

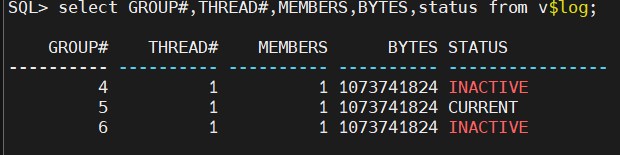
**increase redo log file size on oracle**

NOTED: increase the redo log file on database production with data guard, in case you don't have a data guard you can skip standby step.

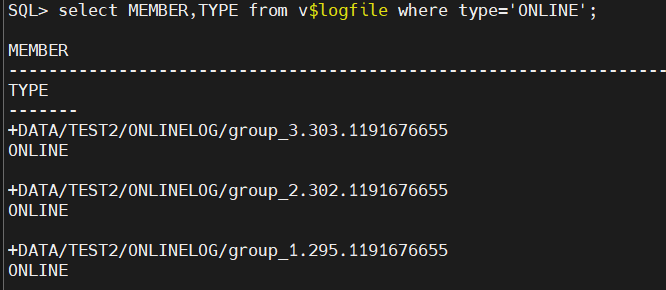
**Step 1: check current redo log groups, size and locations**

on primary database

SQL> select GROUP#,THREAD#,MEMBERS,BYTES,status from v$log;



SQL> select MEMBER,TYPE from v$logfile where type='ONLINE';



**Step 2: add new redo log group with new size**

SQL> ALTER DATABASE ADD  LOGFILE THREAD 1 ('+DATA','+RECO') SIZE 3G;

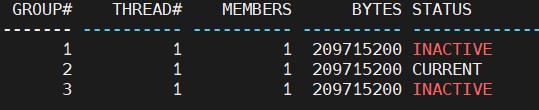
SQL> ALTER DATABASE ADD  LOGFILE THREAD 1 ('+DATA','+RECO') SIZE 3G;

SQL> ALTER DATABASE ADD  LOGFILE THREAD 2 ('+DATA','+RECO') SIZE 3G;

SQL> ALTER DATABASE ADD  LOGFILE THREAD 2 ('+DATA','+RECO') SIZE 3G;

**Step 3:drop the old redo log group when status is inactive**

SQL> select GROUP#,THREAD#,MEMBERS,BYTES,status from v$log;



**Step 3-1: to switch between group until the old redo log group to be inactive**

SQL> alter system switch logfile;

SQL> alter system checkpoint;

SQL> ALTER DATABASE DROP LOGFILE group 1;

SQL> ALTER DATABASE DROP LOGFILE group 2;

SQL> ALTER DATABASE DROP LOGFILE group 3;

**Step 4-1: recreate standby redo log group with new size as redo log group to solve the lag issue**

**Step 4-2:  stop the transport and apply from DG broker**

on primary

dgmgrl /

DGMGRL> show configuration

DGMGRL> edit database ECMCON set state='transport-off';

Succeeded.

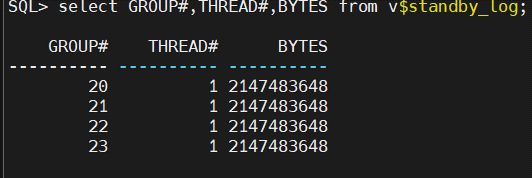
DGMGRL> edit database ecmcndr set state='apply-off';

Succeeded.

on primary and standby

**Step 4-3: check current standby redo log groups,size and locations**

SQL> select GROUP#,THREAD#,BYTES from v$standby\_log;



SQL> select MEMBER,TYPE from v$logfile where type='STANDBY';

**Step 5: drop standby redo log group**

SQL> alter database drop standby logfile group 5;

SQL> alter database drop standby logfile group 6;

SQL> alter database drop standby logfile group 7;

on standby

**Step 5-2: add new standby redo log group with new size**

SQL> alter database add standby logfile thread 1 group 5 ('+DATA','+RECO') size 3G reuse;

SQL> alter database add standby logfile thread 1 group 6 ('+DATA','+RECO') size 3G reuse;

SQL> alter database add standby logfile thread 1 group 7 ('+DATA','+RECO') size 3G reuse;

on primary

**Step 5-3: add new standby redo log group with new size**

SQL> alter database add standby logfile thread 1 group 5 ('+DATA','+RECO') size 3G reuse;

SQL> alter database add standby logfile thread 1 group 6 ('+DATA','+RECO') size 3G reuse;

SQL> alter database add standby logfile thread 1 group 7 ('+DATA','+RECO') size 3G reuse;

on standby

**Step 6: recreate redo log group with new size**

SQL> select GROUP#,THREAD#,MEMBERS,BYTES,status from v$log;

SQL> select MEMBER,TYPE from v$logfile where type='ONLINE';

SQL> show parameter STANDBY\_FILE\_MANAGEMENT

SQL> ALTER SYSTEM SET STANDBY\_FILE\_MANAGEMENT=manual scope=both sid='\*';

SQL> ALTER DATABASE ADD  LOGFILE THREAD 1 ('+DATA','+RECO') SIZE 3G;

SQL> ALTER DATABASE ADD  LOGFILE THREAD 1 ('+DATA','+RECO') SIZE 3G;

SQL> ALTER DATABASE ADD  LOGFILE THREAD 2 ('+DATA','+RECO') SIZE 3G;

SQL> ALTER DATABASE ADD  LOGFILE THREAD 2 ('+DATA','+RECO') SIZE 3G;

SQL> ALTER DATABASE DROP LOGFILE group 1;

SQL> ALTER DATABASE DROP LOGFILE group 2;

SQL> ALTER DATABASE DROP LOGFILE group 3;

SQL> ALTER DATABASE DROP LOGFILE group 4;

ALTER SYSTEM SET STANDBY\_FILE\_MANAGEMENT=AUTO scope=both sid='\*';

on primary

**Step 6-2: start the transport and apply from DG broker**

dgmgrl /

DGMGRL> show configuration

DGMGRL> edit database ECMCON set state='transport-on';

Succeeded.

DGMGRL> edit database ecmcndr set state='apply-on';

Succeeded.

In Oracle Database, a redo log is a crucial component used for data recovery and ensuring the consistency of transactions 🔄. It records all changes made to the database, including insertions, updates, and deletions 📝. These logs are stored in a circular fashion and consist of two main parts: the online redo log files and the redo log buffer 💾. When a transaction is committed, the changes are written to the redo log, ensuring that data can be recovered in case of a failure ⚡. Oracle uses a mechanism called "write-ahead logging" to ensure durability 🔒. Proper management of redo logs is essential for database performance and integrity 🛠️.