<https://blogs.oracle.com/database4cn/为什么数据库中大量的server-process没有对应的session>

为什么数据库中大量的server process没有对应的session?

By: [Shunlun Pan](https://blogs.oracle.com/author/shunlun-pan)

在什么情况下oracle server process没有对应的session？除了px进程以外，还有没有其他情况？下面的一个真实的案例中，有600个个server process没有session对应，并且这样的process越来越多，似乎永远也不自己释放，原因是什么呢？

首先查询一下v$process和v$session，观察差异，可见差异有600多个。

select \* from v$process;

1674 rows selected.

select \* from v$session;

1051 rows selected.

比对了一下，v$process中除了32个px进程以外，还有大量的普通server process，并且它们已经存在了很多天了。

select \* from v$process p where p.addr not in (select addr from v$process p,v$session s where s.creator\_addr=p.addr) order by pid;

SPID    PROGRAM

29674    oracle@mydb01 (P000)

29820    oracle@mydb01 (P001)

29822    oracle@mydb01 (P002)

......

31415    oracle@mydb01 (P031)

31417    oracle@mydb01 (P032)

31419    oracle@mydb01 (P033)

7523    oracle@mydb01

27740    oracle@mydb01

3283    oracle@mydb01

29116    oracle@mydb01

20006    oracle@mydb01

12831    oracle@mydb01

15681    oracle@mydb01

2139    oracle@mydb01

8606    oracle@mydb01

3963    oracle@mydb01

31811    oracle@mydb01

1613    oracle@mydb01

......

先确认一下当前时间是2017年10月12日:

# date

Thu Oct 12 11:03:54 CST 2017

然后选取一个v$process中没有session的进程，比如以上输出中的最后一行的pid为1613的进程，通过ps来观察其状态:

# ps -aeo user,pid,ppid,pri,pcpu,pmem,vsize,rssize,wchan:25,s,start,cputime,command

USER       PID  PPID PRI %CPU %MEM    VSZ   RSS WCHAN                     S  STARTED     TIME COMMAND

oracle 1613 1 19 0.0 0.0 365768 27636 sk\_wait\_data S Oct 04 00:00:00 oraclemydb01 (LOCAL=NO)

发现以上进程在10月4日就存在了，也就是已经过了8天了。同时做一次system state dump，找到这个进程1613

PROCESS 1560:

----------------------------------------

SO: 0x73a3d79790, type: 2, owner: (nil), flag: INIT/-/-/0x00 if: 0x3 c: 0x3

proc=0x73a3d79790, name=process, file=ksu.h LINE:13949, pg=0 conuid=0

(process) Oracle pid:1560, ser:174, calls cur/top: (nil)/0x71232cdd70

......

Process Group: DEFAULT, pseudo proc: 0x7323de7370

O/S info: user: grid, term: UNKNOWN, ospid: 1613 <<<<<< 这里可见进程 1613

OSD pid info:

Short stack dump:

ksedsts <- ksdxfstk <- ksdxcb <- sspuser <- \_\_sighandler

<- read <- nttfprd <- nsbasic\_brc <- nsbrecv <- nioqrc

<- opikndf2 <- opitsk <- opiino <- opiodr <- opidrv

<- sou2o <- opimai\_real <- ssthrdmain <- main

可以看到以上进程的call stack都是一些网络相关的function call，既然与网络有关系，不妨取一下其打开的tcp连接:

# lsof -p 1613 -Pnai TCP

COMMAND     PID USER   FD   TYPE DEVICE SIZE/OFF NODE NAME

oracle\_16 1613 oracle 20u IPv4 1303337121 0t0 TCP 10.5.20.110:1521->10.5.70.234:23966 (ESTABLISHED)

可见server端(10.5.20.110)的确和一个client端(10.5.70.234)建立了连接，找到client端(10.5.70.234)，在client上用如下命令观察是什么进程建立了连接：

