**DG日志传输配置**

云和恩墨(北京)信息技术有限公司

技术顾问 燕鑫

http://www.enmotech.com

**文档控制：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序** | **版本号** | **更改人** | **日期** | **备注** |
| 1 | 1.0版 | 燕鑫 | 2019-03-11 | 初始版本 |
|  |  |  |  |  |
|  |  |  |  |  |

目录

[1. DG日志传输配置 - 4 -](#_Toc3230220)

# DG日志传输配置

Synchronous

The synchronous redo transport mode transmits redo data synchronously with respect to transaction commitment. A transaction cannot commit until all redo generated by that transaction has been successfully sent to every enabled redo transport destination that uses the synchronous redo transport mode.

Although there is no limit on the distance between a primary database and a SYNC redo transport destination, transaction commit latency increases as network latency increases between a primary database and a SYNC redo transport destination.

Asynchronous

The asynchronous redo transport mode transmits redo data asynchronously with respect to transaction commitment. A transaction can commit without waiting for the redo generated by that transaction to be successfully sent to any redo transport destination that uses the asynchronous redo transport mode.

The SYNC attribute specifies that the synchronous redo transport mode be used to send redo data to a redo transport destination.

The ASYNC attribute specifies that the asynchronous redo transport mode be used to send redo data to a redo transport destination. The asynchronous redo transport mode is used if neither the SYNC nor the ASYNC attribute is specified.

The NET\_TIMEOUT attribute specifies how long the LGWR process waits for an acknowledgement that redo data has been successfully received by a destination that uses the synchronous redo transport mode. If an acknowledgement is not received within NET\_TIMEOUT seconds, the redo transport connection is terminated and an error is logged.

Oracle recommends that the NET\_TIMEOUT attribute be specified whenever the synchronous redo transport mode is used, so that the maximum duration of a redo source database stall caused by a redo transport fault can be precisely controlled. See Monitoring Synchronous Redo Transport Response Time for information about monitoring synchronous redo transport mode response time.

Note:

You could also set the database initialization parameter, DATA\_GUARD\_SYNC\_LATENCY, which is global for all synchronous standby destinations. It defines the maximum amount of time (in seconds) that the primary database may wait before disconnecting subsequent destinations after at least one synchronous standby has acknowledged receipt of the redo.

For example, suppose you have three synchronous standby destinations and you set DATA\_GUARD\_SYNC\_LATENCY to a value of 2. If the first standby acknowledges receipt of the redo immediately, then the primary database waits no longer than 2 seconds for the other two standbys to respond. If one or both respond within 2 seconds, then they are maintained as active destinations. Destinations that do not respond in time are marked as failed. In both cases the primary remains in zero data loss protection mode because one synchronous standby has acknowledged receipt of the redo. Any failed synchronous standbys are reconnected as normal after the number of seconds specified for the REOPEN attribute have passed.

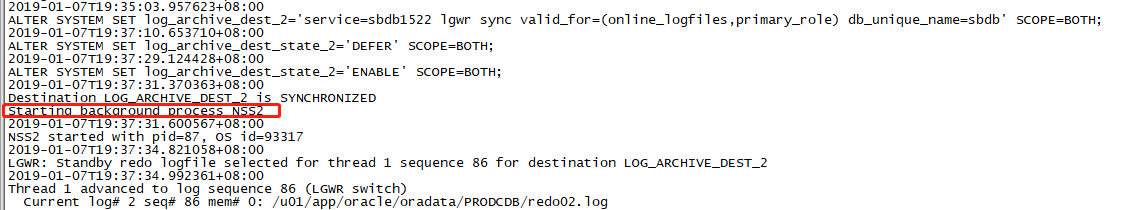
The AFFIRM attribute is used to specify that redo received from a redo source database is not acknowledged until it has been written to the standby redo log. The NOAFFIRM attribute is used to specify that received redo is acknowledged without waiting for received redo to be written to the standby redo log.

