

Flashback Data Archives re-checked or reject?

Beat Ramseier
Consultant, IMS-ZH
22.8.2012



The promising feature “Flashback Data Archives” was introduced with Oracle 11g Release 1. Various limitations prevented practical usage in real-world environments. Because of high customer interest in Oracle integrated data archiving features we want to re-check and judge this feature within Oracle 11g Release 2 (specifically 11.2.0.3). Oracle delivered various improvements - do they hold up to the promise?

1 Introduction

Before we dive into technical details and new features of Flashback Data Archives let's explore why we all want, or probably have to, track and retain historical data.

First and foremost various regulatory bodies mandate traceability of certain data, FDA, SOX, PMDA, Directive 2006/24/EC, just to name a few. Non-compliance may lead to fines, product recalls, public exposure (reputational damage) which all impact business severely.

Besides regulatory requirements, keeping track of the chronological data changes adds value in other areas as well, e.g. data analysis, data error correction, etc.

1.1 What is what

Before July 2012

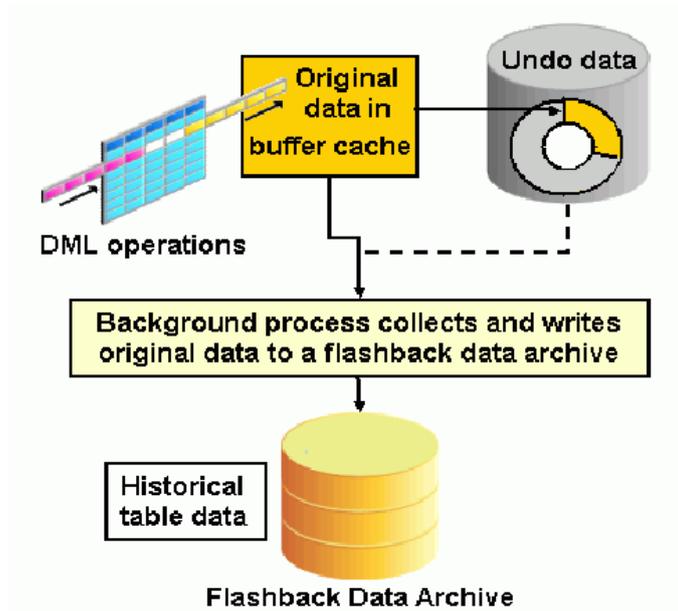
- Oracle Total Recall is an extra cost option which can be licensed with Oracle Enterprise Edition
- Flashback Data Archive is the (only) feature within the Oracle Total Recall option

Starting July 2012

- Oracle Total Recall is no longer a separate option (does not exist anymore)
- Flashback Data Archive feature is contained within the Oracle Advanced Compression option

2 Technical overview/insights into FDAs

As most of Oracle's Flashback technologies Flashback Data Archives rely on undo data. Whenever undo data is generated for a FDA enabled table Oracle flags it so it will be processed by the FBDA background process. As the FBDA processes the undo data asynchronously the performance of the transaction is not affected (e.g. compared to self-made archiving solution with triggers and history tables). FDA flagged undo data is only freed when all depending data has been processed by FBDA, meaning relevant data is stored in the Flashback Data Archive. This might even over-span the undo retention period.



Quelle: Oracle Corporation

2.1 Nitty-gritties of FBDA

The FBDA background process is a lazy beast. It wakes up every 5 minutes by default (*1), checks Tablespace quotas every hour, creates history tables only when it has to and rests otherwise if not called to work. The process adjusts the wakeup interval dynamically based on the workload. Once called to work it shows its real face – FBDA spawns up to 10 slave processes (*2) to perform the work, all of them able to issue parallel DML on the internal processing and history tables.

While stated above that there is no direct influence on transaction performance, heavy activity of the FBDA and its slaves can generate noticeable load on the overall system.

Both parallel DML and slave processes were introduced with Oracle 11g R2.

*1 Contradicting observations to MOS Note 1302393.1 were made. The note says that FBDA wakes up every 6 seconds as of Oracle 11.2.0.2, but a trace on 11.2.0.3 showed:

WAIT #140555800502008: nam= 'fbar timer' ela= 300004936 sleep time=300 failed=0 p3=0 obj#=-1 tim= 1334308846055308

*2 The slaves are *Wnnn* processes, current documentation says *Wnnn* are slaves of SMCO

2.2 FDA tables under the hood

Having mentioned internal processing tables and history tables it's time to see where the data goes. A Flashback Data Archive consists of three tables:

Table	Purpose
SYS_FBA_HIST_ <Object ID >	Contains the historical attribute values (delta)
SYS_FBA_DDL_COLMAP_ <Object ID >	Mapping of historical attributes (structure)
SYS_FBA_TRC_ <Object ID >	Mapping of latest transactions to the corresponding UNDO data

- These tables are created in the same schema as the FDA enabled table
- <Object ID > is the object ID of the FDA enabled table



- SYS_FBA_HIST_% tables are compressed for OLTP and partitioned on ENDSCN

Additionally, there are some global temporary tables created in the SYS schema that are used for internal (undo) data processing.

