

# 数据库应用系统中数据远程传输方法探讨

丁明吉 (辽河油田信息中心 124010)

**摘要:**本文综合地分析了数据库应用系统系统中数据远程传输的几种方法,并实现了相应的算法,同时论述了几种数据传输方法的优缺点。

**关键词:**数据传输 数据库链路 SQL \* PLUS PowerBuilder

## 一、概述

随着信息产业和信息技术的不断发展,越来越多的信息需要处理,需要交流,而处理信息的数据库应用系统也由原来的集中式管理向分布式管理转移。分布式数据库应用系统是基于广域网环境下 Client/Server 体系结构,这样的应用系统中,不仅是一个服务器的多个客户间数据的共享与互访,而且经常存在着不同服务器之间大量的数据传输,数据在传输过程中常存在以下几个问题:① 数据传输效率,② 界面友好性及操作方便性,③ 通用性及可移植性,④ 数据的安全性。这些问题均会影响到数据的最终传输效果,本文针对这些问题详细地研究了儿种不同传输方法,并且比较了各自的优缺点。

目前,分布式数据库应用系统的环境平台多数是:服务器端选择 Oracle、Sybase、Informix 数据库管理系统;客户端选择 PowerBuilder、Delphi 等开发工具;网络协议选择 Tcp/I<sub>p</sub>。不管选择哪一种环境平台所开发的应用系统,数据的远程传输,总的可以分为以下二种:① 直接操作远程数据库表,② 间接操作远程数据库表。本文以 Oracle 数据库管理系统和 PowerBuilder(PB)为环境平台对这二种传输方法进行讨论。

## 二、直接操作远程数据库表

这种方法是由某一客户发出请求,本地服务器执行这一请求,然后直接存储或更新远程服务器上的数据表。直接操作远程数据表按照执行策略的不同又分为三种类型:① 利用 PB 内的多事务性,② 利用数据库链路,③ 利用 PB 调用 SQL \* Plus 产品。

### 1. 利用 PB 的多事务性直接操作远程表

在 PB 环境内,提供了一种事务(Transaction)类型的对象,事务需在客户的应用系统中定义,在一个应用中可以同时定义多个事务,而一个事务对应一个数据库服务器的某一帐户,这样一个应用就可以同时连接多个不同的数据库帐户(既可是本地的,也可是远程的),因此,一个事务完成从本地服务器读数据,其他的事务完成向远

程服务器上写数据。为了保证数据的成功传输,需先删除远程服务器上相同的数据,然后再写入。否则违反数据的唯一性约束,Oracle 数据库管理系统将拒绝写入数据。数据传输中所用到的 SQL 语句是 Select(取数据)、Insert(写数据)和 Delete(删除数据)。下面是具体传输的实例:

在本例中定义二个 Transactions,一个是 PB 内默认的 SQLCA,连接到本地服务器的 Scott 帐户;另一个是 R \_ ta(Remote \_ transaction),连接到远程服务器的 Scott 帐户。在应用程序的初始阶段 SQLCA 已经连接好。相应的 Script 程序如下:

```
long empno, ii
string ename, salary
Declare c _ emp cursor for select empno, ename, salary
from emp where deptno = :dno ;
/* 定义只传送部门号等于 dno 的雇员信息 */
Transaction R _ ta /* 定义连接远程帐户的 Transac-
tion */
R _ ta = create transaction
R _ ta.DBMS = "O71 ORACLE v7.1"
R _ ta.Database = ""
R _ ta.LogID = "scott"
R _ ta.LogPass = "tiger"
R _ ta.UserID = "scott"
R _ ta.DBPass = "tiger"
R _ ta.ServerName = "@remote _ node" /* remote _
node 是指向远程接点的服务器名 */
connect using R _ ta ;
if R _ ta.DBHANDLE() = 0 then
    ii = messagebox("系统提示:", "口令连接串错、
网络或服务器有问题!", & Exclamation!, OKCancel!,
2)
Else
Delete from emp where deptno = :dno Using R _ ta ;
```

```

Commit using R_ta ;
open c_emp ;
fetch c_emp into :empno, :ename, :salary ;
do while sqlca.sqlcode = 0
    insert into emp values(:empno, :ename, :salary) using R_ta ;
    fetch c_emp into :empno, :ename, :salary ;
loop
close c_emp ;
Commit using R_ta ;
Destroy R_ta ;
End If

