INTERNALS OF TRANSACTION RECOVERY INSIDE OF ORACLE BLOCK AND REDO LOGS WITH BLOCK DUMP

Sandip Patel, Abbott

OVERVIEW:

Understanding the transaction handling in Oracle is very important for Database Administrators. A few of the important concepts include: 1) How Oracle performs inserts, update and delete, 2) How Oracle maintains the read consistency for each of the transaction throughout the database, 3) How Oracle identifies the committed and uncommitted transaction and provides the read consistency to all the users, and 4) How oracle handles rollforward and rollback during the crash recovery of an instance. It is very interesting and important to understand this internals of Oracle's transaction processing method. Database Administrators benefits from understanding transaction processing in order to solve problems and work with performance aspects of databases. This paper will describe the internals of Oracle block, how it handles the transactions inside the Data Block and also how it uses the Undo Block to save undo information of the transaction. It will also explain how to track transaction between Data Block and Undo Block. The changes happen inside the block with update or delete followed by commit or rollback. Changes include the SCN (System Change Number) associated with each transaction and the transaction flags associated with each transaction to identify the state of the transaction like committed or uncommitted transaction. This paper also explains the internals of the transaction recovery on the instance crash. How oracle handles rollforward for committed data and rollback for uncommitted data. How oracle is using redo logs to perform rollforward and rollback as a part of instance crash recovery. Overall, it will provide a very good and clear understanding about the internals of the transactions moving inside the oracle block and internals of rollforward and rollback. This paper also provides high-level internal space management inside Oracle block.

OBJECTIVES:

1. To understand and watch the transaction inside the Oracle block by dumping the block into text file.
2. Dump the Oracle block and read the data inside DATA and UNDO blocks.
3. How oracle handles the transaction with read consistency by using the transaction table and SCN.
4. To understand the internals for rollforward and rollback during the instance crash recovery.
5. Read the redo and undo transaction code inside the redo log files by using LogMiner utility

DEFINATIONS:

Oracle Block:

- Oracle block is the smallest unit of storage that the Oracle server can allocate, read, or write.
- At the finest level of granularity, the data in Oracle database is stored in the data blocks.
- The standard data block size for an Oracle database is specified by the DB_BLOCK_SIZE initialization parameter when the database is created.

Undo Block:

- Undo block is the same as data block, but it is a part of Undo data files, which stores the undo information for the transaction.

Read Consistency:

- One of the key functions of a database is to ensure that multiple users can read and write to the database without overwriting each other's changes inadvertently or reading inconsistent data due to in-progress changes.

- Data consistency means that each user sees a consistent view of the data, including visible changes made by the user's own transactions and transactions of other users.
- Uncommitted change in data is only viewable to the user(session) by whom the change is made, and other users are seeing the unchanged data until the user commits the transaction.
- Undo tablespace allow the database to maintain read consistency among multiple transactions. If a SELECT starts on a table while a transaction is modifying the same table, the old value of the data will be read from a undo tablespace even if the changed data is actually read by the SELECT after the transaction has been committed. This is what is called *read consistency*.

Rollforward:

During the instance crash recovery oracle re-apply all the transactions from redo log files after the last check point to retrieve the unwritten committed data in the data files on the disk, which may includes committed and uncommitted transaction as well. The uncommitted data gets rollback later with the rollback.

Rollback (During instance crash recovery):

During the instance crash recovery Oracle first performs rollforward as above, which contains committed and uncommitted both data. To rollback uncommitted transaction oracle performs rollback again after rollforward. Oracle uses the undo data generated during the rollforward to perform the rollback. Oracle performs this rollback on as needed basis.

