**如何诊断RAC数据库上的“IPC Send timeout”问题？**

By: [Jane Zhang](https://blogs.oracle.com/author/0d369878-b7d3-469f-a646-a700095834c7) | Senior Support Manager

   RAC 数据库上比较常见的一种问题就是“IPC Send timeout”。数据库Alert log中出现了“IPC Send timeout”之后，经常会伴随着ora-29740 或者 "Waiting for clusterware split-brain resolution"等，数据库实例会因此异常终止或者被驱逐出集群。

比如:

实例1的ALERT LOG：

Thu Jul 02 05:24:50 2012

**IPC Send timeout detected.Sender: ospid 6143755<==发送者**

**Receiver: inst 2 binc 1323620776 ospid 49715160<==接收者**

Thu Jul 02 05:24:51 2012

IPC Send timeout to 1.7 inc 120 for msg type 65516 from opid 13

Thu Jul 02 05:24:51 2012

Communications reconfiguration: instance\_number 2

**Waiting for clusterware split-brain resolution <==出现脑裂**

Thu Jul 02 05:24:51 2012

Trace dumping is performing id=[cdmp\_20120702052451]

Thu Jul 02 05:34:51 2012

**Evicting instance 2 from cluster <==过了10分钟，实例2被驱逐出集群**

实例2的ALERT LOG：

Thu Jul 02 05:24:50 2012

IPC Send timeout detected. Receiver ospid 49715160 <==接收者

Thu Jul 02 05:24:50 2012

Errors in file /u01/oracle/product/admin/sales/bdump/sales2\_lms6\_49715160.trc:

Thu Jul 02 05:24:51 2012

**Waiting for clusterware split-brain resolution**

Thu Jul 02 05:24:51 2012

Trace dumping is performing id=[cdmp\_20120702052451]

Thu Jul 02 05:35:02 2012

Errors in file /u01/oracle/product/admin/sales/bdump/sales2\_lmon\_6257780.trc:

**ORA-29740: evicted by member 0, group incarnation 122  <==实例2出现ORA-29740错误，并被驱逐出集群**

Thu Jul 02 05:35:02 2012

LMON: terminating instance due to error 29740

Thu Jul 02 05:35:02 2012

Errors in file /u01/oracle/product/admin/sales/bdump/sales2\_lms7\_49453031.trc:

ORA-29740: evicted by member , group incarnation

    在RAC实例间主要的通讯进程有LMON, LMD, LMS等进程。正常来说，当一个消息被发送给其它实例之后，发送者期望接收者会回复一个确认消息，但是如果这个确认消息没有在指定的时间内收到（默认300秒），发送者就会认为消息没有达到接收者，于是会出现“IPC Send timeout”问题。

    这种问题通常有以下几种可能性：

1. 网络问题造成丢包或者通讯异常。

2. 由于主机资源(CPU、内存、I/O等）问题造成这些进程无法被调度或者这些进程无响应。

3. Oracle Bug.

   在这方面的Oracle Bug不是太多，大多数时候都是网络或者资源问题造成这种情况。

   为了分析这种问题，网络和资源监控工具是非常必要的。推荐安装OSWBB，对于如何安装和使用OSWBB，请参考文章[利器OSW (OSWatcher Black Box) 之简介篇](https://blogs.oracle.com/Database4CN/entry/%E5%88%A9%E5%99%A8osw_oswatcher_black_box_%E4%B9%8B%E7%AE%80%E4%BB%8B%E7%AF%87)。

**下面是一个由于主机资源紧张造成的“IPC Send timeout”例子：**

   实例1的Alert log中显示接收者是2号机的进程1596935，

Fri Aug 01 02:04:29 2008

 IPC Send timeout detected.Sender: ospid 1506825 <==发送者

 Receiver: inst 2 binc -298848812 ospid 1596935  <==接收者

   查看当时2号机的OSWatcher的vmstat输出：

 zzz \*\*\*Fri Aug 01 02:01:51 CST 2008

 System Configuration: lcpu=32 mem=128000MB

 kthr     memory             page              faults        cpu

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

  r  b   avm   fre  re  pi  po  fr   sr  cy  in   sy  cs us sy id wa

 25  1 7532667 19073986   0   0   0   0    5   0 9328 88121 20430 32 10 47 11

**58**  0 7541201 19065392   0   0   0   0    0   0 11307 177425 10440 87 13  **0**  0 **<==idle的CPU为0，说明CPU100%被使用**

**61**  1 7552592 19053910   0   0   0   0    0   0 11122 206738 10970 85 15  **0**  0

 zzz \*\*\*Fri Aug 01 02:03:52 CST 2008

   System Configuration: lcpu=32 mem=128000MB

   kthr     memory             page              faults        cpu

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

  r  b   avm   fre  re  pi  po  fr   sr  cy  in   sy  cs us sy id wa

 25  1 7733673 18878037   0   0   0   0    5   0 9328 88123 20429 32 10 47 11

**81**  0 7737034 18874601   0   0   0   0    0   0 9081 209529 14509 87 13  **0**  0 **<==CPU的run queue非常高**

**80**  0 7736142 18875418   0   0   0   0    0   0 9765 156708 14997 91  9  **0**  0 **<==idle的CPU为0，说明CPU100%被使用**

  上面这个例子说明当主机CPU负载非常高的时候，接收进程无法响应发送者，从而引发了“IPC Send timeout”。

**下面是一个由于网络问题造成的“IPC Send timeout”例子：**

   实例1的Alert log中显示接收者是2号机的进程49715160，

Thu Jul 02 05:24:50 2012

IPC Send timeout detected.Sender: ospid 6143755 <==发送者

Receiver: inst 2 binc 1323620776 ospid 49715160 <==接收者

   查看当时2号机的OSWatcher的vmstat输出，没有发现CPU和内存紧张的问题，查看OSWatcher的netstat输出，在发生问题前几分钟，私网的网卡上有大量的网络包传输。

