第14章 UNIX上的Oracle联网技术

对于许多Oracle安装来说，UNIX是首选的操作系统，它既可支持小型的 Oracle数据库，也可支持特别大型的数据库。操作系统的灵活性、调整选项和可伸缩性的组合使得 UNIX成为Oracle的坚实基础。在本章中，你将看到许多用于在 UNIX中实现SQL*Net V2和Net8的指令。可以使用 SQL*Net V2和Net8来访问Oracle8数据库（也同样可用于 Oracle7数据库）。对于Oracle8i, Net8将使用7.3.4以后的版本。由于 TCP/IP通信协议通常用于 UNIX服务器，所以该协议将在本章的例子中使用。

在一个进程能够连接到服务器中的数据库之前，有几个使数据库管理员必须与 UNIX系统管理员联系的步骤。下面的章节详细描述每一个步骤。

14.1 主机的识别

主机(host)在本章中被定义为可通过网络与另一个服务器进行通信的服务器。每台主机都有一个它能与之通信的主机列表，这个列表保存在称为 /etc/hosts的文件中，文件名的“ /etc ”部分表示它位于/etc目录下。这个文件包含主机的 Internet地址和主机名字。它还可以包含每个主机名的别名。下面列出 /etc/hosts文件样例的一部分：

```
127.0.0.1 localhost
130.110.238.109 nmhost
130.110.238.101 txhost
130.110.238.102 azhost  arizona
```

你的UNIX工具可能使用域名服务器（ DNS），在这种情况下，主机 IP地址可能没全部列在/etc/hosts文件中。如果采用了 DNS，就可使用nslookup命令来查询已知主机名的 IP地址（反之亦然）。下面列出从DNS中查找IP地址的样例：

```
> nslookup txhost
Server:  txhost.company.com
Address: 130.110.238.101
```

在这个例子中列出了 4 个主机。第一项是服务器的“ 回送 ”条目，接着的两个条目把主机名（nmhost和txhost）赋予Internet地址，最后一个把主机名（azhost）和别名（arizona）赋予Internet地址。

PC客户机的大多数联网软件都使用相似的文件，这些文件或者位于 PC机，或者位于共享的网络驱动器。在网络软件目录结构中，有主机文件，该文件列出了客户机能直接到达的每个主机的IP地址和主机名。该文件在结构上与前面所示的 UNIX的/etc/hosts文件相同。

尽可能使用主机名而不是 SQL*Net/Net8配置文件中的 IP地址。 UNIX服务器可以使用 DHCP，它是在每个服务器启动过程中把 IP地址分配给主机的一种协议。使用 DHCP的服务器在每次启动时可能具有不同的 IP地址，因此，任何基于硬编码 IP地址的连接都将失败。

14.2 数据库的识别

所有在主机上运行并能通过网络访问的数据库都必须列在通常称为 /etc/oratab的文件中。

该文件名的“/etc”部分指明它位于/etc目录下。该文件由DBA维护。

注意 根据所使用的UNIX版本，/etc/oratab文件的名字和位置可能有所不同。详细情况请参见操作系统特定的Oracle Installtion Guide。

该文件中各条的成份列于表14-1。

表14-1 /etc/oratab文件的条目成份

成 份	描 述
Oracle_SID	实例名（服务器ID）
Oracle_HOME	数据库使用的Oracle软件的根目录的全路径名
STARTUP_FLAG	表明主机启动时是否应启动实例的标志，若设置为Y，则启动实例；若设置为N，则不启动实例。该标志由Oracle提供的db_startup命令文件使用

Startup_Flag部分看起来似乎不合适，那么网络连接到底为什么要重视实例的启动安排呢？由于在系统启动时也要使用该文件（用缺省值，可以改变）来启动服务器的Oracle数据库，所以这个标志也是条目的一个部分。这三个部分都被列在一行上，用冒号（:）隔开。下面所示的是/etc/oratab文件的一个样例：

```
loc:/orasw/app/oracle/product/8.1.5.1:Y
cc1:/orasw/app/oracle/product/8.1.5.1:N
old:/orasw/app/oracle/product/8.1.5.0:Y
```

该样例示出了3个实例条目，分别命名为 loc、cc1和old。前两个实例具有相同的ORACLE_HOME，第三个实例使用Oracle内核程序的旧版本。当服务器启动时，loc和old将自动启动，而cc1则必须手工启动。若要访问这些实例，首先必须启动监听程序进程。下一节将描述UNIX中的监听程序进程的配置。