DETAILED STEPS:

In order to understand the internal transaction handling mechanism of Oracle, perform the following steps:

1. Create tablespace and User.

```
CONN SYS/ORACLE AS SYSDBA;

CREATE TABLESPACE DEMO DATAFILE 'C:\ORACLE\ORA10G\ORCL\ORADATA\demo01.DBF'
SIZE   10M
AUTOEXTEND ON MAXSIZE 20M
EXTENT MANAGEMENT LOCAL UNIFORM SIZE 64K;

COLUMN NAME FORMAT A60
SELECT NAME,FILE# FROM V$DATAFILE;

CREATE USER DEMO
  IDENTIFIED BY DEMO
  DEFAULT TABLESPACE DEMO
  TEMPORARY TABLESPACE TEMP
  PROFILE DEFAULT
  ACCOUNT UNLOCK;
  GRANT DBA TO DEMO;
  GRANT CONNECT TO DEMO;
  GRANT RESOURCE TO DEMO;
  ALTER USER DEMO DEFAULT ROLE CONNECT, RESOURCE;
  GRANT SELECT ANY TABLE TO DEMO;
  GRANT UNLIMITED TABLESPACE TO DEMO;
  GRANT SELECT ANY DICTIONARY TO DEMO;
  GRANT SELECT ANY TRANSACTION TO DEMO;
  ALTER USER DEMO QUOTA UNLIMITED ON TEMP;
  ALTER USER DEMO QUOTA UNLIMITED ON DEMO;
  GRANT UNLIMITED TABLESPACE TO HR;
  GRANT SELECT ANY DICTIONARY TO HR;
  GRANT SELECT ANY TRANSACTION TO HR;
  ALTER USER HR QUOTA UNLIMITED ON TEMP;
  ALTER USER HR QUOTA UNLIMITED ON DEMO;
```

2. Create table and insert the initial row.

```
CONN DEMO/DEMO

CREATE TABLE EMP_DEMO (
```

```
EMPNO NUMBER(10),
EMPNAME VARCHAR2(10));

INSERT INTO EMP_DEMO VALUES (1,'DAN');

COMMIT;
```

3. Identify the Oracle Block containing above inserted row data.

```
CONN SYS/ORACLE AS SYSDBA
SELECT      FILE_ID, BLOCK_ID, BLOCKS
FROM  DBA_EXTENTS
WHERE       SEGMENT_NAME = 'EMP_DEMO'
AND   OWNER = 'DEMO';

   FILE_ID   BLOCK_ID     BLOCKS
---------- ---------- ----------
         6          9          8
```

FILE_ID = Datafile# in which EMP_DEMO table segment exist(can be checked from v$datafile)

BLOCK_ID = Starting Block# for the EMP_DEMO table segment

BLOCKS = Nos. of blocks starting from first block#. In this example the EMP_DEMO table segment has been allocated 8 blocks in datafile 6 starting from block#9,10,11,12…. until 16.

4. Dump the data block and read the details.

First block in segment is having the segment header information; the actual data start from the second block of the segment, so in order to read the data we need to dump the second block of the EMP_DEMO table which is block# 10.

```
CONN SYS/ORACLE AS SYSDBA

ALTER SYSTEM DUMP DATAFILE 6 BLOCK 10;
```

Above statement will create a trace file in bdump directory, which can be opened in Notepad or WordPad.

