Practice 15

Performing Recovery Part V

Practice Target

In this practice you will gain perform recovery scenarios on the redo log files in ORADB database.

Practice Overview

In this practice, you will implement the following recovery scenarios:

- Recovery from loss of an inactive redo log group member
- Recovery from loss of an inactive redo log group on database startup
- Recovery from loss of an inactive redo log group while the database is in operation
- Recovery from loss of an active redo log group
- Recovery from loss of the current redo log group

Assumptions

This practice assumes the srv1 appliance is up and running and its database ORADB is running in OPEN state.

Pre-requisites

Take a snapshot of srv1 appliance. Name the snapshot “**Practice 15 Start**”.

Caution!

Do not implement the practice without creating the snapshot as instructed by the previous step.

Practice Preparation

In the following steps you will take a full backup of the database. Then you will create a script that will be used across this practice. The script simply retrieves the redo log groups, status of each group, members of each group and status of each member.

1. Open Putty and connect to srv1 as oracle user.

2. Invoke RMAN and login as target to the local database.

```
rman target /
```

3. Take a full backup of ORADB, then exit from RMAN.

```
BACKUP DATABASE TAG 'FULL_DB';
```

4. Create a file named display_logs.sql using vi editor

```
vi display_logs.sql
```

5. Enter the following code in it. Save and close the file

```
col MEMBER for a30

SELECT G.GROUP#, G.STATUS G_STATUS, M.STATUS M_STATUS,
  SUBSTR( MEMBER, INSTR(MEMBER,'o1'))  MEMBER
FROM V$LOG G, V$LOGFILE M
WHERE G.GROUP#=M.GROUP#
ORDER BY 1;
```

6. Start SQL*Plus and login to ORADB as sysdba.

```
sqlplus / as sysdba
```

7. Test running the script.

```
@display_logs
```

Recovery Scenario 14:

Recovery from Loss of an Inactive Redo Log Group Member

Scenario Target:

- In this scenario, you will recover from losing one member of an inactive online redo log group.

Scenario Assumptions:

- Database is using OMF
- Database is running

Simulating the data loss

8. Run the following query and select a **`GROUP#`** from the groups which their status is **`INACTIVE`**.

```
SELECT GROUP#, MEMBERS, STATUS, ARCHIVED FROM V$LOG ORDER BY 1;
```

9. Run the following query to return the members of the selected group.

```
SELECT MEMBER FROM V$LOGFILE WHERE GROUP#=<selected group#>;
```

10. Delete one of the group members.

```
host rm -f /u01/app/oracle/fra/ORADB/ORADB/onlinelog/<redo group member>
```

11. View the last lines in the alert log file and observe the reported errors.

The lost file is not yet recognized by the instance.

```
host tail /u01/app/oracle/diag/rdbms/oradb/ORADB/trace/alert_ORADB.log
```

12. Switch the redo log file twice and check the status of the selected group members after every time you switch the redo log files.

Observe that the status of the lost member does not change.

```
@display_logs

ALTER SYSTEM SWITCH LOGFILE;
```

13. View the last lines in the alert log file and observe the reported errors.

You should observe the error `ORA-00313: open failed for members of log group` *n* `of thread` *m*. This is how you are expected to know about the issue.

```
host tail /u01/app/oracle/diag/rdbms/oradb/ORADB/trace/alert_ORADB.log
```

Recovery Actions

Perform the following actions to restore the lost online redo group member.

14. Repeat running the script till the STATUS of the group with the lost member is INACTIVE. It may take a few minutes till the group turns INACTIVE.

```
@display_logs
```

15. Reinitialize the redo group.

Reinitializing a redo group creates new members in it.

```
ALTER DATABASE CLEAR LOGFILE GROUP <group#>;
```

Note: if you are wondering how Oracle knows in which locations the members should be created, here is an excerpt from Oracle documentation the explains how it works.

