DBAII_Lab4

Part one:Block corruption

(1) Do The following steps using SQLPLUS utility:

create tablespace testk datafile '/home/oracle/testk.dbf' size 10M autoextend off;

alter user hr quota unlimited on testk;

conn hr/hr

create table testab tablespace testk as select * from employees;

```
SQL> create tablespace testk datafile '/home/oracle/testk.dbf' size 10M autoextend off;

Tablespace created.

SQL> alter user hr quota unlimited on testk;

User altered.

SQL> conn hr/hr
Connected.
SQL> create table testab tablespace testk as select * from employees;

Table created.

SQL> insert into testab select * from employees;

2 rows created.
```

```
SQL> insert into testab select * from testab;

1280 rows created.

SQL> commit;

Commit complete.
```

create index testind on testab (EMPLOYEE_ID) tablespace testk;

```
SQL> desc testab;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 ID                                                 NUMBER

SQL> create index testind on testab (ID) tablespace testk;

Index created.
```

(2) Perform RMAN cold backup for the tablespace testk

* SQL> alter tablespace testk offline;

```
SQL>
SQL> alter tablespace testk offline;

Tablespace altered.
```

- RMAN> backup tablespace testk tag "KKK";

```
[oracle@node1 ~]$ rman target /

Recovery Manager: Release 19.0.0.0.0 - Production on Mon Apr 8 00:49:42 2024
Version 19.3.0.0.0

Copyright (c) 1982, 2019, Oracle and/or its affiliates.  All rights reserved.

connected to target database: ITI (DBID=2732047794)

RMAN> backup tablespace testk tag "KKK";

Starting backup at 08-APR-2024 00:50:27
using target database control file instead of recovery catalog
RMAN-06908: warning: operation will not run in parallel on the allocated channels
RMAN-06909: warning: parallelism require Enterprise Edition
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=19 device type=DISK
channel ORA_DISK_1: starting full datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
input datafile file number=00004 name=/home/oracle/testk.dbf
channel ORA_DISK_1: starting piece 1 at 08-APR-2024 00:50:29
channel ORA_DISK_1: finished piece 1 at 08-APR-2024 00:50:31
piece handle=/u01/app/oracle/product/19c/db_home/dbs/1o2nmmel_1_1 tag=KKK comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:02
Finished backup at 08-APR-2024 00:50:31

Starting Control File and SPFILE Autobackup at 08-APR-2024 00:50:31
piece handle=/rman_backup/c-2732047794-20240408-01 comment=NONE
Finished Control File and SPFILE Autobackup at 08-APR-2024 00:50:34

RMAN>
```

- SQL> alter tablespace testk online;

```
SQL> alter tablespace testk online;

Tablespace altered.

SQL>
```

(3) DO the following steps from the OS prompt twice

- dd of=/home/oracle/testk.dbf bs=8192 conv=notrunc seek=12 <<EOF

Give me Corruption ! EOF

```
[oracle@node1 ~]$ dd of=/home/oracle/testk.dbf bs=8192 conv=notrunc seek=12 <<EOF
> Give me Corruption !!
> EOF
0+1 records in
0+1 records out
22 bytes copied, 0.00019127 s, 115 kB/s
[oracle@node1 ~]$ dd of=/home/oracle/testk.dbf bs=8192 conv=notrunc seek=13 <<EOF
> Give me Corruption !!
> EOF
0+1 records in
0+1 records out
22 bytes copied, 0.000176091 s, 125 kB/s
[oracle@node1 ~]$
```

(4) Use DB Verify utility to verify the data corruption done in step (3)

dbv file=/home/oracle/testk.dbf blocksize=8192

```
[oracle@node1 ~]$ dbv file=/home/oracle/testk.dbf blocksize=8192

DBVERIFY: Release 19.0.0.0.0 - Production on Mon Apr 8 01:35:44 2024

Copyright (c) 1982, 2019, Oracle and/or its affiliates.  All rights reserved.

DBVERIFY - Verification starting : FILE = /home/oracle/testk.dbf
Page 12 is marked corrupt
Invalid temporary block relative dba: 0x0100000c (file 4, block 12)
Bad header found during dbv:
Data in bad block:
 type: 71 format: 1 rdba: 0x20656d20
 last change scn: 0x6576.7075.72726f43 seq: 0x74 flg: 0x69
 spare3: 0x2120
 consistency value in tail: 0x67701e01
 check value in block header: 0x6e6f
 block checksum disabled

Page 13 is marked corrupt
Invalid temporary block relative dba: 0x0100000d (file 4, block 13)
Bad header found during dbv:
Data in bad block:
 type: 71 format: 1 rdba: 0x20656d20
 last change scn: 0x6576.7075.72726f43 seq: 0x74 flg: 0x69
 spare3: 0x2120
 consistency value in tail: 0x67701e01
 check value in block header: 0x6e6f
 block checksum disabled
```

```
DBVERIFY - Verification complete

Total Pages Examined         : 1280
Total Pages Processed (Data) : 5
Total Pages Failing   (Data) : 0
Total Pages Processed (Index): 6
Total Pages Failing   (Index): 0
Total Pages Processed (Other): 131
Total Pages Processed (Seg)  : 0
Total Pages Failing   (Seg)  : 0
Total Pages Empty            : 1136
Total Pages Marked Corrupt   : 2
Total Pages Influx           : 0
Total Pages Encrypted        : 0
Highest block SCN            : 3304017 (0.3304017)
[oracle@node1 ~]$
```

