

第13章 Net 8 网络：新特性和概念

本章要点：

概述

连接管理器

其他新特性和概念

配置一些新特性

配置文件示例

Oracle 8重新命名SQL*Net 2.x为Net 8.x。伴随这一名字改变的是一组重要功能的加强，如果正确地配置它，数据库能产生更好的可伸缩性和性能。这些加强包括多路复用、连接池、集中、命名服务的改变、连接改变、安全改变和整个性能。管理也得到提高。

13.1 概述

Net 8是服务器反向兼容到SQL*Net 2.1，但不是客户向前兼容。也就是说，Net 8监听器能服务于Net 8和SQL*Net 2.x客户，但SQL*Net 2.x监听器仅能服务于同版本或低版本客户（不包括更高版本，因此不包括Net 8客户）。

可伸缩性通过多路复用、连接池、集中和增强命名服务的可用性得到增强。连接现在支持外来数据源和第二层的开放数据库连接（Open Database Connectivity，ODBC），ODBC支持存储过程调用。安全性是通过高级安全选项（Advanced Security Option，ASO）来增强的，ASO允许基于公共密钥技术X.509数字认证的单方签名，ASO还包括在客户和服务器之间基本的点到点（“通过线路”）加密。在客户和服务器之间用数字证书相互认证，提供单方签名证实，它通常被认为在多数情况下要优于其他外部认证技术。性能的增强是通过TNS Raw接口的使用、客户端名字缓存（类似于UNIX的DNS方式）和优化调度编码路径获得的。

可管理性的提高是通过提供本地原始设备制造商（OEM）和OEM applet集成支持得到的。对分布式数据库可以和本地数据库一样进行集中化配置。Oracle命名（Oracle Names）提供了增强的动态发现。Oracle 8的Net 8客户可以有配置简档，有一个Net 8向导可以用来配置。最后，可以为小规模的系统自动地提供一个缺省配置。

13.2 连接管理器

连接管理器（Connection Manager，CMAN）是一个新的网络特性，它仅能工作在多线程服务器（MTS）选项上。它集中来自多个客户请求，并将它们多路复用到一个物理通讯。除了集中以外，连接管理器能完成几种功能，包括担任防火墙，能够授权传输或在不同协议之间担任转换的网桥。

使用防火墙能力，CMAN能够通过基于的源（主机或客户）、目的主机和数据库（SID）的连接管理器创建客户端授权访问控制列表（ACL）和过滤（允许和拒绝）传输。

使用协议桥能力，CMAN有效地替代以前的多协议交换（MPL），允许主机（客户和服务

器)使用不同网络协议互相通讯。

多路复用包括合并多个逻辑会话到一个物理链接,在本质上,这个物理链接在多个会话中共享,它们当中没有一个会话在时间上会 100%地使用这个物理链接。第一个会话建立物理链接,然后该物理链接被以后的会话分时共享。当所有的会话被关闭时,物理链接就被关闭。

为什么多路复用是有益的?当使用 MTS选项时,多路复用允许比没有使用它时有更多的并发会话。并发会话的数量被操作系统限制,如对文件描述符、信号量、进程和套接字的操作系统最高限度。没有多线程服务器 MTS选择,一般只能有几百个(200~300)并发会话。有了它,几千个并发会话是可能的,也就是增加了单位时间总产量(也就是吞吐量)。正如所提到的,负载的可伸缩性是可能实现的,因为所有会话不总是 100%地忙。换句话说,它们在一些点处于空闲状态。它们在时间上的空闲重叠为并发提供了机会。因此,用长时间运行的 SQL 语句、复杂的 SQL 语句或者那些有高比例计算时间的 SQL 语句的应用将从 MTS和Net 8中多路复用获得最多的利益。因此,MTS选项和Net 8多路复用打破在高并发环境(操作系统和网络)中潜在的主要瓶颈。CMAN负责管理多路复用,用MTS选项提供它工作的基础。

与多路复用一样,连接池是用于有高比例的计算时间的应用有高的并发性的想法。计算时间可能包括查找、分类、格式化或其他 CPU密集型的活动,因此它也被称为“思考时间”。当有意地使用一个有限空闲时隙数少于到来请求数的交换机时,连接池工作得最好。一些端口被建立。基于UNIX网络,一个端口是硬件地址(例如,一主机 IP地址)和软件地址(例如一个UNIX套接字),连接池所做的就是去除空闲会话的开销。

连接和会话这两个术语在本文通常意思相同,尽管有些微妙的区别。不管这么多,连接是在机器间建立的网络连接(客户和服务),而会话是建立程序链接。

在任何事件中,连接池暂时断开空闲会话并将它们的连接传递到想要建立连接的新会话或想重新连接的以前建立的空闲会话。连接池确保尽可能多的非空闲会话并行活动。甚至会话数超过DBA指定的上限时,它仍然允许建立新会话。

