

# Windows NT 下的分布式数据访问

李先锋 曹根宝 (北方交通大学计算机系 100044)

**摘要:**本文介绍 Windows NT 上的基于客户/服务器结构的分布式数据访问,详细叙述了 Windows NT 的客户/服务器模型、SQL Server 数据库管理系统、开放式数据互连,最后介绍在 VB 中如何实现分步式数据访问。

**关键字:**客户/服务器 数据库管理系统 开放式数据互连

## 一、概述

Windows NT 是 Microsoft 公司为 90 年代研制的一种新的操作系统。它作为下一代 Windows 系统,处于 Windows 家族的顶端,沿用了 Windows 图形界面,内置网络功能,而且是 Microsoft 的第一个支持 32 位 Windows API 的操作系统。作为多用户操作系统,为网络上的众多用户需求服务。它不仅能提供基于网络的操作系统功能,如文件服务器、打印服务器和系统管理功能,而且能提供高性能的应用程序功能,如数据库服务。Microsoft SQL Server 就是一运行于 Windows NT Server 上的数据库服务器。

Microsoft SQL Server 是用于分布式客户/服务器计算环境的一个可伸缩、高性能、功能强的关系数据库管理系统。一个综合的分布式管理框架结构允许集中管理组织所有的数据库服务器,并且实现分布式环境命令构成的远程操作的自动执行,并且向各种查询和数据源开放。

## 二、Windows NT 的客户/服务器模型

Windows NT 的结构采用了客户/服务器模型。客户/服务器结构主要是将一个应用分解为前台客户部分和后台服务器部分。服务器指能提供某种服务的任何程序。每个服务器是一个运行在用户态上的进程,它检查是否有客户请求某项服务,如有,则完成请求任务,结果返回请求者。客户可以是另一个服务器,也可以是一个应用程序(一般与用户打交道),它通过发送一个消息给服务器来请求一项服务,等待回答。消息传送机制,把请求消息传给服务器,该服务器执行操作,用另一个消息把结果返回客户。

采用客户/服务器模型,每个服务器以用户态进程方式运行,独立地在分配给它的内存中运行,防止了受其他

进程的影响,并容易增加新的服务器。在网络环境下,计算机以客户/服务器模型为基础,并且使用消息通信,本地服务器可以容易地发送消息给代表客户应用程序的远程计算机,客户不需知道某些请求是在本地还是在远地得到服务,因此完全适宜于分布式计算模型。在这种客户/服务器模型下,应用程序如 Microsoft SQL Server 允许用户提交一个对远程机器进行检索和排序的查询请求,而处理完后,只将结果返回用户。一方面使系统处理能力最低下的部分——网络部分的负荷减到最小,充分发挥网络的能力;另一方面通过利用远程的、通常是性能高的计算机处理周期来扩展单用户工作站能力。

## 三、Windows NT 上的数据库服务器平台

Microsoft SQL Server 是一个高性能、多用户关系型数据库管理系统(RDBMS),支持分布式客户/服务器计算环境,具有集中式数据完整性能力和事务处理能力,事务处理保证了数据库一致性和可恢复性,对典型的系统故障,SQL Server 提供了不需系统管理员干预的完全自动的恢复机制。存储过程加密为基于服务器的过程应用程序代码提供了完整性和安全性。存储过程又叫系统过程,每一个 SQL 命令发送给 SQL Server 作处理时,服务器对 SQL 命令作语法检查,检查引用的对象是否存在,检查请求者是否有权限去执行这个命令,当检查通过后,SQL Server 一次执行语句,并将结果由网络传送到客户端。否则,返回失败代码到客户端。客户端在未收到失败代码的情况下,接收结果,然后对结果进行处理。

Windows NT 上的 SQL Server 使用网络库(Net-library)在客户与 SQL Server 之间传递网络包。网络库为应用程序接口(API)与物理网络的进程间通信(IPC)机制之间提供一个透明的接口,通过动态连接库(.DLL)实现,执行按指定的协议进行的通信所需的网络操作。

