[**防火墙、DCD与TCP Keep alive**](http://www.laoxiong.net/firewall-dcd-and-tcp-keep-alive.html)

老熊 [Oracle数据库管理](http://www.laoxiong.net/category/oracle/oraman) 2012-01-17

在以前我写的一篇文章[《Oracle与防火墙》](http://www.laoxiong.net/oracle_and_firewall.html)中提到，网络防火墙会切断长时间空闲的TCP连接，这个空闲时间具体多长可以在防火墙内部进行设置。防火墙切断连接之后，会有下面的可能：

* + 切断连接之前，连接对应的Oracle会话正在执行一个耗时特别长的SQL，比如存储过程而在此过程中没有任何数据输出到客户端，这样当SQL执行完成之后，向客户端返回结果时，如果TCP连接已经被防火墙中断，这时候显然会出现错误，连接中断，那么会话也就会中断。但是客户端还不知道，会一直处于等待服务器返回结果的状态。
  + 切断连接之前，Oracle会话一直处于空闲状态，在防火墙中断之后，客户端向Oracle服务器提交SQL时，由于TCP连接已经中断，这时客户端侦测到连接中断，那么客户端就会报ORA-3113/ORA-3114这类错误，然后会话中断。但是在Oracle服务器端，会话一直在处于等待客户端消息的状态。
  + 当然，如果服务器和客户端都一直闲着，没有任何消息往来，那么客户端和客户端的会话就一直存在，直到客户端发送消息即提交SQL到Oracle服务器。

从上面的前面2种情况来看，防火墙切断数据库TCP连接，引起的后果就会有：

* + 客户端报ORA-3113/3114错误，对这于长连接的后台应用不是个好事，特别是对那种C/C++开发的后台应用没有重连机制，就会出现问题或者程序退出。如果应用在一个较长的时间内没有任何活动，而这个时间超过了防火墙的设置，那应用的连接被中断。
  + 对于有连接池或重连机制的应用，如果连接池过大，导致空闲连接过多，或者是防火墙的连接断开时间过短同时应用太闲，那么连接频繁地被中断，而在数据库服务器端，则连接越来越多，即会话数越来越多，甚至最终超过了数据为最大连接数。
  + 其他一些影响，比如你在跑一个脚本而长时间没有输出，结果防火墙切断了连接，你之前的工作就白做，当然这些影响范围都较小。

那么如何防止出现以上的问题？ 对于没有重连机制的长连接应用，临时的解决办法是调大防火墙的连接切断时长；或者应用端在闲时定期执行一条类似于select 1 from dual这样的SQL；或者在Oracle服务端开启DCD功能，而DCD（死连接检测，即Dead Connection Detection)的时间长度短于防火墙的连接切断时长；或者使用操作系统的tcp keep-alive功能。对于有重连机制或有连接池的应用，为了避免数据库端连接数满，可以使用DCD或tcp keep-alive功能，如果DCD或TCP keep-alive的时间短于防火墙连接切断时长，那么连接将不会中断，因为防火墙视连接为活动状态（不过按MOS文档的说法，DCD的包有可能被防火墙忽略，即用DCD使连接保持活动状态可能没有作用）。而如果DCD或TCP keep-alive的时间长于防火墙连接切断时长，那么连接被中断的会话和进程将得到清理。

这里简单介绍一下TCP keep-alive，顾名思议，就是让TCP连接保持存活状态，这是由TCP协议层实现的功能，也是在TCP连接空闲时间超过设置的时间，就会发送探测包，因此，这个功能跟Oracle的DCD是极为类似的，所不同的是，这是由协议层实现的功能，是不能通过trace应用进程来跟踪包的发送。

理论是完美的，而现实差距越很远。

下面是一个有关的案例。一套运行在AIX 5.3上的RAC数据库，数据库版本为10.2.0.4，这套数据库在sqlnet.ora文件中有设置sqlnet.expire\_time=5，后果却是，数据库的连接数过几天就要满一次，防火墙设置的时间是8小时，也就是说只会切断空闲时间超过8小时的连接，这个时间已经足够长，晚上系统太闲，总会有部分连接空闲时间过长，但是由于设置了DCD，连接应该不会被防火墙切断，而即使防火墙切断了连接，那么在DCD的作用下，数据库服务器端的会话和进程也应该被清理才是，即使DCD不起作用，那TCP keep-alive也应该起作用才是，那为什么会话和进程数越来越多，甚至达到了最大进程数或会话数呢？