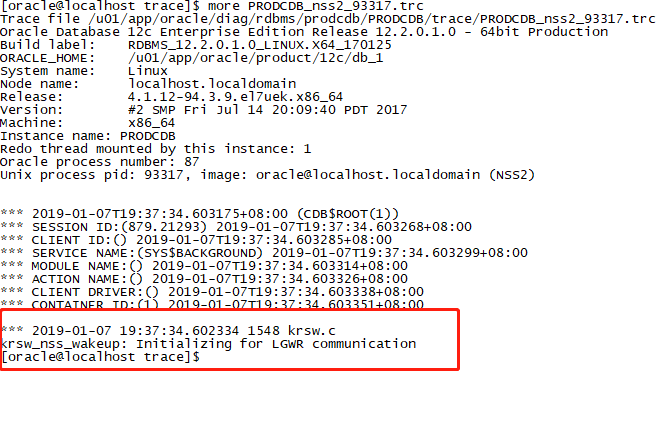
Redo received by a standby database is written directly to an archived redo log file if a standby redo log group is not available or if the redo was sent to resolve a redo gap. When this occurs, redo is written to the location specified by the LOCATION attribute of one LOG\_ARCHIVE\_DEST\_n parameter that is valid for archiving redo received from another database. The LOG\_ARCHIVE\_DEST\_n parameter that is used for this purpose is determined when the standby database is mounted, and this choice is reevaluated each time a LOG\_ARCHIVE\_DEST\_n parameter is modified.

nsa异步，nss同步。12c也是nss。

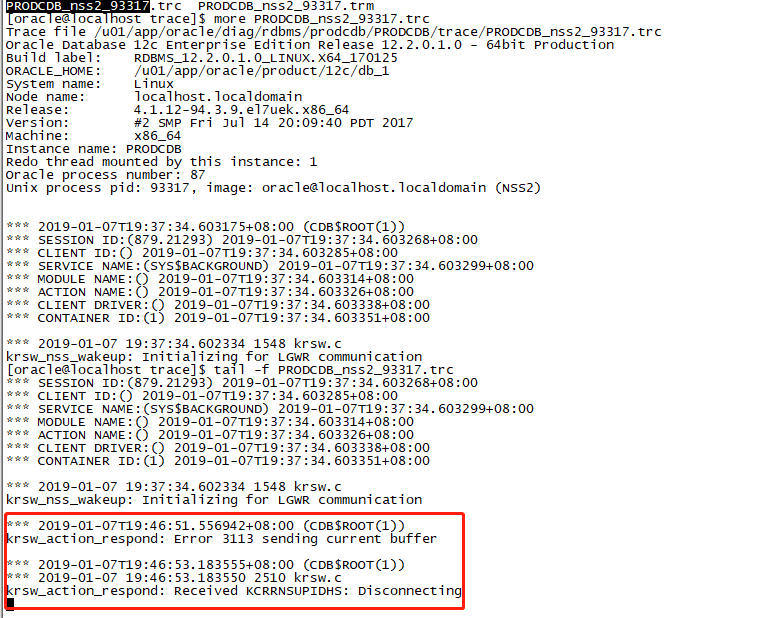
NSAn是11g的主库日志async传输进程，TTnn是12c的日志异步传输进程。

