

Application of Oracle 10/11g Flashback Technology to SAP Patch and Refresh Processes

Aaron Finley

Certified SAP Netweaver Technology Consultant (Oracle)

www.netweaveradm.com

Application of Oracle 10/11_g Flashback Technology to SAP Patch and Refresh Processes

| | |
|--|---|
| Flashback Technology: Flashback Database, Flash Area, and Guaranteed Restore Points..... | 3 |
| Configuring RMAN Flash Recovery Area and Size..... | 3 |
| Managing Guaranteed Restore Points | 4 |
| Using a Guaranteed Restore Point in the SAP Patch Process..... | 5 |
| Using Flashback to Speed Refresh Processes..... | 7 |
| Monitoring Techniques and Examples..... | 9 |

Flashback Technology: Flashback Database, Flash Area, and Guaranteed Restore Points

Oracle 10g introduced numerous improvements including a number of backup and recovery concepts called “flashback technologies.” Flashback allows the “rewind” of database changes. 10g and 11g support flashback query, flashback table, flashback drop, flashback database and implement a managed flash recovery area. Since SAP manages many constraints internally, any row or table level modification in Oracle is not recommended and can lead to data corruption. Therefore flashback database is of the most benefit to SAP customers and offers significant utility. Flashback database allows the point in time recovery in reverse or “rewind” of the entire database.

Flashback database supports variable point-in-time recovery based on a retention policy or fixed point-in-time recovery via Guaranteed Restore Points. This document will focus on the use of Guaranteed Restore Points and their application to refresh and patch processes. Non-guaranteed restore points are of less interest for these processes as their logs may be deleted to reclaim space or aged out due to retention policies, preventing their use in the event a recovery situation is necessitated.

Using Flashback Database and guaranteed restore points is easy. Oracle stores flashback logs in the RMAN Flash Recovery Area, which must be configured before flashback features are used. The following procedures will be outlined in this document:

1. Configuring the flash recovery area.
2. Creating and dropping guaranteed restore points.
3. Estimating restore point size.
4. Using restore points for refresh and patch processes.

Configuring RMAN Flash Recovery Area and Size

Flash area configuration for guaranteed restore points is straightforward.

Configure the RMAN Flash Recovery Area. The database must be running in ARCHIVELOG mode with LOG_ARCHIVE_DEST_n configured (do not use the deprecated LOG_ARCHIVE_DEST parameter.) Two parameters must then be set in order:

```
DB_RECOVERY_FILE_DEST_SIZE  
DB_RECOVERY_FILE_DEST
```

These parameters can be set via SQL*Plus as follows:

```
SQL> ALTER SYSTEM SET db_recovery_file_dest_size=SIZE(G) SCOPE=BOTH;  
SQL> ALTER SYSTEM SET db_recovery_file_dest='/location/oraflash' SCOPE=BOTH;
```

As an example, the following commands would configure a 180G flash recovery area in `/oracle/QAS/oraflash`:

```
SQL> ALTER SYSTEM SET db_recovery_file_dest_size=180G SCOPE=BOTH;  
SQL> ALTER SYSTEM SET db_recovery_file_dest='/oracle/QAS/oraflash' SCOPE=BOTH;
```

Managing Guaranteed Restore Points

If flashback is not enabled, the first guaranteed restore point must be created in `MOUNT` mode. If an existing guaranteed restore point exists, multiple restore points can be added while the database is `OPEN`.

To create the first guaranteed restore point, first shutdown the database and start in `MOUNT` mode:

```
SQL> SHUTDOWN IMMEDIATE;  
SQL> STARTUP MOUNT;
```

The following command creates a new guaranteed restore point called `RESTORE_POINT`:

```
SQL> CREATE RESTORE POINT RESTORE_POINT GUARANTEE FLASHBACK DATABASE;
```

To delete the restore point:

```
SQL> DROP RESTORE POINT RESTORE_POINT;
```

When a restore point is deleted all flashback logs associated with the restore point are automatically deleted by Oracle.