(5) Use DBMS Repair package to only verify (Not Fix) the data corruption done in step (3)

You will use admin_tables , check_objects , dump_orphan_keys (please check repair_table and orphan_table after check_objects procedure)

```
SQL> SQL> BEGIN
  DBMS_REPAIR.admin_tables (
    table_name => 'REPAIR_TABLE',
    table_type => DBMS_REPAIR.repair_table,
    action     => DBMS_REPAIR.create_action,
    tablespace => 'USERS'
  );

  DBMS_REPAIR.admin_tables (
    table_name => 'ORPHAN_KEY_TABLE',
    table_type => DBMS_REPAIR.orphan_table,
    action     => DBMS_REPAIR.create_action,
    tablespace => 'USERS'
  );
END;
/  2    3    4    5    6    7    8    9   10   11   12   13   14   15   16

PL/SQL procedure successfully completed.
```

```
SQL>  SET SERVEROUTPUT ON
 DECLARE
    v_num_corrupt INT;
    BEGIN
    v_num_corrupt := 0;
    DBMS_REPAIR.check_object (
    schema_name       => 'HR',
    object_name       => 'TESTAB',
    repair_table_name => 'REPAIR_TABLE',
    corrupt_count     =>  v_num_corrupt);
    DBMS_OUTPUT.put_line('number corrupt: ' || TO_CHAR (v_num_corrupt));
  END;
SQL>   2    3    4    5    6    7    8    9   10   11   12  /
number corrupt: 0

PL/SQL procedure successfully completed.
```

```
SQL>  DECLARE
    v_num_orphans INT;
    BEGIN
      v_num_orphans := 0;
    DBMS_REPAIR.dump_orphan_keys (
        schema_name       => 'HR',
        object_name       => 'TESTIND',
        object_type       => DBMS_REPAIR.index_object,
        repair_table_name => 'REPAIR_TABLE',
       orphan_table_name => 'ORPHAN_KEY_TABLE',
       key_count         => v_num_orphans);
     DBMS_OUTPUT.put_line('orphan key count: ' || TO_CHAR(v_num_orphans));
    END;  2    3    4    5    6    7    8    9   10   11   12   13
 14  /
orphan key count: 0

PL/SQL procedure successfully completed.
```

```
SQL>     DECLARE
      v_num_fix INT;
    BEGIN
      v_num_fix := 0;
      DBMS_REPAIR.fix_corrupt_blocks (
        schema_name       => 'HR',
       object_name       => 'TESTAB',
       object_type       => Dbms_Repair.table_object,
       repair_table_name => 'REPAIR_TABLE',
       fix_count         => v_num_fix);
     DBMS_OUTPUT.put_line('num fix: ' || TO_CHAR(v_num_fix));
    END;  2    3    4    5    6    7    8    9   10   11   12
 13  /
num fix: 0

PL/SQL procedure successfully completed.
```

the corruption exists in the table header and not in the data or the index

(6) Use the RMAN to verify the data corruption done in step (3)

run{

allocate channel d1 type disk;

backup check logical validate tablespace testk;

}###### OR

RMAN> Backup validate tablespace testk;

```
RMAN> Backup validate tablespace testk;

Starting backup at 08-APR-2024 01:56:00
using channel ORA_DISK_1
channel ORA_DISK_1: starting full datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
input datafile file number=00004 name=/home/oracle/testk.dbf
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:07
List of Datafiles
================
File Status Marked Corrupt Empty Blocks Blocks Examined High SCN
---- ------ -------------- ------------ --------------- ----------
4    FAILED 0              1136         1280            3304017
  File Name: /home/oracle/testk.dbf
  Block Type Blocks Failing Blocks Processed
  ---------- -------------- ----------------
  Data       0              5
  Index      0              6
  Other      2              133

validate found one or more corrupt blocks
See trace file /u01/app/oracle/diag/rdbms/iti/ITI/trace/ITI_ora_7068.trc for details
Finished backup at 08-APR-2024 01:56:10

RMAN>
```