Oracle data block dump looks like below (comments are noted in red letters).

```
Dump file c:\oracle\admin\orcl\udump\orcl_ora_2636.trc
Mon Mar 17 21:38:14 2008
ORACLE V10.1.0.2.0 - Production vsnsta=0
vsnsql=13 vsnxtr=3
Personal Oracle Database 10g Release 10.1.0.2.0 - Production
With the Partitioning, OLAP and Data Mining options
Windows XP Version V5.1 Service Pack 2
CPU             : 1 - type 586
Process Affinity: 0x00000000
Memory (A/P)    : PH:68M/511M, PG:603M/1247M, VA:1756M/2047M
Instance name: orcl

Redo thread mounted by this instance: 1

Oracle process number: 13

Windows thread id: 2636, image: ORACLE.EXE (SHAD)


*** SERVICE NAME:(SYS$USERS) 2008-03-17 21:38:14.606
*** SESSION ID:(162.12) 2008-03-17 21:38:14.606
Start dump data blocks tsn: 6 file#: 6 minblk 10 maxblk 10
        (Data file number#6 ,  Block number From 10 To 10)

buffer tsn: 6 rdba: 0x0180000a (6/10)
        (Related Database Block Address - file#6 and block# 10)

scn: 0x0000.0015c34e seq: 0x01 flg: 0x02 tail: 0xc34e0601
SCN the last block changed to; Convert Hex to Decimal e.g. 0015c34e = 1426254)

frmt: 0x02 chkval: 0x0000 type: 0x06=trans data
```

```
Block header dump:  0x0180000a
 Object id on Block? Y
 seg/obj: 0xc2c8  csc: 0x00.15c34c  itc: 2  flg: O  typ: 1 - DATA
                                (itc:2 = Nos of ITL slots)
     fsl: 0  fnx: 0x0 ver: 0x01

 Itl           Xid                  Uba         Flag  Lck        Scn/Fsc
0x01   0x0002.006.0000050c  0x00800168.0105.1b  --U-    1  fsc 0x0000.0015c34e
0x02   0x0000.000.00000000  0x00000000.0000.00  ----    0  fsc 0x0000.00000000

ITL - Interested Transaction List

Xid - Transaction ID

Uba - Undo Block Address
Flag - Transaction flag
       C---    = Committed,
       ----    = Uncommitted
       -B--    = The UBA (Undo Block Address) contains undo for this ITL
       --U-    = Committed by fast commits & delayed block cleanout has not occurred
       ---T    = Transaction active at block cleanout SCN
       -C-U-   = Block cleaned by delayed block cleanout, and rollback segment info is
                 overwritten.
Lck - Numbers of Row locked due to related transaction
Scn/Fsc- System Change Number / Free space credit


data_block_dump,data header at 0x5b1125c
===============
tsiz: 0x1fa0                          # Size of the block 0x1fa0 = 8096 bytes (8KB)
hsiz: 0x14                            # Size of the block header = 0x14 = 20 bytes
pbl: 0x05b1125c                       # Pointer to block buffer holding the block
bdba: 0x0180000a                      # Relative database block address
     76543210
flag=--------
ntab=1                                # No of tables in block
nrow=1                                # No of rows
frre=-1                               # First free row index entry , if -1 add 1
fsbo=0x14                             # Free space begin offset
fseo=0x1f96                           # Free space end offset
avsp=0x1f82                           # Available block space
tosp=0x1f82                           # Available space post commit
0xe:pti[0]      nrow=1  offs=0        # Table info
0x12:pri[0]     offs=0x1f96           # Row info Rowid
block_row_dump:                       # Row Data next
tab 0, row 0, @0x1f96                 # Table 0, Row 0 = 1(row start with 0), rowid
tl: 10 fb: --H-FL-- lb: 0x1  cc: 2    # lb: 0x1 = This row is locked by ITL -0x01
col  0: [ 2]  c1 02                   # c1 = tenth 02 = 2(Decimal)-1= 1 - Hex to Decimal
col  1: [ 3]  44 41 4e                # 44 41 4e = DAN - Hex to Char
end_of_block_dump
End dump data blocks tsn: 6 file#: 6 minblk 10 maxblk 10
```

In above block dump…

Itl	Xid	Uba	Flag	Lck	Scn/Fsc
0x01	0x0002.006.0000050c	0x00800168.0105.1b	--U-	1	fsc 0x0000.0015c34e

The insert was done by

ITL(Interested Transaction Lock) = 0x01

Xid (Transaction ID) = 0x0002.006.0000050c

Uba (Undo Byte Address) = 0x00800168.0105.1b

Flag (Transaction Status flag) = `--U-` (Committed by fast commit)