In contrast to flashback database, one cannot flashback to a point-in-time between the guaranteed restore point creation time and the current time. For this purpose, Oracle distinguishes between the generation of flashback logs for standard flashback purposes and those for guaranteed restore points. If flashback logging is not enabled and a guaranteed restore point is created, the flashback mode is “`RESTORE POINT ONLY`.”

To determine the current flashback mode, query `V$DATABASE`:

```
SQL> SELECT FLASHBACK_ON FROM V$DATABASE;  
FLASHBACK_ON
```

```
-----  
RESTORE POINT ONLY
```

To obtain more information about restore points, query V\$RESTORE_POINT:

```
SQL> SELECT NAME, TO_CHAR(TIME, 'yyyy:mm:dd:hh24:mi:ss'),  
STORAGE_SIZE/1024/1024/1024 AS STORAGE_GB FROM V$RESTORE_POINT;  
  
NAME          TIME          STORAGE_GB  
-----  
RESTORE_POINT 2010:12:19:16:13:04  6.38881683
```

Using a Guaranteed Restore Point in the SAP Patch Process

In this example a guaranteed restore point is created during the pre-steps process of applying SAP patches. As restore points are not substitutes for backups, as a best practice a hot or cold backup should be taken before the GRP is created.

Steps to perform before applying patches:

- a. Full database backup
- b. Enable flashback and create guaranteed restore point
- c. SAP instance backup
- d. /usr/sap/trans backup (optional)

Make sure that archiving is enabled and that `log_archive_dest_1` is maintained rather than `log_archive_dest`.

Configure `DB_RECOVERY_FILE_DEST_SIZE` and `DB_RECOVERY_FILE_DEST`:

```
SQL> ALTER SYSTEM SET db_recovery_file_dest_size='SIZE(G)' SCOPE=spfile;  
SQL> ALTER SYSTEM SET db_recovery_file_dest='/location/oraflash' SCOPE=spfile;
```

If this is the first time a restore point is created, flashback does not need to be ON. There are two types of flash logging: flashback and restore point only. For this exercise, only “restore point only” is required so a restore point is created without first enabling flashback. When a restore point is created without flashback enabled, flash is enabled but in mode “restore point only.” To create a restore point without flashback on database must be in MOUNT mode.

```
SQL> SHUTDOWN IMMEDIATE;  
SQL> STARTUP MOUNT;
```

Create the restore point and open the database:

```
SQL> CREATE RESTORE POINT BEFORE_PATCHES GUARANTEE FLASHBACK DATABASE;  
SQL> ALTER DATABASE OPEN;
```

Check flashback mode:

```
SQL> SELECT FLASHBACK_ON FROM V$DATABASE;  
  
FLASHBACK_ON  
-----  
RESTORE POINT ONLY
```

Notice the mode is "RESTORE POINT ONLY."

Back up the SAP instance using tar:

| | | |
|-----|--------------------------|--------------------------|
| tar | /usr/sap/SID | Local Application Files |
| tar | /sapmnt/SID | Global Application Files |
| tar | /export/home/(sidadm)/.* | Local Environment Files |

Example:

```
/usr/sfw/bin/gtar cz /storage/usr_sap_SID.tar.gz /usr/sap/SID
```

Remember to use GNU TAR (typical Solaris: `/usr/sfw/bin/gtar`) as Solaris TAR has a 2GB limit and will skip over files larger than 2GB without warning.

Now perform patching.

If an error occurs when patching, perform the following procedure to flashback the database to the GRP:

```
SQL> SHUTDOWN IMMEDIATE;  
SQL> STARTUP MOUNT;  
SQL> FLASHBACK DATABASE TO RESTORE POINT BEFORE_PATCHES;  
SQL> ALTER DATABASE OPEN RESETLOGS;
```

The syntax to drop the restore point:

```
SQL> DROP RESTORE POINT BEFORE_PATCHES;
```

Since flashback was in "RESTORE POINT ONLY" mode, flashback will be disabled once the single restore point is dropped.