# lsof -Pni :5926

COMMAND   PID   USER   FD   TYPE DEVICE SIZE NODE NAME

sqlplus 29544 oracle    9u  IPv4 739424       TCP 10.5.70.234:23966->10.5.20.110:1521 (ESTABLISHED)

原来客户端是一个sqlplus进程，进程号为29544，但是什么情况下sqlplus连接数据库后只有server process而没有session？最大的可能性是这个进程正处于密码验证阶段。

检查一下调用这个sqlplus的脚本，果然发现脚本中指定的登录数据库的密码是错误的！

接下来在12.2.0.1的测试机上模拟一下，sqlplus故意以错误密码登录，然后停住不动等1分钟，看能否重现这个现象。

sqlplus tony/a@server\_r12201

SQL\*Plus: Release 12.2.0.1.0 Production on Thu Nov 2 11:50:46 2017

Copyright (c) 1982, 2016, Oracle.  All rights reserved.

ERROR:

ORA-01017: invalid username/password; logon denied

Enter user-name:   <<<<停在这里不要动

在server上通过两次ps定位是哪个进程被最新创建:

# ps -ef | egrep "oracleR12201 \(LOCAL=NO\)|PID" | grep -v grep

UID        PID  PPID  C STIME TTY          TIME CMD

oracle     962     1  0 Oct31 ?        00:00:04 oracleR12201 (LOCAL=NO)

# ps -ef | egrep "oracleR12201 \(LOCAL=NO\)|PID" | grep -v grep

UID        PID  PPID  C STIME TTY          TIME CMD

oracle     962     1  0 Oct31 ?        00:00:04 oracleR12201 (LOCAL=NO)

oracle   28540     1  0 11:50 ?        00:00:00 oracleR12201 (LOCAL=NO)

两次ps对比可见新产生了进程28540，查询v$process v$session

select \* from v$process where spid=28540;

ADDR    SPID    PROGRAM

000000006A576D48    28540    oracle@testsever

select \* from v$session where creator\_addr='000000006A576D48';

no rows selected.

看来问题重现了，以上的进程28540在v$process中存在但是在v$session中不存在。

但是，在我们的测试中，进程28540会在大约1分钟后自行从server端退出，并且在alert log中打印如下信息

2017-11-02T11:51:02.509020+08:00

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Fatal NI connect error 12170.

  VERSION INFORMATION:

    TNS for Linux: Version 12.2.0.1.0 - Production

    Oracle Bequeath NT Protocol Adapter for Linux: Version 12.2.0.1.0 - Production

    TCP/IP NT Protocol Adapter for Linux: Version 12.2.0.1.0 - Production

  Time: 02-NOV-2017 11:51:02

  Tracing not turned on.

  Tns error struct:

    ns main err code: 12535

TNS-12535: TNS:operation timed out

    ns secondary err code: 12606

    nt main err code: 0

    nt secondary err code: 0

    nt OS err code: 0

  Client address: (ADDRESS=(PROTOCOL=tcp)(HOST=10.182.211.94)(PORT=51681))

2017-11-02T11:51:02.510302+08:00

WARNING: inbound connection timed out (ORA-3136)

看到这个错误可能大家马上想到了SQLNET.INBOUND\_CONNECT\_TIMEOUT这个Oracle\*Net参数，的确，问题是这个参数引发的。在server端打开$ORACLE\_HOME/network/admin/sqlnet.ora，发现如下设置:

SQLNET.INBOUND\_CONNECT\_TIMEOUT=0

以上参数的含义是完成用户验证的最大时间，设置成0代表无限。也就是说，如果客户端输错了密码，只要客户端不关闭，那么对应的server process会永远等待client。

这个时候我们终于知道这个问题的原因了: 客户端输错了密码，停留在提示输入再次密码的界面上，并且数据库的server端的sqlnet.ora设置了SQLNET.INBOUND\_CONNECT\_TIMEOUT=0(无限时间)，导致server process一直等待客户端输入密码验证，因此永远也不自动退出。

这个问题实际上还是比较严重的，它可能会导致ORA-00020错误发生，并且导致新的session无法连接。