

ORACLE 数据库连接配置浅析及故障排除

琚玲 赵芳

(国家气象信息中心技术开发室, 北京 100081)

摘要 ORACLE 是当今世界使用最广泛的商用数据库软件系统之一, 该系统功能强大同时也具有一定的复杂性。文章首先分析了 ORACLE 数据库的结构以及它的客户端服务器通信方式, 在此基础上分别研究了网络通信方式和进程通信方式下 ORACLE 客户端连接 ORACLE 服务器的方法, 并阐述了不同连接方法下如何从客户端和服务端进行配置, 最后以在实际工作中出现的两例客户端连接 ORACLE 故障为例, 给出了相关的故障解决方法。

关键词 ORACLE 客户端 连接 故障排除

引言

ORACLE 是当今世界使用最广泛的商用数据库软件系统之一, 它是一个极其强大、灵活和复杂的系统, 可以用于构造从小型的单用户系统到支持若干个并发用户的大中型应用项目。ORACLE 是 C/S(客户端/服务器)结构的数据库系统, 该种结构的数据库应用由两部分组成, 即客户应用程序和数据库服务器程序。这种结构的优点是能充分发挥客户端 PC 的处理能力, 很多工作可以在客户端处理后再提交给服务器。但同时由于存在客户端与服务器之间的通信, 使得系统有一定的复杂性, 因此用户在不清楚地了解结构时, 在使用过程中尤其在客户端连接数据库时很容易出现一些问题。本文阐述了客户端连接 ORACLE 服务器的几种方法以及如何正确配置, 并以在实际工作中出现的两例客户端连接 ORACLE 故障为例, 给出相关的故障解决方法。

1 ORACLE 数据库客户端服务器通信方式

ORACLE 是 C/S 结构的系统, 当客户端和服务端位于同一台机器时, 可以有两种通信方式: 网络通信和进程(IPC)通信。当客户端和服务端分别位于两台机器时, 只有网络通信一种方式。根据通信方式的不同对 ORACLE 数据库的连接配置也不同, 需要在客户端和服务端进行不同的配置, 然后才能以相应的连接方式登录数据库。

2 ORACLE 数据库连接的配置方法

2.1 网络通信方式下的连接配置

当客户端和服务端使用网络通信进行连接时, ORACLE 数据库需要通过服务器端的监听器(监听进程)来为数据库监听连接请求。此种通信方式下, 对于 ORACLE 数据库的连接配置需要从服务器端和客户端两方面做配置。

对于服务器端的配置主要是通过配置监听器文件来完成。而对于客户端的配置是通过配置连接描述符完成, ORACLE 对这些配置称之为命名方法。命名方法有本地命名、主机命名、目录命名等命名方法。本文分析的是其中最常用的本地命名和主机命名方法, 根据这两种不同的客户端命名方法相应地对服务器端的配置和客户端的配置也不一样。

2.1.1 服务器端配置

负责为数据库服务器监听连接的监听器实际上是一个进程, 它代表数据库监听有哪些客户端应用需要与数据库服务器建立连接。要使监听器能正常工作, 在启动它之前必须先配置监听器文件。监听器文件名为 listener. ora, 它必须和监听器在同一台机器上, 在 Unix 平台该文件缺省位于服务器端的 MYMORACLE_HOME/network/admin 目录。

(1) 本地命名方法下的监听器配置。

使用本地命名方法时, 配置监听器文件需要配置监听器名、监听器地址以及监听的数据库这几项

参数。①监听器名:配置监听器文件时首先要指定监听器的名字,这个名字可以随意取,但要保证每个监听器名字的唯一性。②监听器地址:配置监听器地址需要指定监听器监听的连接所使用的协议、监听的数据库主机名以及监听的端口号。连接协议(关键字为 PROTOCOL)视客户端和服务端之间连接方式的不同可设为 TCP 或 TCPS 等。监听的数据库主机名(关键字为 HOST)可以是主机名也可以是 IP 地址,建议使用主机名,这样当数据库主机的 IP 地址变更时不需要修改监听器文件。监听的端口号(关键字为 PORT)必须与客户端的本地命名文件(tnsnames.ora)保持一致。ORACLE 默认的连接类型是 TCP,端口号是 1521。如果使用默认的端口号,则对于监听器的配置只需要配置监听器文件。如果不使用默认的端口号,则除了配置监听器文件外,还需要在服务器端的 ORACLE 参数文件 PFILE 中定义本地监听器名(对应参数为 local_listener),监听器名与监听器配置文件中的保持一致。同时还要在客户端的本地命名文件(tnsnames.ora)对监听器进行描述,描述的参数包括监听器名、使用的协议、数据库所在的主机名以及监听的端口,这些参数的值必须和监听器配置文件中的保持一致。③监听器所监听的数据库:配置监听的数据库需要指定数据库的实例名(关键字为 SID_NAME)以及 ORACLE 服务的主目录(关键字为 ORACLE_HOME),使得监听器能够找到数据库所在的位置以及 ORACLE 配置文件、可执行程序所在的位置。数据库实例名在 ORACLE 初始化参数文件的实例名参数(instance_name)中定义。ORACLE 服务主目录在操作系统的配置文件中有定义。一个监听器可以监听一台机器上的多个数据库。如果要监听多个数据库,则需要依次配置每个数据库的实例名以及它所对应的服务主目录。