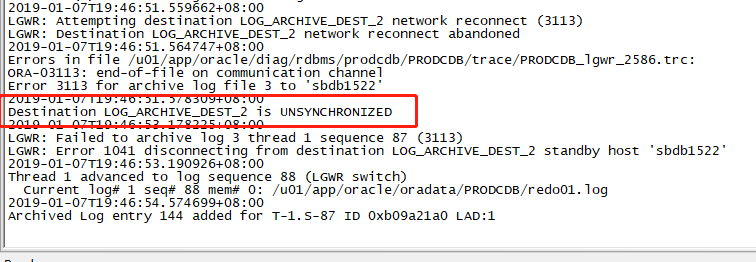
10g异步/同步传输的进程都是LNS。



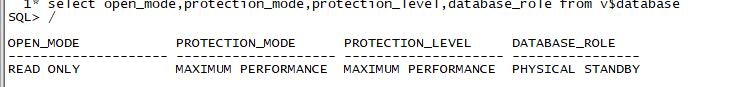
****

然后这时候我把备库停掉。

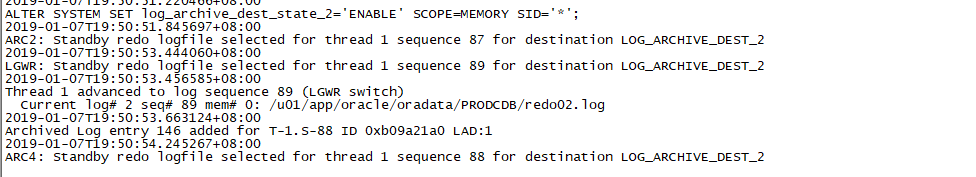
****

****

这时NSS还在。

****

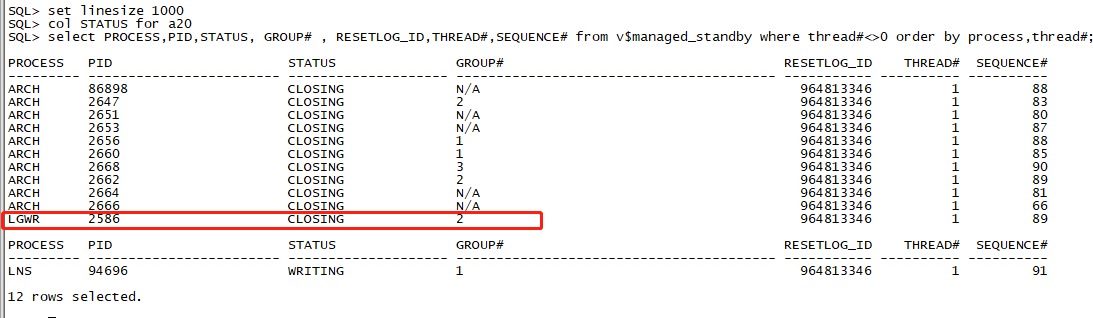
启动dg后，主库arch进程传归档到备库

****

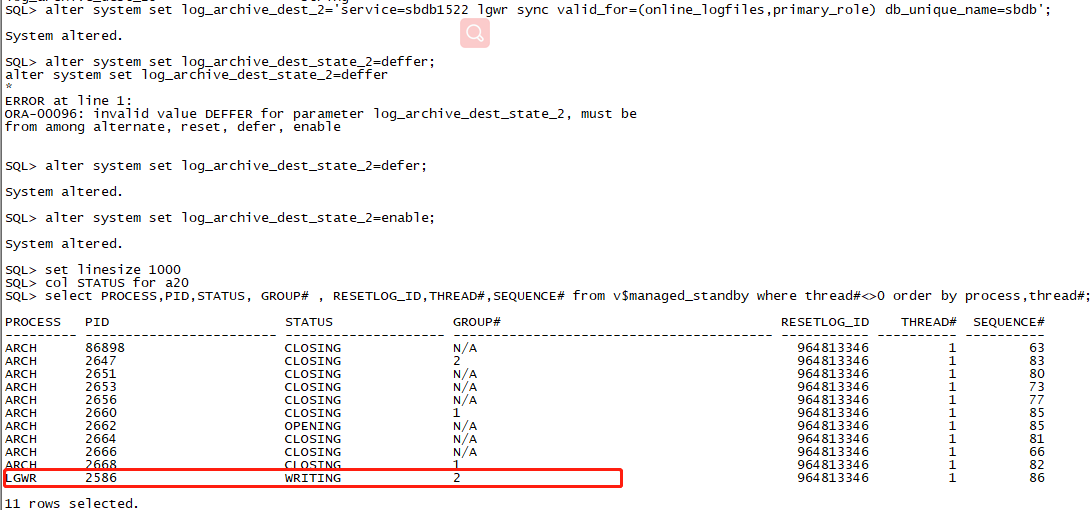
把主库改回async，主库nss日志如下

**C:\Users\yanxin\AppData\Local\Temp\WeChat Files\e3fde21ade0d1cd0c234de51ee6a682.png**

在主库上查询进程状态

****

如果是SYNC，则状态如下：

****