#### 四、开放式数据库互连(ODBC)

客户应用程序访问 SQL Server 中数据的应用程序接口(API)有两种:ODBC(OpenDatabase Connectinng)和 DB-library。开放式数据库互连(ODBC)是一种全方位的数据库连接的 API,用以确保在异构环境中存取数据。ODBC 对于存在于不同的数据库中的数据而言是一种中立的存取方式。本文选择 ODBC 作为 API 介绍应用程序如何通过 ODBC 对 SQL Server 上数据库进行数据操作。

ODBC 是一个针对普通关联数据库的 C 语言接口。它提供最大限度的互操作性,允许单个应用存取多数据库,应用开发者可以开发、移植应用而不必局限于某特定的数据库管理系统。ODBC 通过强迫所有客户都遵循同一个接口协议标准,成功地实现了互操作性。ODBC 驱动程序可以自动翻译某个数据源命令。

ODBC 由三部分组成:①应用(Application),它调用 ODBC 函数与数据源相联,发送和接收数据,然后断开;②驱动管理(Driver Manager),它向应用提供某种信息,必要时动态加载驱动程序以提供参数和传输状态检测;③驱动程序(Driver),它是一个处理 ODBC 函数调用,并在应用与特定的数据源之间交换信息的动态连接库(.DLL)。一个应用不局限于与一个驱动程序通信,它可以通过不同的驱动程序与多个数据源产生多个关联,也可以通过一个驱动程序与同一个资源产生多个关联。

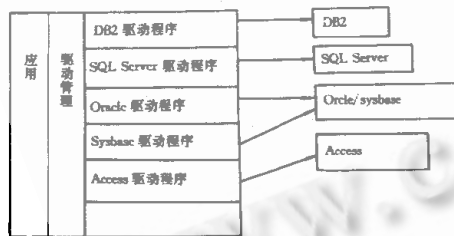


图 1 ODBC 的结构

访问 SQL Server 数据库,必须安装 SQL Server 驱动程序。SQL Server 驱动程序在应用和指定的 SQL Server 之间交换信息,它把标准的 SQL 语法翻译成目标数据源自身的 SQL 语法。图 2 是应用通过 ODBC 接口访问 SQL Server 的结构。

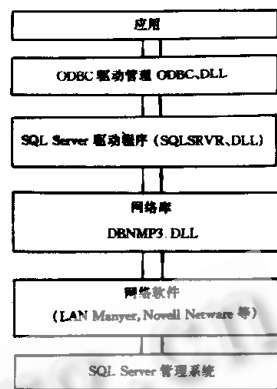


图 2 应用通过 ODBC 访问 SQL Server 的结构

#### 五、客户应用平台

Visual Basic 是一个强有力的、可视化的程序开发环境,利用它可以简单、高效地开发 Windows 3. x 应用程序,而且 VB 中提供了开放式数据库连接(ODBC)功能,可通过直接方式或建立连接方式使用并操作外部数据库。故本文选择 VB 作为客户应用平台,介绍在应用程序中如何通过 ODBC 访问 SQL Server。

在 VB 的开发环境中,数据库访问采用面向对象的思想,每个对象有一组属性和一组方法,应用程序通过生成对象的实例,使用对象的方法,操作和访问数据库。

##### 1. 与 SQL Server 建立连接

应用程序通过调用函数 OpenDatabase,生成对象 Database 的实例,可以打开一个本地库,也可以与远程网络数据库建立连接。其格式为:

```
OpenDatabase(databasename[, exclusive[, readonly[, connect]])
```

此函数返回一 Database 对象的实例。如果要通过 ODBC 接口连接 SQL Server 上的数据库,则要求 connect 的格式为:

```
ODBC;dsn = Server;uid = username;pwd = password;database = databasename
```

例如,与 SQL Server 上的一个学生\_课程数据库 studcurs 建立连接。

```
Dim Db AS Database
```

```
Dim coun AS String
```

```
coun = "ODBC;dsn = SQL Server1;uid = SA;pwd = ;database = studcurs"
```

```
set Db = OpenDatabase("", False, False, coun)
```