Ensure that that flashback mode has turned off:

```
SQL> SELECT FLASHBACK_ON FROM V$DATABASE;

FLASHBACK_ON
-----
NO
```

Flashback is not a substitute for physical database backups. Flashback does not prevent or provide recover for media errors. You should always ensure good backups regardless of whether flashback is used. Neither BRBACKUP nor RMAN currently have functionality to backup flashback logs, so if the flashback area suffers a media error during patching and patching fails, you will need to recover from a backup.

Using Flashback to Speed Refresh Processes

Many SAP systems are very large but produce proportionally little amounts of redo per week. SAP training systems are often large systems with many clients that are refreshed weekly. A training system may be several hundred gigabytes to several terabytes, but the amount of redo generated in one week may well be under a hundred gigabytes. Many SAP customers go to tape for their weekly refresh process due to the size of the system. Flashback is an excellent candidate to replace going to tape.

To repeat the warning above, remember that flashback is not a substitute for physical backups. In the case of a training system, the “Golden” backup and subsequent changes should still be written to tape.

Flashback is not a substitute for having physical (tape or disk) database backups. Flashback will not allow recovery in the event of a media failure.

Example: At one SAP customer, the training system was approximately 1TB in size. The recovery time from tape was approximately eight to ten hours. Redo generation averaged around sixty gigabytes per week. Recovery with flashback using a guaranteed restore point takes approximately six minutes.

To properly size the flashback area, begin with calculating the amount of redo generated during the week. The following SQL statement will return the amount of redo in gigabytes generated in the last seven days:

```
SQL> SELECT ROUND((COUNT(RECID)*VL.BYTES)/1024/1024/1024,2) AS REDO_GB FROM
V$ARCHIVED_LOG, (SELECT MAX(BYTES) AS BYTES FROM V$LOG) VL WHERE
FIRST_TIME>sysdate-7 GROUP BY BYTES;

REDO_GB
-----
```

The above query could be extended to calculate redo for groups of working days. As a general rule, the calculation using SYSDATE-14 or fourteen days (if the last two weeks have contained two working weeks) should establish a safe margin for calculations.

In a real world example, a customer has refreshed their SAP system from flashback on Sunday. A restore point called WEEKLY_REFRESH was created. Today is Tuesday. SYSDATE-3 shows that approximately 18GB of redo have been generated since refresh.

The following command is used to obtain the storage size needed for the restore point:

```
SQL> SELECT STORAGE_SIZE/1024/1024/1024 AS STORAGE_GB FROM V$RESTORE_POINT WHERE
NAME='WEEKLY_REFRESH';

STORAGE_GB
-----
6.40350342
```

The following command is used to obtain the amount of redo generated since the creation of the restore point:

```
SQL> SELECT ROUND((COUNT(RECID)*VL.BYTES)/1024/1024/1024,2) AS REDO_GB FROM
V$ARCHIVED_LOG, (SELECT MAX(BYTES) AS BYTES FROM V$LOG) VL WHERE
FIRST_TIME>(SELECT TIME FROM V$RESTORE_POINT WHERE NAME='WEEKLY_REFRESH') GROUP BY
BYTES;

REDO_GB
-----
13.18
```

In this example, 13.18G of redo is generated for 6.40G of flashback. It is apparent that flashback logs take up less space than redo. In this case, redo consumes 2.05X more space. The customer has a smaller system using flashback. The results are as follows:

| SYSTEM # | REDO (GB) | FLASHBACK (GB) | MULTIPLIER |
|----------|-----------|----------------|------------|
| 1 | 13.18 | 6.40 | 2.05 |
| 2 | .54 | .28 | 1.92 |

We can conclude that at minimum, ½ the size of the expected redo generation should be available to flashback. As a good practice, double the size of the minimum requirement brings the size 1:1 with redo generation. Therefore, estimate space required in the flash recovery area for flashback logs by using the size of generated redo.

BR*Tools does not support the flash recovery area as an archive destination. Therefore `log_archive_dest_1` should be maintained as a location pointing outside of the flash recovery area. Using our case of guaranteed restore points, the only files in the flash recovery area will therefore be flashback logs supporting the restore point(s).