Lck (No of Row locked due to this transaction) = 1

Scn /Fsc: System Change Number/ Free Space Credit (if that data is deleted from the block)

5. Note the block level SCN.

scn: 0x0000.0015c34e, covert Hex to Dec = 1426254 (Insert –Committed)

6. Record the system date/time before and after update the row without commit.

```
CONN DEMO/DEMO

SELECT TO_CHAR(SYSDATE,'DD-MON-YYYY HH:MI:SS') FROM DUAL;

TO_CHAR(SYSDATE,'DD-
--------------------
17-MAR-2008 09:39:06

UPDATE EMP_DEMO
SET EMPNAME = 'SCOTT';

SELECT TO_CHAR(SYSDATE,'DD-MON-YYYY HH:MI:SS') FROM DUAL;

TO_CHAR(SYSDATE,'DD-
--------------------
17-MAR-2008 09:39:29
```

7. Find the related UNDO block containing the undo information for previous update.

```
SET LINESIZE 142
SELECT UBAFIL,UBABLK,UBASQN,UBAREC,STATUS,NOUNDO,XID,START_SCN FROM
V$TRANSACTION; # get the undo file# and block#
```

UBAFIL	UBABLK	UBASQN	UBAREC	STATUS	NOU	XID	START_SCN
2	361	261	3	ACTIVE	NO	020025000D050000	1426293

UBAFIL = Undo Block Address file number

UBABLK = Undo Block Address block number

UBASQN = Undo Block Address sequence number

STATUS = Status of the transaction

NOUNDO = if Yes then no undo

XID = Transaction id

START_SCN = SCN

8. Dump the data block. (with uncommitted transaction)

```
CONN SYS/ORACLE AS SYSDBA
ALTER SYSTEM DUMP DATAFILE 6 BLOCK 10; # Data Block
```

```
Dump file c:\oracle\admin\orcl\udump\orcl_ora_2216.trc
Mon Mar 17 21:39:56 2008
ORACLE V10.1.0.2.0 - Production vsnsta=0
vsnsql=13 vsnxtr=3
Personal Oracle Database 10g Release 10.1.0.2.0 - Production
With the Partitioning, OLAP and Data Mining options
Windows XP Version V5.1 Service Pack 2
CPU             : 1 - type 586
Process Affinity: 0x00000000
Memory (A/P)    : PH:58M/511M, PG:603M/1247M, VA:1757M/2047M
Instance name: orcl

Redo thread mounted by this instance: 1
```

```
Oracle process number: 13

Windows thread id: 2216, image: ORACLE.EXE (SHAD)


*** SERVICE NAME:(SYS$USERS) 2008-03-17 21:39:56.512
*** SESSION ID:(162.17) 2008-03-17 21:39:56.512
Start dump data blocks tsn: 6 file#: 6 minblk 10 maxblk 10
buffer tsn: 6 rdba: 0x0180000a (6/10)
scn: 0x0000.0015c375 seq: 0x01 flg: 0x00 tail: 0xc3750601
frmt: 0x02 chkval: 0x0000 type: 0x06=trans data
Block header dump:  0x0180000a
 Object id on Block? Y
 seg/obj: 0xc2c8  csc: 0x00.15c375  itc: 2  flg: O  typ: 1 - DATA
     fsl: 0  fnx: 0x0 ver: 0x01

 Itl           Xid                  Uba         Flag  Lck        Scn/Fsc
0x01   0x0002.006.0000050c  0x00800168.0105.1b  C---    0  scn 0x0000.0015c34e
0x02   0x0002.025.0000050d  0x00800169.0105.03  ----    1  fsc 0x0000.00000000

data_block_dump,data header at 0x598125c
===============
tsiz: 0x1fa0
hsiz: 0x14
pbl: 0x0598125c
bdba: 0x0180000a
     76543210
flag=---------
ntab=1
nrow=1
frre=-1
fsbo=0x14
fseo=0x1f8a
avsp=0x1f80
tosp=0x1f80
0xe:pti[0]     nrow=1  offs=0
0x12:pri[0]    offs=0x1f8a
block_row_dump:
tab 0, row 0, @0x1f8a
tl: 12 fb: --H-FL-- lb: 0x2  cc: 2
col  0: [ 2]  c1 02
col  1: [ 5]  53 43 4f 54 54
end_of_block_dump
End dump data blocks tsn: 6 file#: 6 minblk 10 maxblk 10
```