```

## 2. 利用数据库链路直接操作远程表

Oracle 数据库管理系统中提供了建立数据库链路的功能,数据库链路(database link)定义了到远程数据库的“路径”,链路名字就是要指向的远程数据库的全称数据库名。因为要对远程数据库任何对象作存取,都要以远程数据库中认可的用户的名义来建立会话并在会话中实际进行存取。在进行数据传输前必须在服务器上先建好相应的链路名,假设我们在本地服务器上已建好一个名为 Dbl\_re1(指向 Remote\_node 上的 Scott 帐户)的链路,然后在 PB 的 Script 内使用链路名先删除>Delete)远程服务器上要传送的数据,再把要传送的数据插入到远程服务器上。具体的实例如下:

### ① 在 SQL \* PLUS 内建链路

```

Create Database Link Dbl_re1
Connect to scott Identified by tiger
Using '@remote_node' /* 以 SQL * NET V2 建立链路 */

```

### ② 传送数据

```

Delete From emp@Dbl_re1 where deptno = :dno ;
Insert Into emp@Dbl_re1 Select * From emp Where deptno = :dno ;
Commit ;

```

注:实际的信息管理系统中,还存在很多微机服务器,其上安装的是 Oracle 7.0.1.13 版本的管理系统,这样不能使用这种方法由 Oracle 7.0.1.13 版本微机服务器向高于此版本的 Unix 或 NT 服务器上的传送数据。相同或不同 Oracle 版本服务器之间可以使用链路传送数据,通信接口可以是 SQL \* NET V1 或 SQL \* NET V2。

## 3. 在 PB 内调用 SQL \* PLUS 直接操作远程表

在 Client/Server 结构中,客户端都需安装 SQL \* NET(V1 和 V2)及 SQL \* PLUS(Plus31、Plus32 或 Plus33)产品,以便与服务器接口并管理服务器。在 PB 的 Script 内,通过调用客户端产品 SQL \* PLUS 内的

Copy 命令来完成数据的传输,但这种操作要离开 PB 的 Window 界面,进入到 SQL \* PLUS 的命令行提示符状态,用户界面不好。这种传输方法的具体实例如下:

```

long file_id
string str_sb
file_id = fopen("c:\example\trans1.txt")
if file_id = 1 then
    fileread(file_id, str_sb)
    fclose(file_id)
    str_sb = str_sb + " " + string(dno) /* dno 是 Where 条件的参数 */
    run(str_sb, normal!)
else
    messagebox("系统提示:", "丢失数据文件(c:\example\trans1.txt)!")
end if

```

其中:trans1.txt 文件中存放的是调用 Plus31 的语句,其内容是:

```

c:\orawin\bin\plus31 scott/tiger@local_node:
ora7 c:\example\trans1.sql

```

这条命令大含义是:由 Plus31 直接联接到本地节点上,然后再执行 trans1.sql 文件内的内容。trans1.sql 文件中存放的是 copy 语句,其内容是

```

copy to scott/tiger@remote_node:ora7 -
replace emp using select * from emp -
where empno = &1 ; /* &1 将接收 dno 参数 */

```

注:没有直接把 Copy 语句写到 PB 的 Script 内,而是通过调用二个文件来实现 Copy 功能,这样做是为了动态地传递参数 dno,另外 trans1.txt 和 trans1.sql 是二个文本文件,用户可以不依赖于 PB 的开发环境在记事本或其他编辑器内进行修改。

## 三、间接操作远程数据库表

这种方法不直接存储或更新远程服务器上的数据表,而具体实现过程是:① 由客户向本地服务器发出请求,把需传送的数据以文本文件的格式存储到客户机中,然后再把文本文件传送到远程服务器或客户机上;② 由远程客户把文本文件中的数据接收到服务器的数据表内。根据传送文本文件方式的不同,这种方法可分成网络传输和报送磁盘二种。

### 1. 网络传输

在广域网互通的基础上,通过 FTP 命令或电子邮件把文本文件传送到远程服务器的某一系统帐户或邮件服务器的某一邮箱内,然后远程客户从服务器上把文本文件取到本机内。再用系统的接收数据功能把文本文件中的数据装载到服务器内。生成文本文件和从文本文件

接收数据的方法是利用 PB 的数据窗的卸出和装入功能来完成的。具体的应用实例如下:

