

ODBC 及 Oracle SQL * NET 在 C/S 中的配置与应用

孙学东 (无锡市经济信息中心 214023)

一、引言

ODBC(Open DataBase Connectivity 开放数据库互联)允许数据库与应用程序实现数据交换,以及对不同 DBMS 的共享。SQL * NET 是 Oracle 公司的远程数据库访问软件包,在网络环境下实现客户机/服务器的连接与通信。

在目前较为流行和成熟的 C/S 解决方案中,如何使用 ODBC 及 SQL * NET 来访问 Oracle 数据库服务器,是此类开发中常遇到的关键问题之一。本文分别针对 UNIX、NetWare 两种系统环境,对上述问题作一介绍。

二、SQL * NET 的结构及功能

SQL * NET 采用 TNS(透明网络底层)和符合工业标准的网络协议来连接 Client 和 Server,并建立一个 ORACLE 会话。SQL * NET 从 TNS 监听进程处接收与数据库连接的请求,并将控制传递给数据库服务器,建立并保持 Client 应用程序和 Server 数据库之间的连接,完成两者之间的信息转换(转换成一系列对 SQL 例程的调用)。

SQL * NET 有两个版本:SQL * NET V1 及 SQL * NET V2。较早的 SQL * NET V1 有两个部分,与 ORACLE RDBMS 和所有工具连接的 SQL * NET 组成部分、SQL * NET 协议驱动程序。其中协议驱动程序是与特定协议有关的网络例程。每个协议和每个公司在每一个平台上实现协议都需要单独的 SQL * NET 驱动程序。而在新的 SQL * NET V2 中,SQL * NET 与 ORACLE 和工具连接部分仍保留,SQL * NET 驱动程序已经重新设计,它提供通用的网络操作,所有的通用网络操作和指定协议支持被 TNS 和 ORACLE 协议适配器所取代。它包括了所有的 TNS 性能及多协议支持、多协议转换连接。因此对于 SQL * NET 的 V1 及 V2,在 Server 端及 Client 端的配置与使用均有所区别。图 1 说明了 SQL * NET V1 及 V2 在结构上的变化。

三、SQL * NET 的配置

1. SQL * NET V1 的配置与启动

(1) 服务器端

①设置 ORACLE SQL * NET V1 的服务进程 orasrv UNIX 系统下,在/etc 目录下的 services 文件中加入一行:

```
orasrv 1525/tcp (1525 是 ORACLE 推荐的 TCP/IP 端口)
```

NetWare 下,在 F:/ORANW722/NLM 目录的 ORALOAD.NCF 下加入一行:

```
load spxsrv
```

②设置数据库服务器的主机名

UNIX 系统下,在/etc 目录下的 hosts 文件中加入 TCP/IP 地址及主机名。如:

```
192.0.4.1    WXIC01
```

```
192.0.4.2    WXIC02
```

(2)客户端。在 oracle.ini(一般在 windows 目录下)文件中加入对 ORACLE 连接串的描述,格式为:

```
s-name = pro-name:server-name:SID
```

其中 s-name 为连接串名

pro-name 为网络协议前缀(TCP/IP 为 t)

server-name 为数据库服务器名(也可直接写 TCP/IP 地址)

SID 为 ORACLE 数据库系统描述符(在 var/opt/oracle/oratab 文件中有 SID 的描述)

例如:yaod1 = t:wxic01:oracle7 或

```
yaod1 = t:192.0.4/1:oracle7
```

在/windows/oracle.ini 文件中修改 local 参数,如:

```
local = yaod1(默认的连接串名)
```

(3)SQL * NET V1 的启动。当数据库启动后,启动 SQL * NET V1(以 UNIX 为例):

```
$ tcpctl start
```

2. SQL * NET V2 的配置与启动

(1) 服务器端

①设置 ORACLE SQL * NETV2 的服务进程 listener

UNIX 系统下,在/etc 目录下的 services 文件中加入一行:

```
listener 1521/tcp (1521 是 ORACLE 推荐的 TCP/IP 端口)
```

NetWare 下,在 F:/ORANW722/NLM 目录的 ORALOAD.NCF 下加入一行:

```
load tnslsnr
```

②配置 TNS 监听进程的参数文件 listener.ora

listener.ora 一般在 \$ORACLE-HOME/network/admin(或在/varopt/oracle 目录下),包含三个部分:定义监听进程地址、描述数据库信息、控制参数。

如果只有一个网域,COMMUNITY 可不指定,如果只监听一个协议,ADDRESS-LIST 不需要,如果只有一个数据库例程,SID-LIST 不需要。例如:

```
LISTENER = (ADDRESS =
             (COMMUNITY = wxic)
             (PROTOCOL = tcp)
             (HOST = wxic01)
             (POST = 1521)
            )
SID-LIST-LISTENER = (SID-DESC =
                    (SID-NAME = oracle7)
                    (ORACLE-HOME = /oracle/ora722)
                   )
```

```
STARTUP-WAIT-TIME-LISTENER = 0
```

```
CONNECT-TIMEOUT-LISTENER = 10
```