Same data block applied an update transaction to the row. So this time the flag for the first insert transaction 0x01 committed in step 4 has changed to C--- showing the committed transaction and the SCN has been issued for 0x01 transaction scn 0x0000.0015c34e (1426254). Transaction 0x02 is the update applied in step 8, which is still uncommitted, is shows the flag as ---- (uncommitted). The block level SCN is scn: 0x0000.0015c375 (1426293). It is also showing that 1 row is having a lock. This increment in SCN is showing the update took place after the insert. The data is also updated from 44 41 4e = DAN to 53 43 4f 54 54 = SCOTT.

9. Dump the Undo block identified in step 7.

```
CONN SYS/ORACLE AS SYSDBA
ALTER SYSTEM DUMP DATAFILE 2 BLOCK 361;

*** 2008-03-17 21:41:52.509
*** SERVICE NAME:(SYS$USERS) 2008-03-17 21:41:52.499
*** SESSION ID:(162.21) 2008-03-17 21:41:52.499
Start dump data blocks tsn: 1 file#: 2 minblk 361 maxblk 361
buffer tsn: 1 rdba: 0x00800169 (2/361)
scn: 0x0000.0015c375 seq: 0x01 flg: 0x00 tail: 0xc3750201
frmt: 0x02 chkval: 0x0000 type: 0x02=KTU UNDO BLOCK

*******************************************************************************
UNDO BLK:
xid: 0x0002.025.0000050d  seq: 0x105 cnt: 0x3   irb: 0x3   icl: 0x0   flg: 0x0000

 Rec Offset      Rec Offset      Rec Offset      Rec Offset      Rec Offset
----------------------------------------------------------------------------
0x01 0x1f50     0x02 0x1ebc     0x03 0x1e24
.
.
```

```
.
*-----------------------------
uba: 0x00800169.0105.01 ctl max scn: 0x0000.0015c132 prv tx scn: 0x0000.0015c135
txn start scn: scn: 0x0000.0015c375 logon user: 74
 prev brb: 8388959 prev bcl: 0
KDO undo record:
KTB Redo
op: 0x03  ver: 0x01
op: Z
KDO Op code: URP row dependencies Disabled
  xtype: XA flags: 0x00000000  bdba: 0x0180000a  hdba: 0x01800009
itli: 2  ispac: 0  maxfr: 4863
tabn: 0 slot: 0(0x0) flag: 0x2c lock: 0 ckix: 0
ncol: 2 nnew: 1 size: -2
col  1: [ 3]  44 41 4e

End dump data blocks tsn: 1 file#: 2 minblk 361 maxblk 361
```

Find the undo data inside the undo block by comparing transaction id (xid: 0x0002.025.0000050d) and data block id (bdba: 0x0180000a) from data block dump (step 8).

In above dump we can see the old value of col 1, which is **44 41 4e = DAN (Hex to Char).** This is an old value of this column
This can be used for rollback the transaction.

10. Start another session as sys user and then execute shutdown abort.

```
SHUTDOWN ABORT;
```

SHUTODOWN ABORT will crash oracle instance as of that time. It will not perform any recovery of the transaction or wait for user to finish the transaction. All the data in the cache will be lost. This requires an instance recovery on startup of the database.