13.3 其他新特性和概念

Net 8既支持两层(客户/服务器)又支持N层结构的分布式系统。在任何结构中,多个监听器能够被均衡装载。换句话说,多个会话能通过多个监听者被几乎相等地分布访问同一个或多个数据库。

当然,两层客户/服务器结构被旧版本的 SQL/*Net所支持。它由在一个固定的基于连接的协议(如TCP)上的同步远程过程调用(RPC)特性化。客户块进入挂起或等待状态,直到结果从服务器返回,如果服务器没有反应(例如被关闭),RPC失败。因为在传统的客户/服务器布局中,连接是固定的或专用的,如果RPC执行大量的计算,则网络利用率被浪费掉了。

用N层模型,异步消息队列能利用无连接协议,如UDP。客户端将它们的消息放置在队列中,然后既可用中止又可继续其他处理;它们不必等待,当服务器可用时(因此它从不失败),信息便被处理。当实时性需要很关键时,消息机制不是很好的方法,但是当高的吞吐量是重点并面临高的并发性时,这个模型是较好的。

TNS Raw是当可能时通过低层协议不使用Net 8头进行最基本通信的能力。基本的客户/服务器连接透明使用TNS Raw调用。换句话说,使用它的用户或DBA不管是客户端还是服务器端都没有配置设置的要求。当可能时 TNS Raw自动地工作,例如当没有安全服务(也就是

ASO)要求时。使用TNS Raw, 有更大的包有效性(或密度), 即每一个包能传输更多的数据, 因为通常传输Net 8头的控制信息空间现在用于搬运数据。更进一步, 在同一会话中发送更少的总数据量。当TNS Raw被透明使用时, 两端Net 8上层的头处理层被绕过, 导致总的网络处理时间更少且总的响应时间更短。

13.3.1 新的安全能力

用高级安全选项(ASO), Net 8传输能够用你选择的加密算法(RSA或DNS)和密钥尺寸(位数)加密。CMAN支持ASO加密, 因此当协议桥存在时, 加密流不打破。ASO也提供通过基于标准协议X.509数字证书的单个符号安全。数字证书是基于公开密钥基础设施(PKI)技术, 每个证书使用实体的私有密钥“签名”, 因此称之为数字签名。用PKI技术, 执行可以是单向(仅服务器)或双向(服务器和客户)的, 后者也称之为强认证。ASO也能提供外部认证适配器插入到Net 8中, 作为对数字认证的替代或补充, 这些附加的外部验证方法包括基于票卷技术(Kerberos, Sesame, DCE)、基于令牌(SecurID)和生物测定学方法(指纹、视网膜等)。

对于ASO有两个真正的部分。一部分ASO提供传输(Net8)加密替代它的前任——高级网络选项(ANO)和安全网络服务(SNS)。这部分ASO符合加密认证方法, 如数字证书, 替代了捆绑在Oracle 8的早期版本中的安全服务器。ASO与Oracle 8i同时发行, 尽管它如同Oracle 8i一样, 在写此书时仍然不完全可用。

13.3.2 新服务、工具和API

Oracle命名是一个对Net 8服务提供名字到地址解决方案的的目录服务。使用Net 8, Oracle命名得到加强, 或新特性都或多或少基于UNIX域名服务器(DNS)技术。

动态解决方案——Net 8服务可以向名字服务器自我注册。

客户发现——客户能在安装时查询名字服务器列表。

客户缓存——客户端名字的缓存能够被维护, 而不是为每次请求解决名字。

安装程序定位或请求名字服务器, 联系到名字服务器, 所有知道的名字服务器的列表被下载。客户询问在那个列表中的每一个服务器, 按相应时间排序结果(如果需要, 以后DBA可以重新分类)。该排序响应列表成为最初的客户端缓存(或本地名字缓存)存储在SDNS.ORA文件中。其他外部命名服务(甚至非数据库)如NIS、NDS和CDS能够被集成。例如本地主机名适配器能使用一个OS命名服务, 如Sun Microsystem的网络信息服务(NIS, 从前在UNIX中被称之为“黄页”)替代Oracle名字服务器。如果协议是TCP/IP, 每一个数据库有一个监听器, 不要求安全服务(也就是ASO)。

管理已经通过OEM applet被简化。OEM applet能被OEM使用或它们自己使用。尽管网络管理软件是过时的, Net 8助手取代它用于服务器端(监听器)配置。然而, Net 8在TCP/IP环境中自动配置自己。如果接受所有的缺省值, 监听器不要求listener.ora文件。用名字服务器, 客户资源环境文件被创建, 并使用客户环境资源编辑器编辑, 这个编辑器可以在多个用户(也就是客户)间共享。Net 8客户自动从包含环境资源的名字服务器装载客户环境资源文件。