“If the DB_CREATE_ONLINE_LOG_DEST_n parameter is not set, but both the DB_CREATE_FILE_DEST and DB_RECOVERY_FILE_DEST initialization parameters are set, then the database creates one Oracle-managed log file member in each of those locations. The log file in the DB_CREATE_FILE_DEST destination is the first member.”

16. Run the script again.

Observe two important points:

- The status of the redo group change to UNUSED. This is the expected status as the redo group has just been re-created and has never been written to yet.
- The group has two new members. Even the one that was intact has been replaced with a new member.

```
@display_logs
```

17. Switch the redo log file as many as needed (twice or three times) till the group status is CURRENT.

```
ALTER SYSTEM SWITCH LOGFILE;
@display_logs
```

Recovery Scenario 15:

Recovery from Loss of an Inactive Redo Log Group on Database Startup

Scenario Target:

- In this scenario, you will recover from a situation when one inactive redo log group is unavailable when you restart the database. In other words, all the group members are unavailable when you start up the database.

Scenario Assumptions:

- Database is using OMF
- Database is down

Simulating the data loss

18. Start SQL*Plus and login to ORADB as sysdba.

```
sqlplus / as sysdba
```

19. Run the following query and select a GROUP# which its status is INACTIVE.

If no group is INACTIVE, wait for a couple of minutes and run the query again.

```
SELECT G.GROUP#, G.STATUS FROM V$LOG G ORDER BY 1;
```

20. Run the following query to retrieve members of the selected group. Take a note of the returned members.

```
col MEMBER for a150
SELECT MEMBER FROM V$LOGFILE WHERE GROUP#=<group#>;
```

21. Shutdown the database.

```
SHUTDOWN IMMEDIATE
```

22. Delete the selected group members.

```
host rm -f /u01/app/oracle/oradata/ORADB/onlinelog/o1_*_.log
host rm -f /u01/app/oracle/fra/ORADB/ORADB/onlinelog/o1_*_.log
```

23. Start up the database.

Starts up should fail.

```
STARTUP
```

Recovery Actions

Perform the following actions to restore the lost online redo group.

24. View the last lines in the alert log file and observe the reported errors.

You will see the error "ORA-00313: open failed for members of log group *n*"

```
host tail -n 50 /u01/app/oracle/diag/rdbms/oradb/ORADB/trace/alert_ORADB.log
```

25. Exit from SQL*Plus and invoke it again. Mount the database.

```
sqlplus / as sysdba
STARTUP MOUNT;
```

26. Reinitialize the redo group.

```
ALTER DATABASE CLEAR LOGFILE GROUP <group#>;
```

27. Open the database.

```
ALTER DATABASE OPEN;
```

28. Run the script.

```
@display_logs
```

29. Switch the redo log file till as many as needed (twice or three times) till the group status becomes CURRENT.

```
ALTER SYSTEM SWITCH LOGFILE;

@display_logs
```

Recovery Scenario 16:

Recovery from Loss of an Inactive Redo Log Group While the Database is in Operation

Scenario Target:

- In this scenario, you will recover from a situation when one inactive redo log group becomes unavailable while the database is on operation.

Scenario Assumptions:

- Database is using OMF
- Database is up and running

Simulating the data loss

30. Start SQL*Plus (if it is not already started) and login to ORADB as sysdba.

```
sqlplus / as sysdba
```

31. Run the following query and select a GROUP# which its status is INACTIVE.

If no group is INACTIVE, wait for a couple of minutes and run the query again.

```
SELECT G.GROUP#, G.STATUS FROM V$LOG G ORDER BY 1;
```

32. Run the following query to retrieve members of the selected group. Take a note of the returned members.

```
col MEMBER for a150
SELECT MEMBER FROM V$LOGFILE WHERE GROUP#=<group#>;
```

33. Delete the selected group members.

```
host rm -f /u01/app/oracle/oradata/ORADB/onlinelog/o1_*_.log
host rm -f /u01/app/oracle/fra/ORADB/ORADB/onlinelog/o1_*_.log
```

34. In the SQL*Plus session, verify the status of the database.

If the database is under constant data changes from client sessions, as it would be in normal operation, it keeps switching the log file until it eventually wants to make the inactive redo log group CURRENT and, because it cannot, it will fail.

```
host ps -ef | grep pmon
SELECT COUNT(*) FROM SOE.ORDERS;
```

35. Run the query script.

Unfortunately, the query does not tell about the incident.

```
@display_logs
```

Recovery Actions

Perform the following actions to restore the lost redo log group.