Node2:

zzz Thu Jul 02 05:12:38 CDT 2012

Name  Mtu   Network     Address            Ipkts Ierrs    Opkts Oerrs  Coll

en1   1500  10.182.3    10.182.3.2       **4073847798**     0 512851119     0     0 **<==4073847798 - 4073692530 = 155268 个包/30秒**

zzz Thu Jul 02 05:13:08 CDT 2012

Name  Mtu   Network     Address            Ipkts Ierrs    Opkts Oerrs  Coll

en1   1500  10.182.3    10.182.3.2       **4074082951**     0 513107924     0     0 **<==4074082951 - 4073847798 = 235153 个包/30秒**

Node1:

zzz Thu Jul 02 05:12:54 CDT 2012

Name  Mtu   Network     Address            Ipkts Ierrs    Opkts Oerrs  Coll

en1   1500  10.182.3    10.182.3.1       **502159550**     0 4079190700     0     0 **<==502159550 - 501938658 = 220892 个包/30秒**

zzz Thu Jul 02 05:13:25 CDT 2012

Name  Mtu   Network     Address            Ipkts Ierrs    Opkts Oerrs  Coll

en1   1500  10.182.3    10.182.3.1       502321317     0 4079342048     0     0 **<==502321317 - 502159550 = 161767 个包/30秒**

查看这个系统正常的时候，大概每30秒传输几千个包：

zzz Thu Jul 02 04:14:09 CDT 2012

Name  Mtu   Network     Address            Ipkts Ierrs    Opkts Oerrs  Coll

en1   1500  10.182.3    10.182.3.2       **4074126796**     0 513149195     0     0 **<==4074126796 - 4074122374 = 4422个包/30秒**

   这种突然的大量的网络传输可能会引发网络传输异常。对于这种情况，需要联系网管对网络进行检查。在某些案例中，重启私网交换机或者调换了交换机后问题不再发生。（请注意，网络的正常的传输量会根据硬件和业务的不同而不同。）

**下面是一个由于I/O问题造成的“IPC Send timeout”例子：**

   实例的Alert log中显示接收者是1号机的LMON进程：

Sun Feb 22 07:57:30 2014

IPC Send timeout detected. Receiver ospid 44105801 [oracle@db1 (LMON)] <========================接收者

查看这个进程生成的trace文件db1\_lmon\_44105801.trc，发现当时LMON的函数都是和IO有关的：

kjxgmpoll: stalled for 94 seconds (threshold 42 sec)

----- Call Stack Trace -----

skdstdst <- ksedst1 <- ksedst <- dbkedDefDump <- ksedmp

       <- ksdxfdmp <- ksdxcb <- sspuser <- 48bc <- sigthreadmask

        <- sslsstehdlr <- sslsshandler <- 48bc <- **skgfsio <- skgfqio**

**<- ksfd\_skgfqio <- ksfd\_io <- ksfdread <- kfk\_ufs\_sync\_io <- kfk\_submit\_ufs\_io**

**<- kfk\_submit\_io <- kfk\_io1 <- kfkRequest <- kfk\_transitIO <- kfioSubmitIO**

           <- kfioRequestPriv <- kfioRequest <- ksfd\_kfioRequest <- 576 <- ksfd\_osmio

            <- ksfd\_io <- ksfdread <- kccrbp <- kccgrd <- kjxgrf\_rr\_read

             <- kjxgrDD\_rr\_read <- kjxgrimember <- kjxggpoll <- kjfmact <- kjfdact

              <- kjfcln <- ksbrdp <- opirip <- opidrv <- sou2o

               <- opimai\_real <- ssthrdmain <- main <- start

   总结一下，对于“IPC Send timeout”：

1) 通过Oracle自带的CHM (Cluster Health Monitor)的输出来检查当时的资源、网络使用情况。CHM只在某些平台和版本上存在，关于CHM，请参考文章[11gR2 新特性：Oracle Cluster Health Monitor(CHM)简介](https://blogs.oracle.com/Database4CN/entry/11gr2_%E6%96%B0%E7%89%B9%E6%80%A7_oracle_cluster_health)。

2) 如果没有CHM，请安装[OSWBB](https://blogs.oracle.com/Database4CN/entry/%E5%88%A9%E5%99%A8osw_oswatcher_black_box_%E4%B9%8B%E7%AE%80%E4%BB%8B%E7%AF%87)来监控网络和主机资源。

3) 检查网络上是否有UDP或者IP包丢失的情况、网络上是否有错误。

4) 检查所有节点的网络设置是否正确。比如，所有节点MTU的设置必须是一致的，如果Jumbo Frame被使用的话，需要保证交换机可以支持MTU为9000.

5) 检查服务器是否有CPU使用率高或者内存不足的情况。

6) 检查实例被驱逐之前是否有数据库hang或者严重的性能问题。

  在下面的MOS文档中有针对“IPC Send timeout”的介绍：

  Top 5 issues for Instance Eviction (Doc ID 1374110.1)