检查应用服务器(weblogic）的连接池，并核对应用服务器到数据库的TCP连接，发现连接池监控中的连接数量并不多，而对比应用服务器上看到的到数据库数据库的TCP连接数量和数据库数据器上看到的应用服务器建立过来的TCP连接数量，二者之间差距非常大，前者远远小于后者，就是说，大量的连接已经被切断，应用服务器的连接已经退出，但是数据库端的连接还没有释放。从weblogic的日志来看，经常出现类似于连接失效并建立新的数据库连接这样的信息。

DCD为什么没有起作用，是不是DCD的BUG？DCD的确是有很多BUG，但是完全不工作的BUG应该是很少的。将sqlnet.expire\_time设为1，5以及其他值，都不起作用，甚至重启过主机都没有起作用（当然并不是专门为了DCD而重启主机）。

使用truss命令来跟踪Oracle进程：

[view plaincopy to clipboardprint?](http://www.laoxiong.net/firewall-dcd-and-tcp-keep-alive.html)

* + SQL> host
  + $ truss -p 828090
  + kread(0, 0x0000000000000000, 0) (sleeping...)
  + ...很长时间没有任何反应...
  + ^CPstatus: process is not stopped
  + $
  + $ exit
  + SQL> oradebug setospid 828090
  + Oracle pid: 247, Unix process pid: 828090, image: oracle@db2
  + SQL> oradebug short\_stack
  + ksdxfstk+002c<-ksdxcb+04e4<-sspuser+0074<-000044C0<-nttrd+0120<-nsprecv+07a0<-nsrdr+0114<-nsdo+1710<-nsbrecv+0040<-nioqrc+04a8<-opikndf2+0688<-opitsk+08a8<-opiino+0990<-opiodr+0ae0<-opidrv+0484<-sou2o+0090<-opimai\_real+01bc<-main+0098<-\_\_start+0098
  + SQL>  oradebug short\_stack
  + ksdxfstk+002c<-ksdxcb+04e4<-sspuser+0074<-000044C0<-nttrd+0120<-nsprecv+07a0<-nsrdr+0114<-nsdo+1710<-nsbrecv+0040<-nioqrc+04a8<-opikndf2+0688<-opitsk+08a8<-opiino+0990<-opiodr+0ae0<-opidrv+0484<-sou2o+0090<-opimai\_real+01bc<-main+0098<-\_\_start+0098

没有看到DCD起任何作用的迹象，使用truss跟踪其他多个空闲Oracle也是如此。

那么TCP keep-alive呢，怎么样来判断有没有起作用？在AIX上其答案是使用kdb（这个需要root用户权限）。

在笔记本电脑上使用sqlplus连接到数据库，在数据库主机上使用netstat -Aan | grep 1521 | grep “你的IP地址”，以获得sqlplus连接的TCP连接信息，显示的第1列是一串16数据，然后使用kdb：

[view plaincopy to clipboardprint?](http://www.laoxiong.net/firewall-dcd-and-tcp-keep-alive.html)

* + #kdb
  + The specified kernel file is a 64-bit kernel
  + Preserving 1418178 bytes of symbol table
  + First symbol \_\_mulh
  + START              END <name >
  + 0000000000001000 0000000003E5C050 start+000FD8
  + F00000002FF47600 F00000002FFDC940 \_\_ublock+000000
  + 000000002FF22FF4 000000002FF22FF8 environ+000000
  + 000000002FF22FF8 000000002FF22FFC errno+000000
  + F100070F00000000 F100070F10000000 pvproc+000000
  + F100070F10000000 F100070F18000000 pvthread+000000
  + PFT:
  + PVT:
  + id....................0002
  + raddr.....0000000002000000 eaddr.....F200800090000000
  + size..............00080000 align.............00001000
  + valid..1 ros....0 fixlmb.1 seg....0 wimg...2
  + Command enhancement entry point is called.
  + Welcome to VXDRV subcommands
  + Command enhancement entry for vxodmdb called.called.
  + Welcome to vxdrv subcommands
  + (0)> sockinfo f100060009b04398 tcpcb | egrep "KEEP|opts"
  + t\_timer....... 00000000 (TCPT\_KEEP)
  + timewait.prv@0000000000000000  inp\_v6opts  @0000000000000000
  + opts........ 0004 (REUSEADDR)

可以看到，这个TCP连接没有KEEPALIVE选项（属性），keep-alive对应的timer(TCPT\_KEEP)数值为0，表示没有设置timer时间。

会不会是DCD功能屏蔽了TCP keep-alive？

将sqlnet.ora中expire\_time设置去掉，然后重复上述步骤，这次有了变化:

[view plaincopy to clipboardprint?](http://www.laoxiong.net/firewall-dcd-and-tcp-keep-alive.html)

* + #kdb
  + The specified kernel file is a 64-bit kernel
  + Preserving 1418178 bytes of symbol table
  + First symbol \_\_mulh
  + START              END <name >
  + 0000000000001000 0000000003E5C050 start+000FD8
  + F00000002FF47600 F00000002FFDC940 \_\_ublock+000000
  + 000000002FF22FF4 000000002FF22FF8 environ+000000
  + 000000002FF22FF8 000000002FF22FFC errno+000000
  + F100070F00000000 F100070F10000000 pvproc+000000
  + F100070F10000000 F100070F18000000 pvthread+000000
  + PFT:
  + PVT:
  + id....................0002
  + raddr.....0000000002000000 eaddr.....F200800090000000
  + size..............00080000 align.............00001000
  + valid..1 ros....0 fixlmb.1 seg....0 wimg...2
  + Command enhancement entry point is called.
  + Welcome to VXDRV subcommands
  + Command enhancement entry for vxodmdb called.called.
  + Welcome to vxdrv subcommands
  + (0)> sockinfo f100060009c1b398 tcpcb | egrep "KEEP|opts"
  + t\_timer....... 00000464 (TCPT\_KEEP)
  + timewait.prv@0000000000000000  inp\_v6opts  @0000000000000000
  + opts........ 000C (REUSEADDR|KEEPALIVE)

可以看到，这个TCP连接已经有了KEEPALIVE选项(属性），同时keep-alive timer为16进制数464，即十进制1124，表示还有562秒（timer这里的时间以半秒为单位）就会发送keep-alive探测包。

反复通过加上DCD设置和去掉DCD设置进行测试，发现只要一加上DCD功能，那么Oracle进程就不会为TCP连接设置keep-alive选项（属性），反之则会设置keep-alive。从这里很明显地看出来，DCD功能开启后，不但没有起作用，还关闭了TCP keep-alive功能。

对于这套系统，最后去掉了DCD设置，并将所有的空闲连接清理掉，经过多天的观察，系统的连接数一直保持稳定，与应用服务器的实际连接数保持一致。

那是不是启用了DCD之后，TCP keep-alive就会被关闭，这里不能贸然下定论，不同的版本、不同的平台或许有差异，至少我一个同事在Linux上测试时，tcp keep-alive是共存的。

在诊断处理这个案例的过程之中，不得不提到另外一点，写在此处，与大家分享。在诊断问题的初期阶段，询问系统维护人员（通常连应用和数据库一起维护），应用服务器和数据库之间是否有防火墙，如果有，是怎么样设置的TCP连接断开时间。而维护人员最开始的回答是没有防火墙。从表面上看，也应该是没有防火墙。数据库服务器其中一个IP地址是192.168.8.42，而两台应用服务器服务器的IP地址是192.168.8.201和192.168.8.202，咋一看是在同一网段，中间应该不会有防火墙。然而仔细检查相关数据：

* + $ifconfig -a
  + ....省略部分输出....
  + en4: flags=5e080867,c0<up ,BROADCAST,DEBUG,NOTRAILERS,RUNNING,SIMPLEX,MULTICAST,GROUPRT,64BIT,CHECKSUM\_OFFLOAD(ACTIVE),PSEG,LARGESEND,CHAIN>
  + inet 192.168.8.42 netmask 0xffffff80 broadcast 192.168.8.127
  + inet 192.168.8.43 netmask 0xffffff80 broadcast 192.168.8.127
  + tcp\_sendspace 131072 tcp\_recvspace 65536 rfc1323 0

可以看到，实际上数据库跟应用服务器是在不同的子网之中。这里子网掩码为0xffffff80，也就是我们通常写的子网掩码255.255.255.128，这样0-127为一个子网，128-255为一个子网（当然子网的第1个和最后一个地址都是不能使用的）。那么最后一个数字为201和202的两个IP地址显然就在第2个子网中，子网之间仍然需要有路由功能的网络设备，那么在之间有防火墙也是有可能的。用上面的方法去检查应用服务器的网络配置，发现其被划在了更小的子网之中。系统维护人员最终确认应用服务器和数据库之间的确是有防火墙存在。因此我们可以总结出两点：1，全面的知识对诊断数据库问题很有帮助。2， 如果经过详尽分析问题发现与别人说的不一致，需要坚持，并需要为自己的坚持寻找更多的数据来证实。

--The End--