Querying the archive tables is not supported by Oracle and historical data should always be accessed using Flashback Queries (AS OF / VERSIONS).

2.3 Flashback queries

You may be familiar with flashback queries already – and it’s all the same for querying historical data from an FDA enabled table from a SQL point of view.

Behind the scenes, the Oracle optimizer knows how to translate the flashback query to enable access to the history tables.

e.g. execution plan of a simple AS OF query: `SELECT * FROM t1 AS OF SCN <scn>`

Plan hash value: 3649640700

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time	Pstart	Pstop
0	SELECT STATEMENT		7	14105	25 (8)	00:00:01		
1	VIEW		7	14105	25 (8)	00:00:01		
2	UNION-ALL							
* 3	FILTER							
4	PARTITION RANGE SINGLE		1	31	14 (0)	00:00:01	KEY	1
* 5	TABLE ACCESS FULL	SYS_FBA_HIST_67017	1	31	14 (0)	00:00:01	KEY	1
* 6	FILTER							
7	MERGE JOIN OUTER		6	276	11 (19)	00:00:01		
8	SORT JOIN		5	95	7 (15)	00:00:01		
* 9	TABLE ACCESS FULL	T1	5	95	6 (0)	00:00:01		
* 10	SORT JOIN		100	2700	4 (25)	00:00:01		
* 11	TABLE ACCESS FULL	SYS_FBA_ICRV_67017	100	2700	3 (0)	00:00:01		

The optimizer generates the execution plan as with any other SQL statement; factors like data volume, selectivity, indexes, etc. all come into play. Consequently, it’s important to ensure object statistics are accurate for the history tables. According to the data access patterns of the flashback queries it is also allowed to create indexes on SYS_FBA_HIST_% tables.

Flashback VERSIONS query feature was introduced with Oracle 11.1.0.7 and is not available in earlier versions of 11g Release 1.

3 FDA features 11gR1 vs 11gR2

Why so harsh on Oracle 11g Release 1 to label the feature unfit for production use? you may ask. Well, in any real-world database schema there will be structural changes at some point. In some environment more frequent than in others. This is exactly where **Oracle 11g Release 1** is very limited. Any of the following DDL statements will cause an error:

- TRUNCATE, RENAME, DROP table
- ALTER TABLE
 - o to modify, rename, drop a column
 - o to perform (sub) partition operations



- to convert LONG to LOB
- that include UPGRADE TABLE clause (with or without INCLUDING DATA)

There is virtually no way to perform any of the above mentioned actions **without losing the entire archive!**

Starting with **Oracle 11g Release 2** most of these limitations have been lifted, the residing restrictions being:

- DROP table (*1)
- ALTER TABLE
 - operations that move or exchange partitions
 - that perform sub-partition operations

*1 we consider this to be the correct behavior as a FDA enabled table should not be dropped before the archiving is disabled

While these are the officially documented limitations we found some more during our testing:

- split partitions
- enabling a disabled constraint

3.1 Associate/Disassociate

Oracle seems to really acknowledge the issues and introduced another feature: Associate/Disassociate.

With the new built-in PL/SQL package `DBMS_FLASHBACK_ARCHIVE` one can temporarily disassociate an archive from its base table, perform the necessary operation and re-associate the archive again.

While this solves the problem of unsupported DDL operations it poses some concerns regarding the integrity of the archive. During the disassociated state DML can be performed on the history table from privileged users.



Example:

```
SQL> EXEC DBMS_FLASHBACK_ARCHIVE.DISASSOCIATE_FBA (
        OWNER_NAME => 'OTRTEST',
        TABLE_NAME => 'T1');

SQL> ALTER TABLE otrtest.t1 SPLIT PARTITION partx
        VALUES(1) INTO (PARTITION part1, PARTITION partx);

SQL> EXEC DBMS_FLASHBACK_ARCHIVE.REASSOCIATE_FBA (
        OWNER_NAME => 'OTRTEST',
        TABLE_NAME => 'T1');
```

It is advisable to just perform actions that are not supported otherwise in disassociated state. If structural changes are made it is imperative to ensure the consistency between the base table and the history table.

4 Known-issues and other limitations

As with many new software there are certain bugs. Luckily, Oracle already fixed the most critical ones. Probably most prominent is bug 10235250, getting ORA-00972 while renaming table column with a long column name.

Other more subtle bugs were fixed in 11.2.0.3, e.g. bug 8349119.8 where multiple records for the same row are returned while querying the historical data.

For a full list of bugs fixed in the 11gR2 patch sets please see MOS notes 1179583.1 and 1348303.1.

4.1 Timestamp < > SCN mapping

Irrespective of Flashback Data Archives the database stores the timestamp < > SCN mappings in the SMON_SCN_TIME table. Every five minute a new row containing ~ 100 mappings (3 second accuracy) is generated and stored in that table. Usually, the mappings are only kept for the duration of the auto-tuned undo retention period. But, to account for the greatest retention period of all FDAs in the database Oracle has to keep all this information.