To start logMiner utility to read this above update transaction in redo log files, follow the following steps:

11. Insert the following initialization parameter in init.ora file and start the database using the init.ora file to create the dictionary.

```
UTL_FILE_DIR = C:\oracle\Ora10g\orcl\oradata    #directory path can be change
STARTUP PFILE= C:\oracle\admin\orcl\pfile\init.ora;
```

During the STARTUP command SMON will check all the datafile header and identify the need of the recovery of the datafiles. No special action is necessary to do the recovery. SMON will take care of this. Oracle will read the redo log files and re-apply all the transactions after the last full checkpoint. This will involve ROLLFORWARD and followed by ROLLBACK.

12. Dump the data block again to verify the rollback of the uncommitted data in the block after the an instance recovery from above step.

```
CONN SYS/ORACLE AS SYSDBA

ALTER SYSTEM DUMP DATAFILE 6 BLOCK 10;

*** 2008-03-17 21:50:33.138
*** SERVICE NAME:(SYS$USERS) 2008-03-17 21:50:33.078
*** SESSION ID:(162.6) 2008-03-17 21:50:33.078
Start dump data blocks tsn: 6 file#: 6 minblk 10 maxblk 10
buffer tsn: 6 rdba: 0x0180000a (6/10)
scn: 0x0000.0016126d seq: 0x01 flg: 0x04 tail: 0x126d0601
frmt: 0x02 chkval: 0x9369 type: 0x06=trans data
Block header dump:  0x0180000a
 Object id on Block? Y
 seg/obj: 0xc2c8  csc: 0x00.15c375  itc: 2  flg: O  typ: 1 - DATA
     fsl: 0  fnx: 0x0 ver: 0x01

 Itl           Xid                  Uba         Flag  Lck        Scn/Fsc
0x01   0x0002.006.0000050c  0x00800168.0105.1b  C---    0  scn 0x0000.0015c34e
```

```
0x02   0x0000.000.00000000  0x00000000.0000.00  ----    0  fsc 0x0000.00000000

data_block_dump,data header at 0x5e5125c
===============
tsiz: 0x1fa0
hsiz: 0x14
pbl: 0x05e5125c
bdba: 0x0180000a
     76543210
flag=--------
ntab=1
nrow=1
frre=-1
fsbo=0x14
fseo=0x1f80
avsp=0x1f82
tosp=0x1f82
0xe:pti[0]     nrow=1  offs=0
0x12:pri[0]    offs=0x1f80
block_row_dump:
tab 0, row 0, @0x1f80
tl: 10 fb: --H-FL-- lb: 0x0  cc: 2
col  0: [ 2]  c1 02
col  1: [ 3]  44 41 4e
end_of_block_dump
End dump data blocks tsn: 6 file#: 6 minblk 10 maxblk 10
```

Compare the block dump from step 8(with uncommitted update transaction) with step 13(after shutdown abort and restart the database with crash recovery)

Before instance crash (step 8)

```
scn: 0x0000.0015c375 seq: 0x01 flg: 0x00 tail: 0xc3750601
frmt: 0x02 chkval: 0x0000 type: 0x06=trans data
Block header dump:  0x0180000a
 Object id on Block? Y
 seg/obj: 0xc2c8  csc: 0x00.15c375  itc: 2  flg: O  typ: 1 - DATA
     fsl: 0  fnx: 0x0 ver: 0x01

 Itl           Xid                  Uba         Flag  Lck        Scn/Fsc
0x01   0x0002.006.0000050c  0x00800168.0105.1b  C---    0  scn 0x0000.0015c34e
0x02   0x0002.025.0000050d  0x00800169.0105.03  ----    1  fsc 0x0000.00000000
```

After instance crash recovery (step 13)

```
scn: 0x0000.0016126d seq: 0x01 flg: 0x04 tail: 0x126d0601
frmt: 0x02 chkval: 0x9369 type: 0x06=trans data
Block header dump:  0x0180000a
 Object id on Block? Y
 seg/obj: 0xc2c8  csc: 0x00.15c375  itc: 2  flg: O  typ: 1 - DATA
     fsl: 0  fnx: 0x0 ver: 0x01

 Itl           Xid                  Uba         Flag  Lck        Scn/Fsc
0x01   0x0002.006.0000050c  0x00800168.0105.1b  C---    0  scn 0x0000.0015c34e
0x02   0x0000.000.00000000  0x00000000.0000.00  ----    0  fsc 0x0000.00000000
```

