

## The Flashback Data Archive Performance Issue

Another 11g new feature is Flashback Data Archive, part of the Oracle Total Recall option. It alleviates the need to use database triggers by transparently archiving historic data to a Flashback Archive tablespace.

A commit delay currently exists in version 11.1.0.7. In the test case below when issuing an UPDATE against a table having 2268 rows, a delay of > 3 seconds occurs before the SQL prompt is returned to the user following a commit.

### Test Case:

Create a FDA tablespace and a FDA

```
SQL> create flashback archive fda_13_mth
      tablespace fda
      retention 13 month;
```

#### With FDA on the table

```
SQL> alter table fda_test flashback archive fda_13_mth;
```

Table altered.

Elapsed: 00:00:00.02

```
SQL> update fda_test set id = id;
```

2268 rows updated.

Elapsed: 00:00:00.08

```
SQL> commit;
```

Commit complete.

Elapsed: 00:00:03.84

#### No FDA on the table

```
SQL> alter table fda_test no flashback archive;
```

Table altered.

```
SQL> set timing on
```

```
SQL> update fda_test set id = id;
```

2268 rows updated.

Elapsed: 00:00:00.07

```
SQL> commit;
```

Commit complete.

Elapsed: 00:00:00.00

## Analysis:

Performing a SQL trace with TKPROF output we see 9 internal SQL statements in SQL trace file for non-FDA vs 50 for FDA. Resulting in 86% more parses and executions overall.

### With FDA

```
1 session in tracefile.
19 user SQL statements in trace file.
50 internal SQL statements in trace file.
69 SQL statements in trace file.
54 unique SQL statements in trace file.
1005 lines in trace file.
18 elapsed seconds in trace file.
```

\*\*\*\*\*

```
SQL ID: dmucg5wr9xxm0
Plan Hash: 761720216
Update fda_test set id = id
```

call	count	cpu	elapsed	disk	query	current	rows
Parse	1	0.00	0.02	0	0	0	0
Execute	1	0.05	0.08	0	13	2353	2282
Fetch	0	0.00	0.00	0	0	0	0
total	2	0.06	0.10	0	13	2353	2282

```
Misses in library cache during parse: 1
Optimizer mode: ALL_ROWS
Parsing user id: SYS
```

Rows	Row Source Operation
0	UPDATE FDA_TEST (cr=13 pr=0 pw=0 time=0 us)
2282	TABLE ACCESS FULL FDA_TEST (cr=13 pr=0 pw=0 time=0 us cost=6 size=9128 card=2282)

\*\*\*\*\*

OVERALL TOTALS FOR ALL NON-RECURSIVE STATEMENTS

call	count	cpu	elapsed	disk	query	current	rows
Parse	2	0.00	0.02	0	0	0	0
Execute	3	0.10	0.21	0	13	2407	2282
Fetch	0	0.00	0.00	0	0	0	0
total	5	0.11	0.23	0	13	2407	2282

```
Misses in library cache during parse: 1
Misses in library cache during execute: 1
```

OVERALL TOTALS FOR ALL RECURSIVE STATEMENTS

call	count	cpu	elapsed	disk	query	current	rows
Parse	64	0.08	0.47	0	0	0	0
Execute	66	1.65	1.71	43	493012	25335	34236
Fetch	56	0.00	0.00	0	240	0	44
total	186	1.75	2.18	43	493252	25335	34280

Misses in library cache during parse: 46  
 Misses in library cache during execute: 11

19 user SQL statements in session.  
 50 internal SQL statements in session.  
 69 SQL statements in session.

\*\*\*\*\*

No FDA

1 session in tracefile.  
 3 user SQL statements in trace file.  
 9 internal SQL statements in trace file.  
 12 SQL statements in trace file.  
 12 unique SQL statements in trace file.  
 183 lines in trace file.  
 10 elapsed seconds in trace file.

\*\*\*\*\*

SQL ID: dmucg5wr9xxm0  
 Plan Hash: 761720216  
 Update fda\_test set id = id

call	count	cpu	elapsed	disk	query	current	rows
Parse	1	0.00	0.00	0	0	0	0
Execute	1	0.05	0.06	0	13	2349	2282
Fetch	0	0.00	0.00	0	0	0	0
total	2	0.06	0.06	0	13	2349	2282

Misses in library cache during parse: 1  
 Optimizer mode: ALL\_ROWS  
 Parsing user id: SYS

Rows	Row Source Operation
0	UPDATE FDA_TEST (cr=13 pr=0 pw=0 time=0 us)
2282	TABLE ACCESS FULL FDA_TEST (cr=13 pr=0 pw=0 time=0 us cost=6 size=9128 card=2282)

\*\*\*\*\*

OVERALL TOTALS FOR ALL NON-RECURSIVE STATEMENTS

call	count	cpu	elapsed	disk	query	current	rows
Parse	2	0.00	0.00	0	0	0	0
Execute	3	0.05	0.07	0	13	2350	2282
Fetch	0	0.00	0.00	0	0	0	0
total	5	0.06	0.07	0	13	2350	2282

Misses in library cache during parse: 1  
 Misses in library cache during execute: 1

OVERALL TOTALS FOR ALL RECURSIVE STATEMENTS

call	count	cpu	elapsed	disk	query	current	rows
Parse	9	0.00	0.00	0	0	0	0
Execute	9	0.00	0.00	0	0	0	0
Fetch	21	0.00	0.00	0	25	0	13
total	39	0.00	0.00	0	25	0	13

Misses in library cache during parse: 0

3 user SQL statements in session.  
 9 internal SQL statements in session.

12 SQL statements in session.  
\*\*\*\*\*

Using Triggers on the table that store an after image of all the columns in another audit version of the table takes 00.00.00.48 seconds for the same 2282 row update. The commit is instantaneous.

## Conclusion

Oracle documentation states; “Regular user transactions will see negligible impact. Flashback Data Archive employs a lightweight mechanism to mark DML operations on tracked tables for archiving. The actual history generation and archiving is done asynchronously through a background process”.

Unfortunately no solution currently exists for Oracle 11.1.0.7, apart from using good old-fashioned Triggers, but this seems to be a backward step. In my research into this performance issue, it is the “actual history generation and archiving through a background process” that consumes database resources and causes commit delay. In version 11.1.0.6 the SQL prompt is returned immediately to the user following a commit, despite the background process overhead. This can be an issue for multiple large transactions as they may cause a backlog of high resource consuming processes that could in severe cases crash the database instance.

*N.B. This issue has been fixed in Oracle 11gR2 (11.2.0.2). A patch for bug 8226666 is now available on 11.1.0.7.1*