14.3 服务的识别

服务器进程（server process）监听来自客户机的连接请求。服务器进程把这些请求引导到合适的UNIX套接字并进行连接。管理Listener服务器进程——SQL*Net V2和Net8的服务器进程——的方式在SQL*Net V2和Net8之间有稍微变动。

监听程序进程所需要的信息（如端口说明和主机名）存储在分布于网络中的文件上。每个主机上的tnsnames.ora文件都将包括服务名（service name）的列表以及相关的连接描述符。这些描述符包含了与UNIX监听程序进程建立连接所必需的信息。下面所示的是tnsnames.ora文件的一个样例条目：

```
HQ =(DESCRIPTION=
      (ADDRESS=
            (PROTOCOL=TCP)
            (HOST=HQ)
            (PORT=1521))
      (CONNECT DATA=
            (SID=loc)))
```

在这个例子中，服务名HQ被给予特定的连接描述符。该描述符指定了主机（HQ）、协议（TCP）、端口（1521）和实例ID(loc)。

为了使客户机连接到远程服务器上的数据库，远程服务器必须正在运行监听程序进程。该进程叫做TNSLSNR(TNS监听程序——TNS是transparent network substrate的缩写)，它等待

到listener.ora文件中所列数据库的连接企图。listener.ora文件列出了服务器中的所有监听程序。因此，listener.ora文件在服务器“监听”外部连接请求方面扮演着重要角色。下列所示是listener.ora文件的一个样例部分：

```
LISTENER =
  (ADDRESS_LIST =
        (ADDRESS=
          (PROTOCOL=IPC)
          (KEY= loc)
        )
   )
SID_LIST_LISTENER =
  (SID_LIST =
    (SID_DESC =
      (SID_NAME = loc)
      (ORACLE_HOME = /orasw/app/oracle/product/8.1.5.1)
    )
  )
```

这个listener.ora段示出了监听程序将要服务的实例（这里是 loc 实例）。如上所示，每个实例的Oracle软件主目录都必须列在这个文件中。在 Oracle8中可以使用 Net8 Configuration Assistant（Net8配置助手）来管理配置文件（参见第 13章）。

另一方面，在Oracle8i中，可以在tnsnames.ora文件中定义网络服务名以便允许连接到数据库（参见第13章）。网络服务命名方法也可用于任何 Oracle 8 连接，但由于Oracle7监听程序要求一个SID值并可能不识别服务名选项，因而对于 Oracle7监听程序可能会引起某些混乱。例如，tnsnames.ora文件中的版本8i条目如下：

```
LOC=
  (DESCRIPTION=
   (ADDRESS =
        (PROTOCOL = TCP)
        (HOST = HQ)
        (PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = loc)
      (INSTANCE_NAME = loc)
    )
)
```

当连接数据被传送到版本 7.3.4的监听程序时， init.ora参数SERVICE_NAME和INSTANCE_NAME 可能不被理解。如果打算与版本 7 的监听程序相互配合，则必须使用如下旧样式的连接数据：

```
(CONNECT_DATA =
     (SID = loc)
  )
```

当试图使用tnsnames.ora文件或listener.ora文件时，Oracle将搜索由TNS_ADMIN环境变量标识的目录。在大多数情况下，/etc目录被用作这两个文件的TNS_ADMIN目录。

启动监听程序进程的情况在下一节中描述。当从 SQL*Net V2移植到Net8时，监听程序控制选项的参数会有某些变化。

14.4 启动监听程序服务器进程

监听程序进程由Listener Control Utility(监听程序控制实用程序)控制，通过lsnrctl命令来执行。可用于lsnrctl命令的选项在14.5节“控制监听程序服务器进程”中介绍。使用下列命令来启动监听程序：

```
> lsnrctl start
```

该命令将启动缺省的监听程序（命名为 LISTENER）。如果想启动另一个监听程序，则可以在lsnrctl命令中包括该监听程序的名称作为第二个参数。例如，如果创建了一个叫做MY_LSNR的监听程序，则可通过下列命令来启动它：

```
> lsnrctl start my_lsnr
```

在下一节中将对可用于Listener Control Utility的其他参数进行描述。

启动监听程序之后，通过使用 Listener Control Utility的status选项可以检查它是否正在运行。下列命令可用来进行这种检查：

```
> lsnrctl status
```