③配置标识服务器参数文件 tnsnames.ora

tnsnames.ora 用于标识连接的数据库目标。每个标识的实体包括两个部分:数据库名及连接描述符(也称为连接串)。具体格式略。

tnsnames.ora 一般位于 \$ORACLE-HOME/network/admin 目录。

④设置数据库服务器的主机名

同 SQL * NET V1 一节的描述。

(2)客户端。tnsnames.ora 的格式及用法参见服务器端的描述。在客户端该文位于/orawin/network/admin 目录。

在/windows/oracle.ini 文件中修改 local 参数:

```
local = oracle1 (默认的连接串名)
```

(3)SQL * NET V2 的启动

①UNIX 系统下,待数据库启动后,启动 SQL * NET V2:

```
$ lsnrctl start
```

②NetWare 系统下,待数据库启动后,启动 SQL * NET V2:

```
load lsnrctl
```

```
lsnrctl>start
```

四、客户端 ODBC 的配置

运行 ODBC Administrator(双击该图标),进入 ODBC 配置画面。

1. 选择 ADD 增加数据源

在 Installed ODBC Drives 列表中选择 Oracle 7 Driver 选项,点击 OK 按钮。如果没有 Oracle7 的驱动程序,应退到上级菜单选 Driver 按钮,先安装相应的驱动程序。

2. ODBC Oracle 设置

在正确选择了驱动程序后,进入 ODBC Oracle Setup 画面:

(1)在 Data Source Name 栏目中定义一个数据源名,如 yaof7

(2)在 Description 栏目中填入对数据源的描述(可省略)。

(3)在 SQL * NET Connect String 栏目中输入连接串名(SQL * NET V1 的连接串名在 oracle.ini 文件中定义,SQL * NET V2 的连接串名在 tnsnames.ora 文件中定义)。

选择 OK 按钮,退到 ODBC 设置主画面,按 CLOSE 按钮关闭主画面,至此完成对 Oracle 的 ODBC 设置。

五、在应用程序中访问 Oracle

一旦配置好 SQL * NET 和 ODBC 后,即可在应用程序中访问 Oracle。各种 C/S 开发工具通过 ODBC 访问 Oracle 的方法有所差别,但基本相同,下面以使用面较广的 VFP3.0、VFP5.0 为例简单加以说明。在 VFP 中有两种方法同 Oracle 建立连接、访问和数据交换。

1. SQL Pass Through 技术

SQL Pass Through(SPT)类似于低层文件操作函数,它能使用户直接访问后端的数据库服务器。SPT 允许用户对建立连接、传送 SQL 命令、错误检查等进行直接控制。所有这些功能是通过调用一组以“SQL”三个字母开头的函数来实现的,常用的有:

SQLCONNECT() 建立同 Oracle 的连接链路,返回连接句柄。

SQLDISCONNECT() 解除同 Oracle 的连接链路。
 SQLSETPROP() 设置连接特性。
 SQLGETPROP() 获得连接特性。
 SQLEXEC() 发送 SQL 语句到数据源, 语句在 Server 端被执行。

如为查询, 则返回结果集到本地游标(CURSOR)。
 SQLCOMMIT() 递交一个事务。
 SQLROLIBACK() 取消当前事务的所有改变。
 例如:

```
H1 = SqlConnect("yaof7")
IF H1 < 0
    = MessageBox("连接远程数据源错误!", 62, "出错信息窗口")
    RETURN
ELSE
    ST = SqlExec(H1, "select * from Invoice")
    IF ST < 0
        = MessageBox("获取远程数据错误!", 62, "出错信息窗口")
        = SqlDisconnect(H1)
    RETURN
ENDIF
ENDIF
```

当需要对事件进行精确控制或需进行复杂查询时, 使用 SPT 是很适宜的, 因为它能最大限度地利用 Oracle 的特性, 如强大的 Oracle 函数与句法。但需将具有大量字段的本地表数据更新到远程数据源时, 使用 SPT 并不很方便。

2. 远程视图技术

在通常情况下, 可通过建立远程视图来实现对远程数据源的查询与更新, 使用较为方便与灵活, 尤其是需将具有大量字段的本地表数据更新到远程数据源の場合。

3. 调用 Oracle 存储过程

使用 Oracle 存储过程是一种高效方便的开发方法, VFP 不支持对 Oracle 存储过程的直接调用, 但可通过间接的方法实现对 Oracle 存储过程的调用。

首先可以在 Oracle 中定义一张表, 该表只有一条记录, 每个字段代表一个存储过程。对每个字段定义一个列更新触发器, 在触发器中调用对应的存储过程。然后当客户端需要调用存储过程时, 在应用程序中对该表的相应字段实施更新操作, 引发对应的触发器被执行, 触发器再调用存储过程, 从而间接实现了对 Oracle 存储过程的调用。如有参数传递, 可另外定义一张参数表来完成。

因篇幅关系, 不再举例。

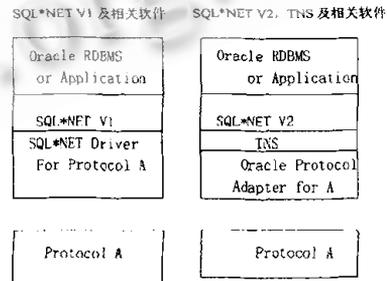


图 1 SQL * NET V1 和 SQL * NET V2 的比较

(来稿时间: 1998 年 8 月)