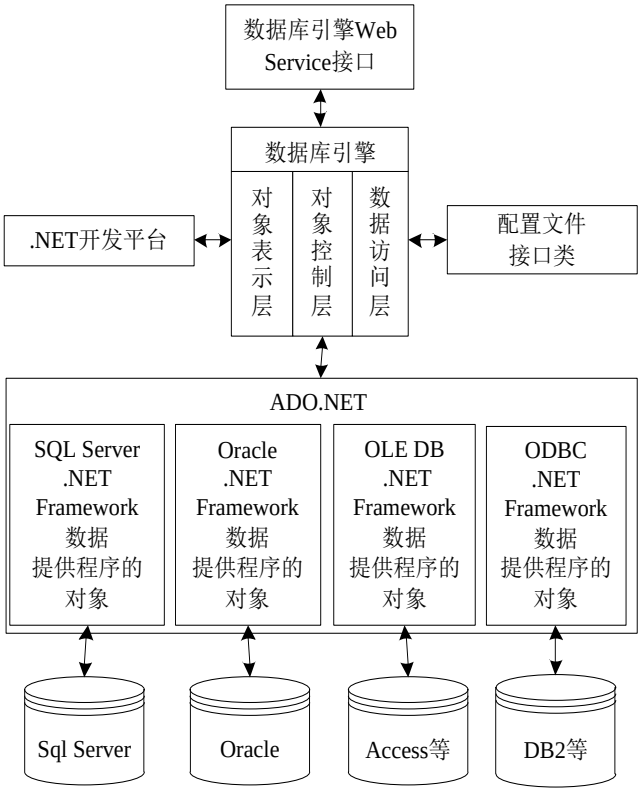

Figure 2. System schematic of heterogeneous database engine
图 2. 异构数据库引擎系统原理图

1) 数据访问对象再定义：系统设计了 Connection 对象，用于连接数据库；Command 对象用于执行命令；Parameter 对象用于处理 SQL 参数。利用“系统配置模块”从配置文件中获取数据库类型后，应针对 SQL Server、Oracle、Access、ODBC 等特定数据库访问类型，对 Connection、Command、Parameter 对象进行再定义，并设置相应参数，以实现特定数据库的连接与访问。

2) 数据库的连接与断开：调用“系统配置模块”提供的方法获取各种参数并生成数据库连接字符串，建立与数据库的连接。系统采用了 ADO.NET 提供的连接池技术，可在配置文件数据库连接模板中设置连接池的大小。为了提高系统性能，数据库关闭方法并不真正断开与数据库服务器的 TCP 连接，而是将连接资源释放到 ADO.NET 连接池中，以便其它程序调用。

3) 数据库查询和处理：该模块主要依赖于 ADO.NET 数据提供程序的 Command 对象实现，该对象提供了 Execute Reader、Execute No Query、Execute Scalar 等方法以完成特定数据操作。数据访问层主要实现了以下通用数据库访问方法：Exec Single Value 用于获取单值，Exec No Query 用于执行无返回值的操作，Exec Query 返回数据集的查询，Exec Insert 执行插入操作并返回插入行 ID，Modify Blob 修改 Blob 值，Get Blob 获取 Blob 值。通过以上方法，可以规范地实现数据库的跨平台访问。

4.3. 异构数据库引擎的 Web Service 接口

以上方案很好地解决了.NET 开发平台中异构数据库的访问问题，构建其 XML Web Service 接口是为了能使异构数据库引擎适用于任何开发工具。在 Web Service 服务程序上实现异构数据库引擎主要功能，以统一开放的 HTTP、XML、SOAP、WSDL 等标准协议，提供异构数据库引擎的跨平台、跨防火墙调用 [13] [14]。

5. 结论

以上设计很好地实现了.NET 开发平台中异构数据库访问的问题，运用 O/R Mapping 技术完成了异构数据库引擎的设计和实现，该引擎支持 Unix、Windows 等操作系统，而且与各种通用数据库都可进行集成，为中间件技术未来的发展也奠定了一定的基础。经过在企业应用程序的运用，能够为企业用户减少一定工作量，缩短软件开发周期，提高软件质量，产生一定的经济效益。

参考文献 (References)

[1] 刘飞. Linux 下异构数据库访问引擎的设计和实现[D]: [硕士学位论文]. 上海: 复旦大学, 2011.

[2] 徐爱萍, 宋先明, 徐武平. 分布式异构数据库集成系统研究与实现[J]. 计算机工程与科学, 2015(10): 1909-1916.

[3] 赵美丽. 基于 CDMA 网络的短信中间件研究与实现[D]: [硕士学位论文]. 西安: 西安科技大学, 2008.

[4] 杜蕊. 智能消息中间件的研究与应用[D]: [硕士学位论文]. 哈尔滨: 哈尔滨理工大学, 2012.

[5] 曾国林. 异构数据库访问与集成模型的应用研究[D]: [硕士学位论文]. 广州: 广东工业大学, 2011.

[6] 李娜, 王维哲. 基于 CORBA 的异构数据库访问中间件的研究与实现[J]. 计算机应用与软件, 2010, 27(5): 162-164.

[7] 黄学彬, 赵春, 郑伟. 异构数据库高效数据交换引擎设计[J]. 西南师范大学学报(自然科学版), 2014, 09:100-108.

[8] 陈执. O/R Mapping 技术在.NET 框架中的应用[J]. 中国科技信息, 2011(14): 100-101.

[9] Fowler, M. (2003) Patterns of Enterprise Application Architecture. Addison-Wesley, Boston.

[10] Yan, L., Fu, X.Y. and Chen, Y.L. (2011) Data Exchange of Heterogeneous Database Based on XML. *Energy Procedia*, **13**.

[11] 胡顺扬. 基于 XML 异构数据库访问中间件技术研究与实现[D]: [硕士学位论文]. 金华: 浙江师范大学, 2009.

[12] 赵美丽, 朱宇. 基于.NET 的异构数据库引擎设计与实现[J]. 科技信息(科学教研), 2007(23): 75-76.

[13] Ferrara A. and MacDonald, M. (2003) NET WEB 服务编程. 天宏工作室译. 北京: 清华大学出版社, 2003: 12-13.

[14] 皮莹莹. 处理海量数据的异构数据库访问中间件的设计与实现[D]: [硕士学位论文]. 南京: 南京邮电大学, 2013.