Net 8向导自动初始化客户端配置。在安装时它被调用, 处理客户端初始化创建名字服务器缓存, 提供现有的或缺省的客户资源文件(从名字服务器)的选择, 并使能这些配置文件的编

辑。同时SQL*Net Easy Config被简单地重新命名为Net 8 Easy config，作为客户端配置工具。

Net 8能够通过Oracle Trace使用追踪，这有几个好处。Oracle Trace是Net 8的外部程序，因此比从前的SQL*Net追踪代码，有较小的可执行“足迹”。它能从点到点追踪，实际上从客户到服务器。追踪助手能从追踪文件中抽取或显示错误、提取包类型和客户发送的或服务器接受的信息以及维护包的统计。

最后，Net 8提供开放应用程序接口（API），称为Net 8 OPEN API，它允许可移植性。当从一个协议移动到另一个协议时，建立在Net 8网络上的应用能够很容易地移植，因为Net 8方法是协议独立的。适当的低层调用嵌入在Net 8自身内部，可以用适当的协议适配器库重新链接代替。但是高层Net 8调用嵌入在应用中，不能被改变。

13.4 配置一些新特性

在服务器端，和有一个lsnrctl监听控制程序和一个相关的listener.ora结构文件一样，有一个相似的cmctl连接管理器控制程序和一个相似的cman.ora结构文件。可是，在客户端相同的配置文件tnsnames.ora仍然只有一个将被修改。你可以在cmctl中完成许多相同的用于lsnrctl的命令，例如start、stop、status和version。除了cmctl之外，还有两个cmctl控制的进程：连接管理器网关（cmgw）和连接管理器系统管理器（cmadm）。cmgw与实际的监听进程相似。它是通信中心，主要负责所有连接集中职责以及相关的新特性职责。cmadm与Oracle名字服务器协作，主要负责地址信息和注册。当被用作start和status变元时，cmctl内在引用cmgw作为cm，cmada作为adm。cm是唯一的stop变元，因为当cmgw终止时cmada也自动终止。

通过编辑每一客户的tnsnames.ora文件和编辑init.ora文件使能集中（多路复用和连接共享池）和桥，通过编辑cman.ora文件建立防火墙，通过编辑init.ora文件使能监听器负载平衡，当然也可以通过编辑init.ora文件使能MTS选项。

13.4.1 多路复用和连接共享池

多路复用被建立在每个客户和服务上。假设cman.ora文件内容如下：

```
cman=(address=(protocol=tcp)(host=cman)(port=1610))
```

在这种情况下，每个客户的tnsnames.ora被编辑成如下这样：

```
mysid=(description=
  (address_list=
    (address=(protocol=tcp)(host=myserver)(port=1521))
    (address=(protocol=tcp)(host=cman)(port=1610))
    (connect_data=(sid=mysid))
    (source_route=yes)
  ))
```

在服务器上，你的init.ora中MTS节的下一行将被设置成如下这样：

```
MTS_DISPATCHERS="(PROTOCOL=TCP)(POOL=YES)(MULT=YES)"
```

当使能多路复用或连接共享池时，注意这个设置不同于SQL*Net 2.x。当不使用这些Net 8特性时，设置保持不变。注意应为整个协议启用这些集中特性并且不指定任何初始启动调度进程数。

当多路复用在每个客户端被启用和MTS选项被启用（在init.ora文件中）时，那么相关特性的连接共享池也可获得。连接共享池是更特定类型的多路复用。如果MTS没被启用，这两个特性都不能存在，它们都是连接集中的类型。

13.4.2 网桥

多协议支持也主要建立在每个客户端，每个客户端的 tnsnames.ora 将被编辑成如下这样：

```
mysid=(description=
  (address_list=
    (address=(protocol=spx)(service=cman))
    (address=(protocol=tcp)(host=myserver)(port=1521))
    (connect_data=(sid=mysid))
    (source_route=yes)
  ))
```

13.4.3 防火墙

过滤支持建立在服务器上，cman.ora 文件可能被编辑成这样：

```
cman_rules=(rules_list=
  (rule=
    (src=good_host2)(dst=myhost)(srv=mysid)(act=accept))
  (rule=
    (src=bad_host1)(dst=myhost)(srv=mysid)(act=reject))
)
```