其中,数据库服务器的名字是 SQL Server1,用户标志是 SA,password 为空,数据库的名字就是 studcurs。

关闭数据库或断开与 SQL Server 中的数据库的连接,要调用 Database 对象的方法 Close。例如上例,断开与 SQL Server1 中 studcurs 数据库的连接。

```
Db.Close
```

## 2. 数据库操作和查询

在 VB 环境中,应用程序打开数据库表或建立查询,通过调用 Database 对象的方法 CreateDynaset,生成 Dynaset 对象的实例来实现。其格式为:

```
CreateDynaset(Source[, Option])
```

其中, Source 可以是一个库表的名字,或者是一个 SQL 语句。Option 参数是一些标志位的组合,要以 ODBC 方式访问 SQL 服务器,应加上常数项 DB-SQL-PASSTHROUGH,即 SQL 语句的执行是由 SQL 服务器完成,而不是在本地完成的。该方法返回一个对象 Dynaset 的实例。

例如,在 studcurs 数据库中有一个存放学生信息的表 S,打开此表为:

```
Dim Db AS Database, ds AS Dynaset, coun AS String
coun = "ODBC;dsn=SQL Server1;uid=SA;pwd=;
database=studcurs"
```

```
set Db = OpenDatabase("", False, False, coun)
```

```
set ds = Db.CreateDynaset("S")
```

再如,数据库 studcurs 中有一个学生选课信息的库表 C-S,在此库表上查询学号为 s1,成绩不及格的课程的课程名及成绩:

```
Dim str AS String
```

```
str = "select CN, G from C-S"
```

```
str = str + " where (SN='s1')and(G<60)"
```

```
set ds = Db.createDynaset ( str, DB-SQL-
PASSTHROUGH)
```

对记录集的记录进行操作,使用 Dynaset 对象的一组方法。例如,在学生信息表 S 中增加一名学生("00001", "张三", "计算机系", 18):

```
ds.Addnew
```

```
ds("SN") = "00001"
```

```
ds("SName") = "张三"
```

```
ds("SD") = "计算机系"
```

```
ds("SA") = 18
```

```
ds.Update
```

记录集指针的定位,使用 Dynaset 对象的一组方法: MoveFirst, MoveNext, MovePrevious, MoveLast。还可以利用下述方法进行记录的查找:

```
FindFirst Criteria
```

```
FindNext Criteria
```

```
FindPrevious Criteria
```

```
FindLast Criteria
```

其中, Criteria 相当于 select 语句中的 where 子句。如果找到相匹配的记录,则 Dynaset 对象实例的 NoMatch 属性的值为 False,否则为 True。例如,在学生信息表 S 中,查找姓名为“张三”的学生:

```
Dim s AS String
```

```
s = "SName = '张三'"
```

```
ds.MoveFirst
```

```
ds.FindFirst s
```

```
Do Until ds.NoMatch
```

```
...
```

```
ds.FindNext s
```

```
Loop
```

上面对 VB 的数据访问的强大功能作了简单的介绍。我们利用 VB 的强大数据访问功能,再有 SQL Server 的安全性和数据集集成性,就不难开发出用于管理数据的强有力的客户/服务器结构的客户平台。

## 六、结束语

Windows NT 作为 90 年代先进的操作系统, Microsoft SQL Server 作为功能强大的关系型数据库管理系统,因此,应用二者基础上的应用程序开发环境,将能高效、快速地开发出 Windows 环境下功能强大的应用程序,并且安全、可靠而且更容易地实现分布式数据访问和数据管理。因而必将有强大的生命力。

## 参考文献

- [1] 曹康等译,网络服务器的技术支持,学苑出版社,1994年9月
- [2] 陈河南译,网络技术支持基础,学苑出版社,1994年10月
- [3] 程渝荣,Windows NT 技术内幕,清华大学出版社,1993年  
(来稿时间:1996年9月)