####### Then check the view v$database_block_corruption

SQL> select * from v$database_block_corruption;

```
SQL> select * from v$database_block_corruption;

     FILE#     BLOCK#     BLOCKS CORRUPTION_CHANGE# CORRUPTIO     CON_ID
---------- ---------- ---------- ------------------ --------- ----------
         4         12          2                  0 CORRUPT            0

SQL>
```

(7) Use the RMAN to recover the corrupted blocks produced in step (3)

RMAN> blockrecover datafile <file#> block <block#> ;

```
[oracle@node1 ~]$ rman target /

Recovery Manager: Release 19.0.0.0.0 - Production on Mon Apr 8 02:17:44 2024
Version 19.3.0.0.0

Copyright (c) 1982, 2019, Oracle and/or its affiliates.  All rights reserved.

connected to target database: ITI (DBID=2732047794)

RMAN> blockrecover datafile 4 block 12 ;

Starting recover at 08-APR-2024 02:18:09
RMAN-00571: ===========================================================
RMAN-00569: =============== ERROR MESSAGE STACK FOLLOWS ===============
RMAN-00571: ===========================================================
RMAN-03002: failure of recover command at 04/08/2024 02:18:09
RMAN-05009: Block Media Recovery requires Enterprise Edition

RMAN>
```

SQL> select * from v$database_block_corruption;

```
SQL> select * from v$database_block_corruption;

     FILE#     BLOCK#     BLOCKS CORRUPTION_CHANGE# CORRUPTIO     CON_ID
---------- ---------- ---------- ------------------ --------- ----------
         4         12          2                  0 CORRUPT            0

SQL>
```

Part two: Execution Plan

1) Make sure your database is open and all datafiles are online

(2) Open A seperate session with SYS user and execute the following query (You should execute it frequently afterwards):

alter session set nls_date_format='dd-mon-yyyy hh24:mi:ss';

set linesize 140

col sql_id for a20

col event for a35

col program for a20

col process for a8

select sid,serial#,logon_time,sql_id,program,event,process,status from v$session where type <> 'BACKGROUND' order by logon_time;

```
SQL> alter session set nls_date_format='dd-mon-yyyy hh24:mi:ss';

Session altered.

SQL> set linesize 140;
SQL> col sql_id for a20;
SQL>
col event for a35;
SQL> SQL>
col program for a20;
SQL> SQL>
col process for a8;
SQL> SQL> select sid, serial#, logon_time, sql_id, program, event, process, status
from v$session
where type <> 'BACKGROUND'
order by logon_time;  2    3    4

       SID    SERIAL# LOGON_TIME          SQL_ID              PROGRAM            EVENT                          PROCESS  STATUS
---------- ---------- -------------------- ------------------- ------------------ ------------------------------ -------- --------
        17      50206 08-apr-2024 23:26:18 1u00cdvwyg6ub       sqlplus@node1.mdp.lo SQL*Net message to client       2120     ACTIVE
                                                               cal (TNS V1-V3)


SQL>
```

(3) Open a seperate session with SYS user and don't close it

(4) Do the followings (Take care of your archived logs generation rate ... delete the archived logs using RMAN if needed):

conn hr/hr

set timing on

insert into testab select * from testab;

commit;

insert into testab select * from testab;

commit;

insert into testab select * from testab;

commit;

insert into testab select * from testab;

commit;

insert into testab select * from testab;

commit;

```
SQL> insert into testab select * from testab;

40960 rows created.

Elapsed: 00:00:00.33
SQL> commit;

Commit complete.

Elapsed: 00:00:00.06
SQL>
```

(5) In a new session connect as HR and perform

set linesize 140

set pages 10000

set pagesize 10000

set timing on

EXPLAIN PLAN SET STATEMENT_ID='ALEX' FOR select * from testab a;

SELECT * FROM TABLE(DBMS_XPLAN.DISPLAY('PLAN_TABLE','ALEX','ALL'));

```
SQL> conn hr/hr
Connected.
SQL> set linesize 140;
SQL> set pages 10000;
SQL> set pagesize 10000;
SQL> set timing on;
SQL> EXPLAIN PLAN SET STATEMENT_ID='ALEX' FOR select * from testab a;

Explained.

Elapsed: 00:00:01.12
SQL> SELECT * FROM TABLE(DBMS_XPLAN.DISPLAY('PLAN_TABLE','ALEX','ALL'));

PLAN_TABLE_OUTPUT
--------------------------------------------------------------------------------------------------------------------
Plan hash value: 3287856319

-------------------------------------------------------------------------------
| Id  | Operation          | Name   | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------
|   0 | SELECT STATEMENT   |        |  2560 |  7680 |     4   (0)| 00:00:01 |
|   1 |  TABLE ACCESS FULL| TESTAB |  2560 |  7680 |     4   (0)| 00:00:01 |
-------------------------------------------------------------------------------

Query Block Name / Object Alias (identified by operation id):
-------------------------------------------------------------

   1 - SEL$1 / A@SEL$1

Column Projection Information (identified by operation id):
-----------------------------------------------------------

   1 - "A"."ID"[NUMBER,22]

18 rows selected.

Elapsed: 00:00:00.82
SQL>
```

