

# TCP/IP 网络环境下数据库的远程传输

车文富 (北京系统工程研究所 100010)

**摘要:**本文提出了一种在 TCP/IP 网络协议下自动实现 INFORMIX 数据库库备份本地拆卸、远程传输、异地重构和访问的方法,并举例说明实现控制全过程。

**关键字:**INFORMIX 数据, TCP/IP 网络, UNXI 操作系统, SQL 语言, 数据复制, 信息共享。

## 一、引言

UNIX 是目前大、中、小型各档次计算机最流行、使用最广泛的操作系统,绝大多数版本的 UNIX 系统内核都嵌有 TCP/IP 网络协议。随着信息社会的发展、信息交换的频繁,人们对网络传输信息的需求也发生了显著的变化:文件一级的远程传输和信息共享已满足不了特定应用的需要,必须上升到数据库一级,提供系统、灵活而又内容丰富的共享访问支持。然而,目前 UNIX/PC 平台上的数据库大多数是集中式的,本身不具备数据的分布管理和远程访问的能力。在这种情况下,实现数据库一级的信息共享具有一定的难度,从技术角度讲,需要解决如下两个问题:一是从系统的、数据管理的角度,如何将数据库复制到远地;二是从方便使用的角度,借助 TCP/IP 网络协议的支持,如何实现数据库复制的自动化。

INFORMIX 数据库是 UNIX 平台数据库系统的主流。它具有高的可靠性、兼容性和系统安全性;它采用先进的数据处理技术,提供众多的管理工具,并具有简单灵活、功能强大的新一代语言 INFORMIX - 4GL。对集中式的 INFORMIX 数据库,利用它提供的管理工具并借助 TCP/IP 网络协议的支持,有能力解决上述问题。

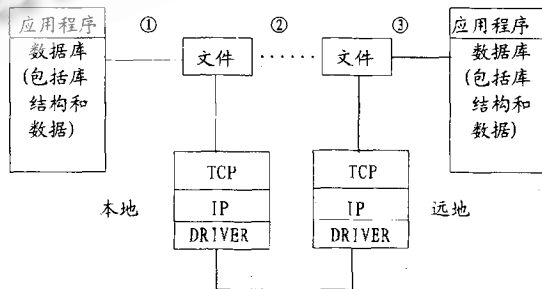
## 二、数据库远程传输的实现策略

TCP/IP 网络协议提供可靠的文件传输服务,而数据库系统又是在文件系统的基础上发展起来的,那么能否利用文件传输直接复制数据呢?实际上数据库发展到今天,它与文件系统在形式、内容、数据存放和管理机制等方面有着本质的不同。数据库中的数据虽然最终以文件的方式交赋操作系统管理,但为了增加系统性和安全性,数据库首先要对数据进行处理和转换,然后将同一表中描述不同对象(如数据和索引等)的信息存放在不同

的文件里。因此在对应关系上,逻辑上独立的库或表并不完全对应操作系统平台上完全确定的文件集合。即使勉强将这些文件直接传至远地,也将难以与远地数据库系统接轨而有效地实施访问。因此从这个意义来讲,数据库的远程传输从总体策略上,不应该依据网络协议直接传输数据库“对应”的底层文件,而应该从数据库管理的观点、系统的观点,分析、研究并充分利用 INFORMIX 数据库提供的管理工具,先将逻辑上独立的数据库分“块”、“拆卸”成若干“部件”并转换成操作系统平台上的确定的文件;然后利用可靠的网络传输协议将文件传至远地;最后在远地同样利用数据库管理工具将接收的各类文件即各“部件”按序回装重构数据库以便远地用户访问。

因此,数据库远程传输可分如下三个阶段,如图 1 所示。

1. 本地数据库的拆卸;
2. 本地卸出文件到远地的可靠传输;
3. 远地数据库的重构。



- ①阶段一: 用 INFORMIX 工具拆卸数据库;
- ②阶段二: 借助 TCP/IP 传输文件;
- ③阶段三: 用 INFORMIX 工具重构数据库;