36. Reinitialize the lost redo group.

```
ALTER DATABASE CLEAR LOGFILE GROUP <group#>;
```

37. Run the query script again.

You should notice that new members are created in the group.

```
@display_logs
```

38. Switch the redo log file as many as needed (twice or three times) till the group status is `CURRENT`.

```
ALTER SYSTEM SWITCH LOGFILE;

@display_logs
```

Recovery Scenario 17:

Recovery from Loss of an Active Redo Log Group

Scenario Target:

- In this scenario, you will recover from losing an active redo log group.

Scenario Assumptions:

- Database is using OMF
- Database is up and running

Simulating the data loss

In the following steps, you will switch the log file. As a result, the CURRENT redo log will turn into ACTIVE. Then, you will delete the active group members.

39. Start SQL*Plus (if it is not already started) and login to ORADB as sysdba.

```
sqlplus / as sysdba
```

40. Run the following query to retrieve the CURRENT group number.

```
SELECT G.GROUP# FROM V$LOG G WHERE STATUS='CURRENT';
```

41. Run the following query to retrieve members of the CURRENT group. Take a note of the returned code.

```
col CODE for a150
SELECT 'host rm -f ' || MEMBER AS CODE FROM V$LOGFILE WHERE GROUP#=(SELECT G.GROUP# FROM
V$LOG G WHERE STATUS='CURRENT');
```

42. Switch the log file so that the CURRENT group becomes ACTIVE.

```
# run the following commands in the SQL*Plus session, then...
ALTER SYSTEM SWITCH LOGFILE;
SELECT G.GROUP#, G.STATUS FROM V$LOG G ORDER BY 1;
```

43. Execute the noted code to delete the active group members.

44. Verify the status of the database instance.

```
host ps -ef | grep pmon
SELECT COUNT(*) FROM SOE.ORDERS;
```

45. View the last lines in the alert log file and check if any errors are reported about the incident.

```
host tail -n 50 /u01/app/oracle/diag/rdbms/oradb/ORADB/trace/alert_ORADB.log
```

46. Run the query in the script to display the status of redo log groups and members.

```
@display_logs
```

Recovery Actions

Perform the following actions to restore the lost redo log group.

47. Reinitialize the lost redo group.

The UNARCHIVED clause is a must in this scenario.

```
ALTER DATABASE CLEAR UNARCHIVED LOGFILE GROUP <group#>;
```

48. Run the query script.

You should notice that new members are created in the group.

```
@display_logs
```

49. Switch the redo log three times and check the status of the redo group and their members after each time.

```
ALTER SYSTEM SWITCH LOGFILE;
@display_logs
```

Caution: The data that was not archived in the lost members is lost. Therefore, specifying UNARCHIVED makes backups unusable if the redo log is needed for recovery.

Recovery Scenario 18:

Recovery from Loss of the Current Redo Log Group

Scenario Target:

- In this scenario, you will recover from losing the current redo log group.

Scenario Assumptions:

- Database is using OMF
- Database is up and running

Simulating the data loss

In the following steps, you will delete the members of the CURRENT redo log.