实际的有或没有通配符的 IP 地址可以用于 SRC 和 DST 声明，例如 (SRC=234.136.*.*)。

13.4.4 监听器负载均衡

要启用监听器负载均衡，在你的 init.ora 文件 MTS 部分，进行如下设置：

```
MTS_MULTIPLE_LISTENERS=TRUE
```

13.4.5 CMAN 环境资源文件

为了设置特定的 CMAN 运行参数，编辑 cman.ora 文件，包括下列 cman_profile 段的一部分或全部：

```
cman_profile=
  (parameter_list=          /* DEFAULTS      */
    (maximum_relays=512)    /* 8          */
    (log_level=1)           /* 0          */
    (tracing=yes)           /* NO         */
    (relay_statistics=yes)  /* NO         */
    (show_tns_info=yes)     /* NO         */
    (use_async_call=yes)    /* YES        */
    (authentication_level=1) /* 0          */
  )
```

表13-1 简要概括对每个参数的描述。

表13-1 CMAN 运行参数

参 数	描 述
maximum_relays	并发连接的最大数（最大值是 1024）；与 init.ora 参数 SESSIONS 对照
log_level	0 是无日志，1~4 是递增的日志水平
tracing	是或不是；如果是，Oracle Trace 必须打开
relay_statistics	是或不是；如果是，在其他事件中记录 IN/OUT 字节和包的数目
show_tns_info	是或不是；如果是，并且 log_level 大于 0，TNS 事件被记录在日志文件里
use_async_call	是或不是；如果不是，调用将是同步的
authentication_level	0 或 1；如果是 1，拒绝不使用 ASO 安全服务的客户

13.5 配置文件示例

正如以前提到的，启用高级集中特性的多路复用和连接共享池，首先需要启用 MTS 选项，意思是你必须重新配置 init.ora 文件。MTS 选项和地址与 tnsnames.ora 和 listener.ora 文件的配置紧密结合。另外，CMAN 的配置与 listener.ora 文件和所有客户的 tnsnames.ora 文件配置紧密结合（这些例子没有使用 Oracle 名字服务）。注意黑体部分，这是这些文件中与我们的讨论最相关的部分。MTS、监听器负载均衡、多路复用、连接共享池、网桥、防火墙和 CMAN 环境资源文件被启用。例子如下：

```
INIT.ORA
/* To enable MTS, multiplexing, connection pooling and listener load balancing */
/* This is just the MTS section of an init.ora file */
MTS_SERVICE=MYSID
MTS_SERVERS=10
MTS_MAX_SERVERS=100
MTS_DISPATCHERS="(protocol=tcp)(pool=yes)(mult=yes)"
MTS_DISPATCHERS="(protocol=spx)(pool=yes)(mult=yes)"
MTS_DISPATCHERS="(protocol=ipc)(pool=yes)(mult=yes)"
MTS_MAX_DISPATCHERS=100
MTS_LISTENER_ADDRESS=
"(address=(protocol=tcp)(host=myhost)(port=1521))"
MTS_LISTENER_ADDRESS=
"(address=(protocol=tcp)(host=cman)(port=1610))"
MTS_LISTENER_ADDRESS=
"(address=(protocol=spx)(service=cman))"
MTS_LISTENER_ADDRESS=
"(address=(protocol=ipc)(key=mysid))"
MTS_MULTIPLE_LISTENERS=TRUE
```

同时，你可能需要设置 SESSIONS 参数，它告诉实例启用集中特性（多路复用和连接共享池）打开的最大激活的会话数。

```
TNSNAMES.ORA
/* To enable multiplexing, connection pooling, and bridging */

mysid=(description=
  (address_list=
    (address=(protocol=tcp)(host=myhost)(port=1521))
    (address=(protocol=tcp)(host=cman)(port=1610))
  )
  (address=(protocol=spx)(service=cman))
  (address=(protocol=ipc)(key=mysid))
  (connect_data=(sid=mysid))
  (source_route=yes)
))

LISTENER.ORA
/* The foundation of all the addressing */

listener=(address_list=
  (address=(protocol=tcp)(host=myhost)(port=1521))
  (address=(protocol=tcp)(host=myhost)(port=1526))
  (address=(protocol=spx)(service=cman))
  (address=(protocol=ipc)(key=mysid))
  sid_list_listener=
    (sid_list=
      (sid_desc=mysid)
      (sid_desc=another_sid))
    )
)

CMAN.ORA
/* To enable firewall and set the CMAN profile */
```

```
cman=(address_list=
(address=(protocol=spx)(service=cman))
(address=(protocol=tcp)(host=cman)(port=1610))
)

cman_rules=(rules_list=
(rule=
(src=good_host2)(dst=myhost)(srv=mysid)(act=accept))
(rule=
(src=bad_host1)(dst=myhost)(srv=mysid)(act=reject))
)

cman_profile=
(parameter_list=
(maximum_relays=512)
(log_level=1)
(tracing=yes)
(relay_statistics=yes)
(show_tns_info=yes)
(use_async_call=yes)
(authentication_level=1)
```