该命令的输出样例如下所示：

```
LSNRCTL for Solaris: Version 8.1.5.0.0 - Production on 25-JUN-99 16:53:20
(c) Copyright 1998 Oracle Corporation.  All rights reserved.
Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=IPC)(KEY=EXTPROC0)))

STATUS of the LISTENER
------------------------
Alias                     LISTENER
Version                   TNSLSNR for Solaris: Version 8.1.5.0.0 - Production
Start Date                25-JUN-99 13:36:53
Uptime                    0 days 3 hr. 16 min. 27 sec
Trace Level               off
Security                  OFF
SNMP                      OFF
Listener Parameter File   /oracle/products/815/network/admin/listener.ora
Listener Log File         /oracle/products/815/network/log/listener.log
Services Summary...
  loc          has 1 service handler(s)
```

上面列出的状态输出表明监听程序已经启动，并且表明当前只支持一个服务（loc），正如其listener.ora文件所定义的那样。Listener参数文件被标识为/etc/listener.ora，其日志文件位置也被显示出来。

如果希望看到所包含的操作系统级的进程，可使用下列命令。该样例使用 UNIX的ps -ef命令来列出系统的活动进程，grep tnslsnr命令删除那些不包含“tnslsnr”的行。

```
> ps -ef | grep tnslsnr
```

该命令的输出样例如下：

```
oracle  4022     1  0 13:36:53 ?        0:00 /oracle/products/815/bin/tnslsnr
                                        LISTENER -inherit
oracle  5469  2419  1 13:56:23 ttypc    0:00 grep tnslsnr
```

该输出显示两个进程：监听程序进程和检查该进程的进程。该输出的第一行换行到第二行，并且可由操作系统截断。

14.5 控制监听程序服务器进程

必须定期修改监听程序服务器进程。由于不想仅仅为了改变监听程序的参数而不得不关

闭并重新启动服务器，那么可使用lsnrctl实用程序来管理它们。

可以使用Listener Control Utility来启动、停止和修改服务器上的监听程序进程，其命令选项列于表14-2。每个命令都可能有一个值，除set password命令外，这个值将是监听程序名。如果没指定监听程序名，就将使用缺省值（ LISTENER）。一旦进入lsnrctl，就可通过set current_listener命令来改变正被修改的监听程序。

表14-2 Listener Control Utility命令

命 令	描 述
CHANGE_PASSWORD	给监听程序设置新口令，系统会提示输入监听程序的旧口令
DBSNMP_START	启动服务器上数据库的DBSNMP子代理
DBSNMP_STATUS	提供DBSNMP子代理的状态信息
DBSNMP__STOP	停止服务器上的DBSNMP子代理
EXIT	退出lsnrctl
HELP	显示lsnrctl命令选项的列表，也可以通过 help set和help show命令查看附加选项
QUIT	退出lsnrctl
RELOAD	允许在启动监听程序之后修改该监听程序。它强制 SQL * Met读取并使用最新的listener.ora文件
SAVE_CONFIG	Net8中的新命令。创建现有的 listener.ora文件的备份,然后用已由lisnrctl更改的参数来更新listener.ora文件
SERVICES	显示可用服务及其连接历史。它也列出是否为远程 DBA或自动注册访问而启动每个服务
SET	设置参数值。这些选项是： connet_timeout:以秒为单位，监听程序启动之后等待合法连接请求的时间 current_listener:改变其参数正被设置或显示的监听程序进程 log_directory:监听程序日志文件的目录 log_file:监听程序日志文件的名称 log_status:日志记录是ON还是OFF password:监听程序口令 save_config_on_stop:在Net8中新引入。当退出lsnrctl时把配置变化保存到listener.ora文件 startup_waittime:监听程序在响应lsnrctl start命令之前的休眠秒数 trc_directory：监听程序跟踪文件的目录 trc_file:监听程序跟踪文件的名称 trc_level:跟踪级(ADMIN、USER、SUPPORT或OFF)。参见lsnrctl trace
SHOW	显示当前参数设置。这些选项与除 password命令外的set选项一样
SPAWN	产生一个以listener.ora文件中的别名运行的程序
START	启动监听程序
STATUS	提供有关监听程序的状态信息，包括它的启动时间、参数文件名、日志文件和它支持的服务。该命令可用来查询远程服务器上的监听程序的状态
STOP	停止监听程序
TRACE	把监听程序的跟踪级设置为下列４种选择之一： OFF USER（有限跟踪） ADMIN（高级跟踪） SUPPORT(Oracle支持)
VERSION	显示监听程序、TNS和协议适配器的版本信息