50. Run the following query to retrieve the CURRENT group number.

```
SELECT G.GROUP# FROM V$LOG G WHERE STATUS='CURRENT';
```

51. Run the following query to generate the code to delete the members of the CURRENT group. Take a note of the returned code.

```
col CODE for a150

SELECT 'host rm -f ' || MEMBER AS CODE FROM V$LOGFILE WHERE GROUP#=(SELECT G.GROUP# FROM
V$LOG G WHERE STATUS='CURRENT');
```

52. Delete the CURRENT redo group members by pasting the code obtained from the previous step into the SQL*Plus session.

53. Verify the status of the database.

The commands may or may not terminate the instance.

```
host ps -ef | grep pmon
SELECT COUNT(*) FROM SOE.ORDERS;
CREATE TABLE SOE.TEST AS SELECT * FROM DBA_TABLES;
DROP TABLE SOE.TEST;
```

54. View the last lines in the alert log file and observe the reported errors.

```
host tail -n 50 /u01/app/oracle/diag/rdbms/oradb/ORADB/trace/alert_ORADB.log
```

Recovery Actions

Perform the following actions to restore the lost redo log group.

55. Exit from SQL*Plus, start it again with logging on as sysdba. Mount the database. If the database is still up and running, shut it down first.

```
STARTUP MOUNT
```

56. Try reinitializing the lost redo group.

Rebuilding the current group members may fail and returns the following error:

ORA-01624: log *n* needed for crash recovery of instance ORADB (thread 1)

```
ALTER DATABASE CLEAR UNARCHIVED LOGFILE GROUP <group#>;
```

57. If the previous step failed, perform the following to implement incomplete database recovery:

a. Invoke RMAN and login as target to ORADB.

b. Restore the database.

```
RESTORE DATABASE;
```

c. Exit from RMAN.

d. Invoke SQL*Plus and login as sysdba to ORADB.

e. Issue the following command

The command is issued from SQL*Plus because this format of RECOVER command is not supported by RMAN. This command recovers from the redo until you instruct it to cancel.

```
RECOVER DATABASE UNTIL CANCEL;
```

When you are prompted to respond to the command above, press [**Enter**]. When the command fails, run it again and type “**Cancel**” then press enter.

Following is the output that I received and my response is colored in red:

```
SQL> RECOVER DATABASE UNTIL CANCEL;
ORA-00279: change 1465112 generated at ** needed for thread 1
ORA-00289: suggestion :
/u01/app/oracle/fra/ORADB/ORADB/archivelog/2018_06_12/o1_*_%u_.arc
ORA-00280: change 1465112 for thread 1 is in sequence #3

Specify log: {<RET>=suggested | filename | AUTO | CANCEL}
[here I pressed ENTER]
ORA-00308: cannot open archived log
'/u01/app/oracle/fra/ORADB/ORADB/archivelog/2018_06_12/o1_*_%u_.arc'

[ I kept pressing on ENTER till I received the following error]

ORA-27037: unable to obtain file status
Linux-x86_64 Error: 2: No such file or directory
Additional information: 7


SQL> RECOVER DATABASE UNTIL CANCEL;   [I ran the same command again]
ORA-00279: change 1465112 generated at ** needed for thread 1
ORA-00289: suggestion :
```

```
/u01/app/oracle/fra/ORADB/ORADB/archivelog/**/o1_*_%u_.arc
ORA-00280: change 1465112 for thread 1 is in sequence #3

Specify log: {<RET>=suggested | filename | AUTO | CANCEL}
CANCEL      [here I typed CANCEL]
Media recovery cancelled.
```

f. Open the database using RESETLOGS option.

```
ALTER DATABASE OPEN RESETLOGS;
```

58. Run the query script.

You should notice that new members are created in the groups.

```
@display_logs
```

59. Switch the redo log three times and check the status of the redo group and their members after each time.

```
ALTER SYSTEM SWITCH LOGFILE;
@display_logs
```

Note: The data that was in the CURRENT redo log group is lost.

Clean UP

60. Restore srv1 to the snapshot “**Practice 15 Start**”. Start srv1.

61. Delete the snapshot “Practice 15 Start”.

Summary

In this practice, you implemented the following recovery scenarios:

- Recovery from loss of an inactive redo log group member
- Recovery from loss of an inactive redo log group on database startup
- Recovery from loss of an inactive redo log group while the database is in operation
- Recovery from loss of an active redo log group
- Recovery from loss of the current redo log group