图 1 INFORMIX 数据库远程传输各阶段示意图

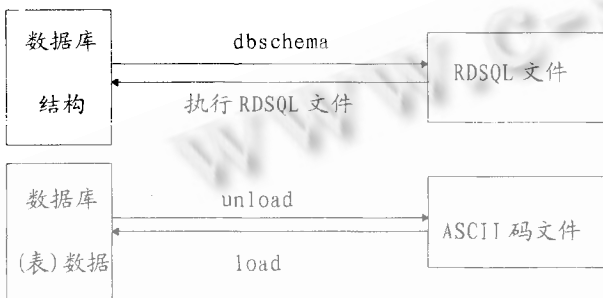
### 三、数据库远程传输的实现过程

上述实现数据库远程传输的三个阶段,对 INFORMIX 数据库来讲,从技术上是完全可行的。INFORMIX 提供的管理工具和实用程序,可用来将数据库“分块”进行“拆卸”和“重构”。

根据 INFORMIX 数据库管理工具的特点和作用范围,从内容上将数据库的远程传输划分为如下几部分:

1. 数据库的结构;
2. 数据库的数据内容;
3. 访问数据库的应用程序。

下面分别阐述各部分内容的卸出、传输和重构过程,如图 2 所示。



说明: → 方向操作在本地; ← 方向操作在远地;

图 2 数据库各“部件”与操作系统文件的转换关系

#### 1. 数据库的库结构

之所以将远程传输的数据库的库结构和数据内容分开,完全是由 INFORMIX 数据库提供的管理工具的工作模式决定的。INFORMIX 提供的实用程序 `dbschema` 可以快速产生一个复制整个数据库和表结构模式的 RDSQL 命令文件。该文件包含建立数据库所需的包括表定义、权限定义、视图定义的所有的 RDSQL 语句,在其他结点执行此命令文件可以重新生成与原数据库结构完全相同的新的数据库(结构)。

#### 2. 数据库的数据内容

前面提到,INFORMIX 数据库的数据是分散存放在不同文件中的。要实现数据库的远程传输且保证传输后数据库能被访问,需要将逻辑上独立的数据库中的表包含的数据从不同的文件中抽取出来(卸出)并转换成 UNIX 平台下的文本文件;传至异地;再将卸出文件中的数据回装到原来对应的数据库表中。INFORMIX 数据库

提供从数据库的表数据到操作系统下 ASCII 码文件相互转换的命令 `unload` 和 `load`。

`unload` 命令用于将表中的数据转换成 ASCII 码文件。

`load` 命令用于将事先用 `unload` 命令卸下的 ASCII 码文件装入数据库还原转换成数据库系统可识别和访问的数据库表中。

至此,选用 `unload` 语句将数据库内容卸出,利用 TCP/IP 传至远地,再使用 `load` 语句将数据库内容重装入库即可实现数据库数据部分的复制。

#### 3. 访问数据库的应用程序

为提高访问数据库的效率,常常事先开发特定的应用软件。如果只复制数据库,而不复制开发的应用软件,可以说只解决了数据库库备份的异地安全存放即数据复制。在数据库不具备分布功能的条件下,异地只能使用 RDSQL 等通用的工具访问数据库或重新开发新的应用软件,这样导致软件的利用率不高。因此在复制数据库的时候,需要同时将开发的应用软件传至远地。

从技术上讲,应用软件的复制相对简单,它不涉及数据库的特性,只需利用 TCP/IP 提供的可靠的网络传输协议直接传输目标文件。

### 四、数据库远程自动传输实例

我们在两台安装有 UNIX SVR 4.0 及相应的 TCP/IP 网络协议的环境下实现集中式 INFORMIX 数据库库备份的卸出、传输(包括应用程序)和重构的自动生成过程。

#### 1. 实例说明

本地结点 XTA;远地结点 XTB。在结点 XTA 上建立一个名为 `sample-db` 的数据库实例,存放着部门、设备以及部门-设备购买关系这三个实体的相关信息。该库包含如下三个表:部门表 `department`;设备表 `device` 和购买关系表 `depart-device`。

此外,开发若干访问 `sample-db` 数据库的应用程序。

#### 2. `sample-db` 库结构的卸出和重构

使用 `dbschema` 实用程序卸出库 `sample-db` 的结构,将结果存放在 RDSQL 命令文件 `sample-db "sample-db.sql"`