In order to use flashback for a weekly refresh process:

- 1) Estimate the size required for flashback logs via the methods above and determine a suitable location for flashback logs. The SAP standard is `/oracle/SID/oraf1ash`.
- 2) Configure the flash recovery area:

```
SQL> ALTER SYSTEM SET db_recovery_file_dest_size='SIZE(G)' SCOPE=spfile;
SQL> ALTER SYSTEM SET db_recovery_file_dest='/location/oraf1ash' SCOPE=spfile;
```

- 3) Restore the database from tape/disk and then create restore point:

(restore database to golden from tape/disk/etc.)

```
SQL> SHUTDOWN IMMEDIATE;
SQL> STARTUP MOUNT;
SQL> CREATE RESTORE POINT WEEKLY_REFRESH;
SQL> ALTER DATABASE OPEN;
```

- 4) At the end of the week (or whenever the system is traditionally refreshed), recover the restore point:

```
SQL> SHUTDOWN IMMEDIATE;
SQL> STARTUP MOUNT;
SQL> FLASHBACK DATABASE TO RESTORE POINT WEEKLY_REFRESH;
SQL> ALTER DATABASE OPEN RESETLOGS;
SQL> DROP RESTORE POINT WEEKLY_REFRESH;
```

- 5) Perform any changes to the golden which should persist.
- 6) Repeat starting at Step #3 to create a new restore point.

Monitoring Techniques and Examples

The following SQL*Plus code determines the oldest SCN and displays the oldest time to flashback to in hours.

```
SQL> COLUMN CURRENT_SCN HEADING 'CURRENT SCN' FORMAT 999999999999
```

```

SQL> COLUMN OLDEST_FLASHBACK_SCN HEADING 'OLDEST SCN' FORMAT 999999999999
SQL> COLUMN SCNS HEADING 'DIFFERENCE'          FORMAT 999999999999

SQL> SELECT B.CURRENT_SCN, A.OLDEST_FLASHBACK_SCN, (B.CURRENT_SCN-A.
OLDEST_FLASHBACK_SCN) SCNS, TO_CHAR(A.OLDEST_FLASHBACK_TIME, 'yyyy:mm:ddhh24:mi:ss')
OLDEST_TIME, TRUNC((SYSDATE-A.OLDEST_FLASHBACK_TIME)*24) HOURS

FROM

V$FLASHBACK_DATABASE_LOG A, V$DATABASE B;

```

| CURRENT SCN | OLDEST SCN | DIFFERENCE | OLDEST_TIME | HOURS |
|-------------|-------------|------------|-------------|-------------|
| 13326809144 | 13326717535 | 91609 | 2010:11:10 | 03:23:52 15 |

To check free space in the flash/fast recovery area, use the following SQL*Plus code:

```

SQL> COLUMN SL HEADING 'TOTAL (G)'
SQL> COLUMN SU HEADING 'USED (G)'
SQL> COLUMN SR HEADING 'RECLAIMABLE (G)'

SQL> SELECT SPACE_LIMIT/1073741824 SL, ROUND(SPACE_USED/1073741824,2)
SU, ROUND(SPACE_RECLAIMABLE/1073741824,2) SR, NUMBER_OF_FILES

FROM

V$RECOVERY_FILE_DEST;

```

| TOTAL (G) | USED (G) | RECLAIMABLE (G) | NUMBER_OF_FILES |
|-----------|----------|-----------------|-----------------|
| 475 | .37 | 0 | 18 |

Determine percent used by redo and flashback logs in the flash recovery area:

```

SQL> SELECT FILE_TYPE, PERCENT_SPACE_USED, PERCENT_SPACE_RECLAIMABLE

FROM

V$FLASH_RECOVERY_AREA_USAGE

WHERE

FILE_TYPE IN ('ARCHIVELOG', 'FLASHBACKLOG');

```

| FILE_TYPE | PERCENT_SPACE_USED | PERCENT_SPACE_RECLAIMABLE |
|--------------|--------------------|---------------------------|
| ARCHIVELOG | 0 | 0 |
| FLASHBACKLOG | .08 | |