生成文本文件的 Script

```
string str ; long ll
```

datastore ds\_1 /\* datastore 是 PB 的一种对象, 等同于数据窗 \*/

```
ds_1 = create datastore
```

```
str = 'c: \ example \ emp. txt'
```

ds\_1.DataObject = "d\_emp\_sbc" /\* d\_emp\_sbc 是物理数据窗 \*/

```
ds_1.SetTransObject(sqlca)
```

```
ll = ds_1.retrieve(dno)
```

if ll > 0 then ds\_1.SaveAs(str, Text!, FALSE) /\* 卸载 ds\_1 中的数据 \*/

从文本文件接收数据的 Script

```
long ii
```

```
ii = ds_1.importfile('c: \ example \ emp. txt')
```

```
if ii=0 then
```

```
messagebox('系统提示:', '接收成功!')
```

```
ds_1.update()
```

```
commit ;
```

```
else
```

```
messagebox('系统提示:', '接收失败, 数据文件有问题!')
```

```
end if
```

## 2. 报送磁盘

当广域网不通时, 把装载数据的文本文件拷贝到软盘内, 然后将软盘送给接收数据的用户, 再从软盘把数据装载到服务器内。其数据窗的卸出和装入方法与网络传输节中方法相同。

## 四、各种传输方法的对比

综上已叙述了各种数据传输方法及实现过程, 如下分析各种传输方法的效果。

### 1. 三种直接传输方法的对比

(1) 传输率方面: 第三种方法速度最快, 因为通过 SQL \* Plus 调用服务器, 数据记录以批量方式传送; 第二种方法次之, 数据记录也以批量方式传送, 但 PB 调用服务器不如 SQL \* Plus 调用服务器效果好; 第一种方法最慢, 通过光标, 数据记录逐条传送。

(2) 界面友好性及操作方便性方面: 第一、二两种方法不离开 PB 环境, 用户界面好; 第三种方法离开 PB 环境, 进入字符界面, 用户界面不好。

(3) 通用性及可移植性方面: 第一种方法通用性最好, 不受任何条件的限制; 第二种方法次之, 受 Oracle 系统版本的限制; 第三种方法最差, 受 SQL \* PLUS 版本及

二个文本文件存放位置的限制。

(4) 安全性方面: 第二种方法最好, 远程服务器的帐户名和口令都在数据库链路内, 对最终用户来说是透明的。第一种方法次之, 帐户名和口令写到 Script 内, 用户虽不知道, 但缺乏灵活性, 当然也可以由用户自己通过键盘输入。第三种方法最差, 因帐户名和口令存到文本文件内, 用户很容易知道。

### 2. 二种间接传输方法的对比

(1) 传输率方面: 第一种方法比第二种方法快, 因为第一种方法通过网络, 而第二种方法通过软盘。

(2) 界面友好性及操作方便性方面: 第一种方法较第二种方法略好, 因第一种方法使用软盘。

(3) 通用性方面: 各有优缺点, 当二地距离较远时, 第一种方法更方便; 而当广域网不通时, 第二种方法更有效。

(4) 安全性方面: 二种方法基本相同, 因本地客户都不知道远程服务器上帐户名和口令。

### 3. 直接传输法和间接传输法的对比

(1) 传输率方面: 直接传输法比间接传输法快得多, 因直接传输法是直接更新远程服务器的数据表。

(2) 界面友好性及操作方便性方面: 直接传输法比间接传输法要方便好得多。

(3) 通用性方面: 直接传输法必须要求远程数据库必须准备(启动)好, 间接传输法则不必; 当网络通时, 直接传输法更好; 当网络不通时, 间接传输法更好。

(4) 安全性方面: 间接传输法比直接传输法安全性要高得多, 因间接传输法不需本地用户知道远程服务器的帐户名和口令。

## 五、结束语

本文所述的各种数据传输方法, 其综合性能各不相同。在信息管理系统中, 需要使用哪一种方法, 要根据应用系统的具体实际情况而定, 选择适合于自己情况的一或二种传输方法。目前, 这些传输方法在辽河油田的“开发”、“计划”及“科技”等几个大型数据库应用系统中都使用过, 并取得良好的效果。

### 参考文献

- [1] ORACEL 7 Server Administrations Guide, Oracle Corp. 1996
- [2] PowerBuilder5.0 Advanced Applications, Powersoft Corp. 1997
- [3] 萨师煊, 王珊. 数据库系统概论(第二版), 1995.3
- [4] 李建中, 王珊, 周龙骧. “数据库技术新进展—VLDB'97 综述”, 计算机世界, 1998.3.2:D1-D2  
(来稿时间: 1999 年 7 月)