The space needed in the SYSAUX Tablespace needs to be accounted for, e.g.:
10 year FDA retention would produce 1'051'200 entries in SMON_SCN_TIME table.
Multiplied by an average row-length of about 1KB, this would require about 1GB of space.

4.2 Export/import of FDAs

Neither Data Pump nor the old exp/imp utilities support Flashback Data Archives. While Data Pump exports of FDA enabled tables work fine it ignores the history tables entirely and though they cannot be imported again. With the exp/imp utilities the history tables can be exported and imported as any other table. The problem is that the relationship between the base table and the history table is lost as the definition of flashback data archives is not known by the utilities. Therefore Oracle Support provides two PL/SQL packages that assist in transferring history data from one database to another:

- DBMS_FDA_MAPPINGS
- DBMS_FDA_IMPORT



Example:

1. Create the table with same structure as source in the target DB and enable archiving (ensure that retention covers history to import)
2. Create time < > SCN mappings (*1) on target DB:
`SQL> EXEC DBMS_FDA_MAPPINGS.EXTEND_MAPPINGS;`
3. Create a staging history table on target DB:
`SQL> EXEC DBMS_FDA_IMPORT.CREATE_TEMP_HISTORY_TABLE("OTRTEST", "T1");`
4. Transfer history data from source archive to staging history table (*2)
5. Import history data
`SQL> EXEC DBMS_FDA_IMPORT.IMPORT_HISTORY("OTRTEST", "T1");`

*1 The EXTEND_MAPPING procedure generates 2 entries per day back to 1 January 1988 in SMON_SCN_TIME

*2 e.g. use SQL*Plus to spool the data from the source archive to a flat file by querying the history table directly and load the data into the staging history table with SQL*Loader

The procedure is quite cumbersome with lots of manual steps. Also, the timestamp < > SCN mapping is very sparse and might not be accurate enough in all cases.

5 Implementation recommendations

To let you save some trouble implementing Flashback Data Archives we'd like to give some word of advice.

5.1 Space management

To ease the space management for FDA we recommend following these rules:

- Create an archive for different retention periods
 - o to optimally use space and take advantage of automatic purging when retention expires
- Group tables with same retention policy (share FDAs)
 - o to reduce maintenance effort of multiple FDAs with same characteristics
- Dedicate Tablespaces to one archive and do not set quotas
 - o to reduce maintenance/monitoring effort, quotas are only checked every hour by FBDA

5.2 Security

In larger environments it might even be desirable to separate the duties between a table owner and a FDA owner on schema level. E.g. a compliance officer is in charge of managing the historical data and table owners should not be allowed to disable/disassociate archives.

Following scenario achieves this separation of duty:

- DBA grants FLASHBACK ARCHIVE ADMINISTER to FDA owner
- FDA owner creates archives and grants FLASHBACK ARCHIVE on archives to base table owner
- Tablespace quota (unlimited) for FDA must be granted to both, FDA and base table owner

Unfortunately, there seems to be a bug that only occurs when the setup is like that. Instead of raising ORA-55617 or ORA-55623 the session performing the DML on the base table is hanging



while writing ORA-01688 errors to the alert.log file. Therefore it is important to closely monitor space usage of the FDAs.

If FDA owner and base table owner is the same schema it works as expected.

ORA-55617: "Flashback Archive \"%s\" runs out of space and tracking on \"%s\" is suspended"

ORA-55623: "Flashback Archive \"%s\" is blocking and tracking on all tables is suspended"

ORA-01688: "unable to extend table string.string partition string by string in tablespace string"

5.3 Performance

Unfortunately, there is no much headroom for performance optimization within the data processing for FDAs as Oracle declares them to be self-managing, e.g. there is no way to influence the number of FBDA slave processes.

As mentioned earlier, what we can influence is Flashback Query performance:

- Gather object statistics for history tables
- Create indexes (non-unique) on history according to Flashback Query access patterns

6 Conclusion

With Oracle 11g Release 2 Oracle has lifted the major limitations that have prevented practical usage of Flashback Data Archives in the previous release. Lots of bug fixing and other improvements (e.g. parallel processing) matured the feature so it can be considered for production use in good conscience.

Naturally, there is still room for improvement. Here's what we'd expect from FDAs in future Oracle releases:

- Full support for schema evolution (all possible DDLs)
- Data Pump support for FDA enabled tables as well as historical data itself
- User-context tracking (who performed an operation)
- Named restore-points for Flashback Queries

One final concern to share is that while in 11g Release 1 FDAs truly were tamper proof, the introduction of associate/disassociate in 11g Release 2 leaves some bitter-sweet aftertaste with regards to data integrity and traceability.

Beat Ramseier

Trivadis AG

Europa-Strasse 5 Tel: +41-44-808 70 20

CH-8152 Glattbrugg Fax: +41-44-808 70 21

Internet: www.trivadis.com Mail: info@trivadis.com