SCN from step 8 is **scn: 0x0000.0015c375(1426293)**
SCN from step 13 is **scn: 0x0000.0016126d(1446509)**
SCN is increased proves that the transaction has been took place during the recovery.
Also, transaction 0x02 (Update) has been cleared shows that the transaction has been rollback and the data value has been reset to the original value from **53 43 4f 54 54 (SCOTT) to 44 41 4e(DAN).**

13. Compare the SCN from above steps.

    SCN with Insert after Commit = 1426254

    SCN with Update before Commit = 1426293

    SCN with after instance recovery = 1446509

    SCN increased during each transaction.

14. Execute following steps to read the redo log files

```
CONN SYS/ORACLE AS SYSDBA

#Build the dictionary
EXECUTE DBMS_LOGMNR_D.BUILD(DICTIONARY_FILENAME => 'orcldict.ora',DICTIONARY_LOCATION =>
'C:\oracle\Ora10g\orcl\oradata');

#Add redo log files for LogMiner utility
execute dbms_logmnr.add_logfile(logfilename => 'C:\oracle\Ora10g\orcl\redo\REDO01.LOG',options =>
dbms_logmnr.new);

execute dbms_logmnr.add_logfile(logfilename => 'C:\oracle\Ora10g\orcl\redo\REDO02.LOG',options =>
dbms_logmnr.addfile);

execute dbms_logmnr.add_logfile(logfilename => 'C:\oracle\Ora10g\orcl\redo\REDO03.LOG',options =>
dbms_logmnr.addfile);
```

15. Check the last checkpoint time for all the datafiles

```
set linesize 100
column name format a60
select NAME,to_char(checkpoint_time, 'DD-MON-YYYY HH:MI:SS') from
v$datafile_header;
```

```
       NAME                                                  TO_CHAR(CHECKPOINT_TIME)
---------------------------------------------------------- --------------------
C:\ORACLE\ORA10G\ORCL\ORADATA\SYSTEM01.DBF                    17-MAR-2008 09:07:02
C:\ORACLE\ORA10G\ORCL\ORADATA\UNDOTBS01.DBF                   17-MAR-2008 09:07:02
C:\ORACLE\ORA10G\ORCL\ORADATA\SYSAUX01.DBF                    17-MAR-2008 09:07:02
C:\ORACLE\ORA10G\ORCL\ORADATA\USERS01.DBF                     17-MAR-2008 09:07:02
C:\ORACLE\ORA10G\ORCL\ORADATA\TEST01.DBF                      17-MAR-2008 09:07:02
C:\ORACLE\ORA10G\ORCL\ORADATA\DEMO01.DBF                      17-MAR-2008 09:07:02
```

Check the SCN

```
SELECT CURRENT_SCN,CHECKPOINT_CHANGE#,ARCHIVE_CHANGE# FROM V$DATABASE;

CURRENT_SCN CHECKPOINT_CHANGE# ARCHIVE_CHANGE#
----------- ------------------ ---------------
    1424889            1424420         1381016
```

16. Start the LogMiner Utility

```
EXECUTE dbms_logmnr.start_logmnr(DictFileName=>'C:\oracle\Ora10g\orcl\oradata\orcldict.ora');
```