注意 lsnrctl的新选项会随着Net8的每一个新版本而不断引入。

可以通过lsnrctl命令进入它，并且进入lsnrctl实用程序的外壳，所有其他命令都可以从这里运行。表14-2中列出的命令选项给出了大量对监听程序进程的控制，如下列例子所示。在大多数例子中，首先进入lsnrctl命令，这样就把用户放到lsnrctl实用程序中（由LSNRCTL提示符表示）。其他的命令从该实用程序内输入。下面的例子示出了使用 lsnrctl实用程序来停止、启动监听程序和生成有关该监听程序的诊断信息。

停止监听程序：

```
> lsnrctl
LSNRCTL> set password lsnr_password
LSNRCTL> stop
```

列出监听程序的状态信息：

```
> lsnrctl status
```

要列出另一主机中的监听程序的状态，可把该主机的服务名作为参数添加到 status命令。下面的例子使用本章前面所示的HQ服务名：

```
> lsnrctl status hq
```

列出监听程序的版本：

```
> lsnrctl version
```

列出关于监听程序所支持的服务的信息：

```
> lsnrctl
LSNRCTL> set password lsnr_password
LSNRCTL> services
```

重新装载listener.ora文件中列出的服务：

```
> lsnrctl
LSNRCTL> set password lsnr_password
LSNRCTL> reload
```

把修改的配置参数保存到listener.ora文件（Net8中有效）：

```
> lsnrctl
LSNRCTL> set password lsnr_password
LSNRCTL> save_config
```

更新所执行的跟踪级：

```
> lsnrctl
LSNRCTL> set password lsnr_password
LSNRCTL> trace user
```

启动监听程序进程：

```
> lsnrctl
LSNRCTL> set password lsnr_password
LSNRCTL> start
```

大多数命令都要求口令，因此，不适合于通过批命令来运行它们，因为那样做的话，要么就把口令存储在文件中，要么就把口令作为一个参数传送给批处理程序。

14.6 调试连接问题

正如本章所描述，UNIX中的SQL*Net/Net8连接要求正确配置许多通信机制。这些连接包括主机到主机的通信、服务和数据库的正确识别、监听程序进程的正确配置等。当使用SQL*Net V2或Net8时，如果出现连接问题，那么尽可能多地消除这些成份是很重要的。

确保连接正试图到达的主机可通过网络访问。这可以通过下列命令来检查：

```
> telnet host_name
```

如果该命令成功了，就提示你输入远程主机的用户名和口令。如果 ping命令是可用的，就可以使用该命令。下面所示的这个命令将检查远程主机是否可供使用，并将返回一个状态消息：

```
> ping host_name
```

如果主机在网络中可供使用，那么下一步就检查监听程序是否在运行，与此同时，可以查看它当前正在使用的是什么参数。如果正试图进行远程自动注册访问，那么这是很重要的。lsnrctl status命令将提供这种信息：

```
> lsnrctl status net_service_name
```

net_service_name将引用远程服务器中的服务名。如果不使用 net_service_name，那么该命令将返回本地服务器中的监听程序的状态。

这两项检查 —— 主机可用性和监听程序可用性 —— 将在服务器／服务器通信中解决 95%以上的SQL*Net和Net8连接问题。其他的问题是由正在使用的数据库说明障碍引起的。这些问题包括无效的用户名／口令组合、关闭数据库和需要恢复数据库。

在客户机／服务器通信中，应用调试连接问题的相同原则。首先验证远程主机是否可访问，大多数客户机的通信软件都包括 telnet或ping命令。如果远程主机不能访问，那么问题可能出在客户机端。验证其他客户机是否能访问保存有数据库的主机。如果它们能访问，那么问题就出在这个客户机上；如果它们也不能访问，问题就出在服务器端，就应检查服务器、服务器的监听程序进程和数据库。

可以使用tnsping命令来测试从客户机到监听程序的连接，用两个参数来执行 tnsping命令：要检查的服务名和企图连接的次数。例如，如果运行命令 tnsping hq 20，那么Oracle就连续20次试图连接到hq服务。因为tnsping命令的输出显示可能很快，所以要使用多次连接，强制它多次测试可给操作者更多的时间去读取输出。 tnsping命令是作为Windows客户机SQL*Net和Net8的连通性软件的一部分提供的。 Net8 Assistant包含一个连接测试部件，它执行与 tnsping相同的功能。