

在Oracle8x中实现自动断开后再连接（1）PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/143/2021_2022__E5_9C_A8Oracle8_c102_143326.htm 在实际的数据库应用中，我们经常遇到这样一个问题，连接到Oracle数据库的用户在作了一次操作后，再也没有后续操作，但却长时间没有和数据库断开连接。对于一个小型的应用系统来讲，本身的连接数目就有限，这好像没有什么严重的后果，但如果对于一个大型的数据库应用。如税务、工商等，如果数据库的连接数目很多，对于数据库服务器来讲，多一个连接就要多消耗一份资源，如果大量用户连接进入数据库系统但却不进行任何的操作，这无形之中就白白造成了服务器系统资源的浪费，同时造成服务器负载的提高，对于那些确实在工作的用户来讲，就不能最大限度的利用服务器的资源，严重情况下可能造成系统性能的急剧下降。针对这种问题，该如何处理呢？对于目前流行的三层结构（Browser/Application/Server）开发来讲，这个问题可以通过设置应用服务器端的连接共享池（shared pool）来避免。但对于传统的两层结构（Client/Server）应用，就必须由我们人为干预来避免这种资源浪费情况。具体可以通过一个后台任务来监控系统中的所有进程，对于那些空闲超过一定时间的进程采取一定的特殊处理措施，如在客户端提示用户连接时间太长，如果没有后续操作系统将自动杀掉该连接或者直接将该空闲连接杀掉。下面就来具体讨论如何在Oracle 8x环境下实现用户进程的自动监控及其对超过一定空闲时间连接的处理办法。

一、识别系统中超过一定空闲时间的连接 要实现后台任务自动对超过一定时间空闲连接的

处理，首先第一步工作就是要从所有与数据库服务器的连接中识别出那些连接需要处理，也就是需要获得与服务器连接的每个用户的登陆时间及其最后一次操作后的空闲时间。在Oracle系统中，有一个动态性能视图v\$session，该视图保存着系统当前连接的各种动态信息。其中，有两个字段LOGON_TIME和LAST_CALL_ET可以得到上面的两个答案。LOGON_TIME是一个日期型（Date）字段，为用户登陆时间；LAST_CALL_ET是一个数字型（Number）字段，其含义是用户最后一条语句执行完毕后的时间，单位为秒。每次用户执行一个新的语句后，该字段复位为0，重新开始记数。我们可以通过该字段来获得一个连接用户最后一次操作数据库后的空闲时间。下面的SQL查询语句可以得到与当前数据库连接的所有用户的一些基本情况，如用户名、状态、连接机器的名称，操作系统中用户的名称，UNIX系统的进程号，在UNIX操作系统级断开连接的语句，Oracle数据库系统断开连接的语句，登陆时间以及最后一次操作到现在的空闲时间等等。

```
SELECT s.username 用户名, s.status 状态, s.machine 机器名称, osuser 操作系统用户名, spid UNIX进程号, kill -9 ||spid UNIX级断开连接, alter system kill session ||s.sid||, ||s.serial#|| . Oracle级断开连接, TO_CHAR (logon_time, dd/mm/yyyy hh24:mi:ss) 登陆时间, last_call_et 空闲时间秒, TO_CHAR (TRUNC (last_call_et / 3600, 0))|| || HRS || TO_CHAR (TRUNC ((last_call_et - TRUNC (last_call_et / 3600, 0) * 3600) / 60, 0) ) || MINS 空闲时间小时分钟, module 模块 FROM v$session s, v$process p WHERE TYPE = USER AND p.addr = s.paddr AND status != KILLED -- AND SUBSTR (machine, 1, 19) NOT IN (机器
```

名) AND last_call_et > 60 * 60 * 1 -- 空闲时间超过1小时的连接
ORDER BY last_call_et desc.在上面的查询中，我们可以通过
SUBSTR (machine, 1, 19) NOT IN (机器名)这个条件来屏蔽一些
机器，这些机器可能需要运行一些耗费很长时间的SQL语
句或其他一些特殊情况的机器。屏蔽这些机器的原因就是
在后面的后台自动识别及处理任务中对这些机器不作处理。
100Test 下载频道开通，各类考试题目直接下载。详细请访问
www.100test.com