See the time, cost, plan_hash_value of the query

select * from testab a; ---> While you are executing this query see its sql_id and event in session opened in step (2)

```
      1000
      2000

31920 rows selected.

Elapsed: 00:00:05.14
SQL>
```

(5) execute the following:

EXPLAIN PLAN SET STATEMENT_ID='KKK' for select * from hr.testab where SALARY > 8000 and FIRST_NAME like 'A%' and employee_id=147;

```
Elapsed: 00:00:00.03
SQL> EXPLAIN PLAN SET STATEMENT_ID='KKK' for select * from hr.testab where SALARY > 8000 and NAME like 'A%' and id=147;

Explained.
```

SELECT * FROM TABLE(DBMS_XPLAN.DISPLAY('PLAN_TABLE','KKK','ALL'));

```
SQL> SELECT * FROM TABLE(DBMS_XPLAN.DISPLAY('PLAN_TABLE','KKK','ALL'));

PLAN_TABLE_OUTPUT
--------------------------------------------------------------------------------
Plan hash value: 3287856319

-----------------------------------------------------------------------------
| Id  | Operation         | Name   | Rows  | Bytes | Cost (%CPU)| Time     |
-----------------------------------------------------------------------------
|   0 | SELECT STATEMENT  |        |     1 |    78 |     3   (0)| 00:00:01 |
|*  1 |  TABLE ACCESS FULL| TESTAB |     1 |    78 |     3   (0)| 00:00:01 |
-----------------------------------------------------------------------------

Query Block Name / Object Alias (identified by operation id):
-------------------------------------------------------------

PLAN_TABLE_OUTPUT
--------------------------------------------------------------------------------

   1 - SEL$1 / TESTAB@SEL$1

Predicate Information (identified by operation id):
---------------------------------------------------

   1 - filter("ID"=147 AND "SALARY">8000 AND "NAME" LIKE 'A%')

Column Projection Information (identified by operation id):
-----------------------------------------------------------


PLAN_TABLE_OUTPUT
--------------------------------------------------------------------------------
   1 - "NAME"[VARCHAR2,100], "ID"[NUMBER,22], "SALARY"[NUMBER,22]

Note
-----
   - dynamic statistics used: dynamic sampling (level=2)
```

(6) Try to create index (with name IND1 on tablespace testk) on the where conditions columns then repeat step (5) and see the difference

```
SQL> CREATE INDEX IND1 ON testab (SALARY, NAME, id) TABLESPACE testk;

Index created.

Elapsed: 00:00:00.10
SQL>
```

Part three: AWR / ASH / ADDM / Gather stats

- insert some rows in any HR schema table.

```
Elapsed: 00:00:00.15
SQL>  INSERT INTO hr.employees (employee_id, first_name, last_name, email, hire_date, job_id, salary, department_id)
VALUES (102, 'Jane', 'Smith', 'jane.smith@example.com', TO_DATE('2024-04-08', 'YYYY-MM-DD'), 'SA_REP', 70000, 80);
  2
1 row created.

Elapsed: 00:00:00.01
SQL> INSERT INTO hr.employees (employee_id, first_name, last_name, email, hire_date, job_id, salary, department_id)
VALUES (101, 'John', 'Doe', 'john.doe@example.com', TO_DATE('2024-04-08', 'YYYY-MM-DD'), 'IT_PROG', 60000, 60);

  2
1 row created.
```

- check number of rows in this table.

```
SQL> SELECT COUNT(*) FROM hr.employees;

  COUNT(*)
----------
         2
```

- check number of rows in dba_tables for this table.

```
Elapsed: 00:00:00.00
SQL> SQL>  SELECT num_rows FROM dba_tables WHERE owner = 'HR' AND table_name = 'EMPLOYEES';


  NUM_ROWS
----------


Elapsed: 00:00:00.09
```

- gather statistics for this table and check if number of rows now changed.

```
Elapsed: 00:00:00.07
SQL> BEGIN
    DBMS_STATS.GATHER_TABLE_STATS(
        ownname      => 'HR',
        tabname      => 'EMPLOYEES',
        estimate_percent => DBMS_STATS.AUTO_SAMPLE_SIZE,
        method_opt   => 'FOR ALL COLUMNS SIZE AUTO',
        cascade      => TRUE
    );
END;
  2    3    4    5    6    7    8    9   10
 11  /

PL/SQL procedure successfully completed.

Elapsed: 00:00:01.15
SQL> SELECT COUNT(*) FROM hr.employees;

  COUNT(*)
----------
         4

Elapsed: 00:00:00.00
```