17. Check the time range of each online redo log files

```
SELECT FILENAME, TO_CHAR(LOW_TIME,'DD-MON-YYYY HH:MI:SS') ,TO_CHAR(HIGH_TIME,'DD-MON-YYYY
HH:MI:ss')  FROM v$logmnr_logs;
```

```
FILENAME                                 TO_CHAR(LOW_TIME,'DD  TO_CHAR(HIGH_TIME,'D
---------------------------------------  --------------------  --------------------
C:\oracle\Ora10g\orcl\redo\REDO03.LOG    16-MAR-2008 12:43:09  17-MAR-2008 09:07:02
C:\oracle\Ora10g\orcl\redo\REDO02.LOG    17-MAR-2008 09:07:02  17-MAR-2008 09:45:39
C:\oracle\Ora10g\orcl\redo\REDO01.LOG    17-MAR-2008 09:45:39  01-JAN-1988 12:00:00
```

Compare the timestamp from step 6 before and after the update transaction with the above time ranges. It shows that the redo02.LOG file will have the redo-undo commands for this update transaction.

18. Read online redo log files for the above update transaction.

```
COLUMN SQL_REDO FORMAT A20
COLUMN SQL_UNDO FORMAT A20
select sql_redo,'---', To_char (timestamp,'DD-MON-YYYY HH:MI:SS'),'---',sql_undo,'---
',scn,operation from v$logmnr_contents WHERE USERNAME = 'DEMO' AND table_name =
'EMP_DEMO';
```

```
SQL_REDO            '-- TO_CHAR(TIMESTAMP,'D '-- SQL_UNDO            '--     SCN     OPERATION
```

```
-------------------- --- -------------------- --- -------------------- --- ---------- -----------
CREATE TABLE emp_dem --- 17-MAR-2008 09:35:57 ---                      ---    1426200 DDL
o (
empno NUMBER(10),
empname VARCHAR2(10)
);

insert into "DEMO"." --- 17-MAR-2008 09:37:58 --- delete from "DEMO"." ---    1426252 INSERT
EMP_DEMO"("EMPNO","E                              EMP_DEMO" where "EMP
MPNAME") values ('1'                              NO" = '1' and "EMPNA
,'DAN');                                          ME" = 'DAN' and ROWI
                                                  D = 'AAAMLIAAGAAAAAK

update "DEMO"."EMP_D --- 17-MAR-2008 09:39:21 --- update "DEMO"."EMP_D ---    1426293 UPDATE
EMO" set "EMPNAME" =                              EMO" set "EMPNAME" =
 'SCOTT' where "EMPN                               'DAN' where "EMPNAM
AME" = 'DAN' and ROW                              E" = 'SCOTT' and ROW
ID = 'AAAMLIAAGAAAAA                              ID = 'AAAMLIAAGAAAAA
KAAA';                                            KAAA';

update "DEMO"."EMP_D --- 17-MAR-2008 09:45:44 ---                      ---    1446509 UPDATE
EMO" set "EMPNAME" =
 'DAN' where ROWID =
'AAAMLIAAGAAAAAKAAA';
```

Above, highlighted transaction was performed during the instance recovery on startup of the database after shutdown abort. The SCN for this transaction is similar with the block dump done after startup the database in step 13 (scn: 0x0000.0016126d = 1446509). The second last transaction was the transaction done before the shutdown abort and that was uncommitted update transaction. So during the startup instance was recovered and as a part of that this transaction was also rollbacked.

CONCLUSION:

In above steps it shows that Oracle tracks transactions through transaction table and SCN. Each time any transaction takes place it gets registered with the transaction table in the header of the undo segment. This can be checked by querying the v$transaction view. If the transaction is active in any of the data block and some other process is trying to read that block, process first check in to transaction table and find out the status of the transaction. If the status is active then it takes the undo file number and undo block number and visit that undo block to get the uncommitted value. If there is no entry in transaction table then the data exists in that block is the most accurate and valid. Oracle also keeps forwarding the SCN with each change in the data. This SCN is used to provide the read consistency during the long running queries.

Oracle keeps track of the state of the transaction inside the block within the ITL with the flag. During the instance crash recovery oracle uses this transaction flag and SCN and perform the recovery from the redo log files. During the instance recovery oracle performs rollforward and then it followed by rollback.