(2)主机命名方法下的监听器配置。

主机命名方法是使用主机名来连接数据库。使用这种方法无论是对监听器的配置还是对客户端的配置都很简单:只需在监听器文件的配置中指定全局数据库名,即在监听器文件中配置好全局数据库(关键字为 GLOBAL_DBNAME)参数(该参数值可以从 ORACLE 初始化参数文件中的 SERVICE_NAMES 参数或 DB_NAME 和 DB_DOMAIN 参数中得到)。在此种命名方法下,用户只需提供主机名

就可以建立一个连接,不需要在客户端创建和维护本地命名配置文件(tnsnames.ora)。不过每个结点上只有一个数据库实例可以用主机命名方法,其他的实例只能用本地命名方法。

2.1.2 客户端配置

对客户端的配置主要是对连接描述符进行定义。连接描述符定义后,ORACLE 客户端就可以通过输入数据库账户名和口令以及连接描述符与数据库建立连接。

(1)本地命名方法。

本地命名方法是通过配置客户端访问方式文件(sqlnet.ora)和本地命名配置文件(tnsnames.ora)来连接数据库(在 Unix 平台,该文件默认位于服务器端的 MYMORACLE_HOME/network/admin 目录)。

本地命名方法需要进行以下配置:①配置访问方式文件。访问方式文件指明客户端以何种命名方法连接数据库。该文件中的目录路径参数(关键字为 NAMES_DIRECTORY_PATH)控制 ORACLE 网络服务采用哪种命名方式作连接描述符连接数据库。在访问方式文件中指明客户端是以本地命名方法(关键字为 TNSNAMES)。②配置本地命名文件。本地命名文件主要是用来存储网络服务名、访问的数据库地址、数据库服务名等信息。网络服务名,实际上就是连接描述符,它的名字可以随便定义。但一旦定义好后,客户端登录数据库的连接描述符必须与该参数保持一致。访问的数据库地址,需要指定访问协议(PROTOCOL)、数据库主机名(HOST)、访问端口(PORT)。这 3 项参数的配置只要与前面配置好的监听器文件保持一致即可。数据库服务名(关键字为 SERVICE_NAME)要与 ORACLE 初始化参数文件中定义的服务名(关键字为 SERVICE_NAMES)保持一致。

完成对访问方式文件和本地命名文件的配置后,就可以使用本地命名文件中定义的网络服务名作为连接描述符登录数据库。

(2)主机命名方法。

主机命名方法的配置比较简单,不需要创建和维护本地命名配置文件(tnsnames.ora),只需要配置访问方式文件(sqlnet.ora),在该文件中指定访问方式是主机访问方式,即将目录路径参数设为 HOSTNAME。

完成配置之后,在连接数据库时将监听器文件中的全局数据库名(GLOBAL_DBNAME)作为连接描述符就可以登录数据库。

2.2 进程通信方式下的连接配置

客户端与服务器在同一机器时,它们之间可以不通过网络通信,而直接通过进程间通信来实现。此时不需要配置监听器,客户端登录服务器也不需要连接描述符。配置比较简单,只需要配置客户端的操作系统文件:在操作系统配置文件中定义环境变量 ORACLE_SID。

ORACLE_SID 是操作系统的环境变量,它用于与操作系统交互,也就是说,使用本地进程通信访问 ORACLE 数据库必须通过获取 ORACLE_SID 的值。ORACLE_SID 的值必须与 ORACLE 实例名(INSTANCE_NAME)一致。

完成配置后,使用用户名连接数据库,不需要使用连接描述符。

3 ORACLE 数据库连接故障排除

下面结合笔者在实际工作中遇到的 ORACLE 数据库连接故障,分析 ORACLE 连接故障的排查过程。

3.1 故障背景

数据库系统版本是 ORACLE9i,该数据库系统安装在 HP-UX Unix 平台上。在应用中,为了减轻数据库服务器(SERVER)的处理压力,通常将客户端(CLIENT)和数据库服务器分别安装在两台机器上。由于实际工作中的一些原因,笔者的工作环境是将 ORACLE 客户端和服务端安装在同一台机器上。下文谈到的故障就是在这样的背景下产生的。