该命令执行后, `sample-db.sql` 文件存放着重构整个 `sample-db` 数据库库结构所需的所有的 RDSQL 语句。

在结点 XTB 执行如下操作系统命令,即可实现对

sample-db 数据库库结构的重构:

```
isql -d sample-db -qr sample-db
```

### 3. sample-db 库内容的卸出和装入

使用 unload 命令将表 department、表 device 和表 depart-device 中的数据卸出转换成 ASCII 码并分别存入文件 sample-1.data、sample-2.data 和 sample-3.data。将卸出各表的 unload 命令放入命令文件 sample-unload.sql, 以方便系统调用。该文件内容如下:

```
unload to "sample-1.data"
select * from department;
unload to "sample-2.data"
select * from device;
unload to "sample-3.data"
select * from depart-device;
```

执行命令文件 sample-unload.sql 的系统调用命令如下:

```
isql -d sample-db -qr sample-unload.
```

在结点 XTB 执行命令文件 sample-load.sql 重构数据库库结构后, 执行 RDSQL 命令文件 sample-load.sql 即可重构数据库 sample-db 包含的各表的数据内容。该文件包括如下 load 语句:

```
load from "sample-1.data"
insert into department;
load from "sample-2.data"
insert into device;
load from "sample-3.data"
insert into depart-device;
```

### 4. 数据库库备份远程自动传输控制全过程

要实现数据库的远程自动传输, 必须借助 TCP/IP 提供的可靠数据传输协议编制实际的通讯控制程序, 将分散的命令相互协调、有机地结合在一起, 构成数据库库备份本地卸出、远程传输和异地重构的一体化自动生成过程。

本例采用 C 语言, 利用 TCP/IP 提供的 socket 编程接口编制了一个网络通讯控制程序。该通讯程序的客户方为 client, 在本地结点执行; 服务方为 server, 在远地结点执行。双方的工作过程如下:

客户方 client:

- (1) 初始化客户方网络通讯参数;
- (2) 卸出数据库 sample-db 的库结构。系统调用如

下:

```
system("dbschema -d sample-db sample-db.sql");
```

- (3) 卸出数据库 sample-db 各表中的数据并转换成 ASCII 码文件。系统调用如下:

```
system("isql -s sample-db -qr sample-unload");
```

- (4) 请求与 XTB 结点的 server 进程建立通讯连接;

- (5) 连接建立后, 传送下列文件到达 XTB 结点:

```
sample-db.sql; sample-1.data; sample-2.data; sample-
```

- 3.data 及访问数据库 sa/sample-db 的应用程序;

- (6) 断开与 XTB 的通讯连接;

- (7) 进程结束。

服务方 server:

- (1) 初始化服务方网络通讯参数;

- (2) 等待 XTA 结点的建立连接请求;

- (3) 接受 XTA 结点的建立连接请求;

- (4) 接收 XTA 结点发送的各个文件;

- (5) 接受 XTA 结点的断开连接请求;

- (6) 初始化数据库 sample-db, 如删除旧版本;

- (7) 重构数据库 sample-db 的库结构。系统调用如下:

```
system("isql -s sample-db -qr sample-db");
```

- (8) 重构数据库 sample-db 中各类的数据内容, 将各表中的数据由 ASCII 码转换成数据库系统可直接访问的形式。系统调用命令如下:

```
system("isql -s sample-db -qr sample-load");
```

- (9) 进程结束。

## 五、结论

前述 INFORMIX 数据库远程自动传输及网络通讯控制程序实现的实现具有普遍意义。它表明: 采用本文提出的方法实现 INFORMIX 数据库的数据复制是可行的。该方法将 UNIX 平台上的 INFORMIX 数据库与 TCP/IP 网络协议完美地结合在一起, 可以解决应用中的许多实际问题。

### 参考文献

- [1] 王尽然、林新观等著。《INFORMIX——SQL 关系数据库管理系统技术备查》, 学苑出版社, 1993。
- [2] 孙义、章立生等著。《UNIX 环境下的网络程序设计》, 北京希望公司, 1991 年 9 月。
- [3] 吴建、牛光远等著。《UNIX 系统 V/386 第四版网络程序员指南》, 电子工业出版社, 1992 年 8 月。