3.2 故障 1 及排查

数据库连接不上,错误信息如下:“ORA-12154: TNS:could not resolve service name。”

错误信息显示找不到数据库的服务名。

该故障是 C/S 在使用网络通信方式下客户端应用连接数据库时出现的,使用的是本地命名方法。以服务器端用户登录,在仔细检查了监听器文件、访问方式文件以及本地命名文件后,发现各项配置都没有问题。再检查客户端应用程序是否使用了连接描述符登录数据库,发现没有问题。在排查过程中发现,发现同样的连接描述符,以服务器端的用户登录可以连上数据库,这更证明各项配置是正确的。

然而以客户端用户登录却不能连接数据库,这究竟是什么原因呢?是否文件的访问权限出了问题?经进一步检查发现访问方式文件(sqlnet.ora)和本地命名文件(tnsnames.ora)只对服务器端用户有访问权限,而对客户端用户没有访问权限。因此,问题在于客户端找不到本地命名文件中的网络服务名,所以才会造成连接描述符无从匹配网络服务名。将这两个文件的访问权限赋予客户端用户,问题得以解决。但其实最初在配置客户端时是赋予了客户端访问这些文件的权限的,为什么在应用程序运行一段时间后突然没有权限了呢?与系统管理员联系,经过系统排查,告之并未发现有人为修改文件访问权限的记录。那为什么系统会自动改变文件的访问权限呢?在日常的业务运行过程中,曾经出现过磁盘空间满而导致文件的访问权限被自动修改。估计这次故障的出现可能是由于系统瞬间的磁盘空间满,而导致系统自动将访问方式文件及本地命名文件变成不可访问。

本故障难于排查的关键在于数据库服务器和客户端在同一台机器上,它们的配置文件位于同一台机器的相同目录,不容易察觉服务器端和客户端的用户对配置文件是否都有访问权限。因此有条件最好将客户端和服务端分别安装在两台机器上,这样便于维护和发现问题。

3.3 故障 2 及排查

故障 2 是在排除故障 1 时发生的,由于当时检查网络通信下的各项配置没有发现问题,一时又查不到原因,所以决定选择 C/S 的另一种通信方式——进程通信,在修改客户端应用程序的数据库登录方式后进行登录。结果发生数据库连接不上,错误信息如下:

“ORA-01034: ORACLE not available

ORA-27101: shared memory realm does not exist”

错误信息显示数据库不可用,共享内存区域不存在。

本故障是 C/S 在使用进程通信时出现的。2.2 节已经说明使用进程通信需要在操作系统配置文件中定义环境变量 ORACLE_SID。查看操作系统配置文件,发现该参数存在,是以前就设置好的,但由于之前使用 C/S 网络通信配置,所以一直未使用。在检查了 ORACLE_SID 的值后,发现它和 ORA-

CLE 实例名不相符,将 ORACLE_SID 改成与实例名一致后问题得以解决。

4 结束语

ORACLE 数据库系统功能很强大,但相对来说也增加了操作的复杂度,在使用过程中要多查阅 ORACLE 技术文档,理清思路,在遇到问题、排查故障原因时更要多查看技术文档,同时也要拓宽思路,不仅要从数据库的角度去排查,还要考虑到操作系

统的层面上,因为系统之间不是孤立的。

参考文献

[1] ORACLE University. Oracle9i:Database Administration Fundamentals II (volume 1)[R]. Oracle University, 2002.

[2] ORACLE Corporation. Oracle9i:Database Concepts[R]. ORACLE Corporation, 2002.

[3] Loney K, Theriault M, 蒋蕊. Oracle9i DBA 手册[M]. 北京:机械工业出版社, 2002.

Configuration Method and Troubleshooting of
ORACLE Database Connection

Ju Ling Zhao Fang

(National Meteorological Information Center, Beijing 100081)

Abstract: ORACLE, being powerful and complicated, is one of the most popular business database software systems. The emphasis of the study is put on the configuration connection method. The structure of ORACLE database and its communication mode between client and server are discussed, based on which the connection method between ORACLE client and server is explained by networking and process communication modes, respectively. The configurations from client side and server side under the different connection modes are described. Two solutions to connection malfunctions are given as examples.

Key words: ORACLE, client, database connection, troubleshooting