All About Data Blocks Corruption in Oracle

Vijaya R. Dumpa

Data Block Overview:

Oracle allocates logical database space for all data in a database. The units of database space allocation are data blocks (also called logical blocks, Oracle blocks, or pages), extents, and segments. The next level of logical database space is an extent. An extent is a specific number of contiguous data blocks allocated for storing a specific type of information. The level of logical database storage above an extent is called a segment. The high water mark is the boundary between used and unused space in a segment.

The header contains general block information, such as the block address and the type of segment (for example, data, index, or rollback).

Table Directory, this portion of the data block contains information about the table having rows in this block.

Row Directory, this portion of the data block contains information about the actual rows in the block (including addresses for each row piece in the row data area).

Free space is allocated for insertion of new rows and for updates to rows that require additional space.

Row data, this portion of the data block contains rows in this block.

Analyze the Table structure to identify block corruption:

By analyzing the table structure and its associated objects, you can perform a detailed check of data blocks to identify block corruption:

SQL> analyze table\_name/index\_name/cluster\_name ... validate structure cascade;

Detecting data block corruption using the DBVERIFY Utility:

DBVERIFY is an external command-line utility that performs a physical data structure integrity check on an offline database. It can be used against backup files and online files. Integrity checks are significantly faster if you run against an offline database.

Restrictions:

DBVERIFY checks are limited to cache-managed blocks. It’s only for use with datafiles, it will not work against control files or redo logs.

The following example is sample output of verification for the data file system\_ts\_01.dbf. And its Start block is 9 and end block is 25. Blocksize parameter is required only if the file to be verified has a non-2kb block size. Logfile parameter specifies the file to which logging information should be written. The feedback parameter has been given the value 2 to display one dot on the screen for every 2 blocks processed.

$ dbv file=system\_ts\_01.dbf start=9 end=25 blocksize=16384 logfile=dbvsys\_ts.log feedback=2

DBVERIFY: Release 8.1.7.3.0 - Production on Fri Sep 13 14:11:52 2002

(c) Copyright 2000 Oracle Corporation. All rights reserved.

Output:

$ pg dbvsys\_ts.log

DBVERIFY: Release 8.1.7.3.0 - Production on Fri Sep 13 14:11:52 2002

(c) Copyright 2000 Oracle Corporation. All rights reserved.

DBVERIFY - Verification starting : FILE = system\_ts\_01.dbf

DBVERIFY - Verification complete

Total Pages Examined : 17

Total Pages Processed (Data) : 10

Total Pages Failing (Data) : 0

Total Pages Processed (Index) : 2

Total Pages Failing (Index) : 0

Total Pages Processed (Other) : 5

Total Pages Empty : 0

Total Pages Marked Corrupt : 0

Total Pages Influx : 0

Detecting and reporting data block corruption using the DBMS\_REPAIR package:

Note: Note that this event can only be used if the block "wrapper" is marked corrupt.

Eg: If the block reports ORA-1578.

1. Create DBMS\_REPAIR administration tables:

To Create Repair tables, run the below package.

SQL> EXEC DBMS\_REPAIR.ADMIN\_TABLES(‘REPAIR\_ADMIN’, 1,1, ‘REPAIR\_TS’);

Note that table names prefix with ‘REPAIR\_’ or ‘ORPAN\_’. If the second variable is 1, it will create ‘REAIR\_key tables, if it is 2, then it will create ‘ORPAN\_key tables.

If the thread variable is

1 then package performs ‘create’ operations.

2 then package performs ‘delete’ operations.

3 then package performs ‘drop’ operations.

2. Scanning a specific table or Index using the DBMS\_REPAIR.CHECK\_OBJECT procedure:

In the following example we check the table employee for possible corruption’s that belongs to the schema TEST. Let’s assume that we have created our administration tables called REPAIR\_ADMIN in schema SYS.

To check the table block corruption use the following procedure:

SQL> VARIABLE A NUMBER;

SQL> EXEC DBMS\_REPAIR.CHECK\_OBJECT (‘TEST’,’EMP’, NULL,

1,’REPAIR\_ADMIN’, NULL, NULL, NULL, NULL,:A);

SQL> PRINT A;

To check which block is corrupted, check in the REPAIR\_ADMIN table.

SQL> SELECT \* FROM REPAIR\_ADMIN;

3. Fixing corrupt block using the DBMS\_REPAIR.FIX\_CORRUPT\_BLOCK procedure:

SQL> VARIABLE A NUMBER;

SQL> EXEC DBMS\_REPAIR.FIX.CORRUPT\_BLOCKS (‘TEST’,’EMP’, NULL,

1,’REPARI\_ADMIN’, NULL,:A);

SQL> SELECT MARKED FROM REPAIR\_ADMIN;

If u select the EMP table now you still get the error ORA-1578.

4. Skipping corrupt blocks using the DBMS\_REPAIR. SKIP\_CORRUPT\_BLOCK procedure:

SQL> EXEC DBMS\_REPAIR. SKIP\_CORRUPT.BLOCKS (‘TEST’, ‘EMP’, 1,1);

Notice the verification of running the DBMS\_REPAIR tool. You have lost some of data. One main advantage of this tool is that you can retrieve the data past the corrupted block. However we have lost some data in the table.

5. This procedure is useful in identifying orphan keys in indexes that are pointing to corrupt rows of the table:

SQL> EXEC DBMS\_REPAIR. DUMP ORPHAN\_KEYS (‘TEST’,’IDX\_EMP’, NULL,

2, ‘REPAIR\_ADMIN’, ‘ORPHAN\_ADMIN’, NULL,:A);

If u see any records in ORPHAN\_ADMIN table you have to drop and re-create the index to avoid any inconsistencies in your queries.

6. The last thing you need to do while using the DBMS\_REPAIR package is to run the DBMS\_REPAIR.REBUILD\_FREELISTS procedure to reinitialize the free list details in the data dictionary views.

SQL> EXEC DBMS\_REPAIR.REBUILD\_FREELISTS (‘TEST’,’EMP’, NULL, 1);

NOTE

Setting events 10210, 10211, 10212, and 10225 can be done by adding the following line for each event in the init.ora file:

Event = "event\_number trace name errorstack forever, level 10"

When event 10210 is set, the data blocks are checked for corruption by checking their integrity. Data blocks that don't match the format are marked as soft corrupt.

When event 10211 is set, the index blocks are checked for corruption by checking their integrity. Index blocks that don't match the format are marked as soft corrupt.

When event 10212 is set, the cluster blocks are checked for corruption by checking their integrity. Cluster blocks that don't match the format are marked as soft corrupt.

When event 10225 is set, the fet$ and uset$ dictionary tables are checked for corruption by checking their integrity. Blocks that don't match the format are marked as soft corrupt.

Set event 10231 in the init.ora file to cause Oracle to skip software- and media-corrupted blocks when performing full table scans:

Event="10231 trace name context forever, level 10"

Set event 10233 in the init.ora file to cause Oracle to skip software- and media-corrupted blocks when performing index range scans:

Event="10233 trace name context forever, level 10"

To dump the Oracle block you can use below command from 8.x on words:

SQL> ALTER SYSTEM DUMP DATAFILE 11 block 9;

This command dumps datablock 9 in datafile11, into USER\_DUMP\_DEST directory.

Dumping Redo Logs file blocks:

SQL> ALTER SYSTEM DUMP LOGFILE ‘/usr/oracle8/product/admin/udump/rl. log’;

Rollback segments block corruption, it will cause problems (ORA-1578) while starting up the database.

With support of oracle, can use below under source parameter to startup the database.

\_CORRUPTED\_ROLLBACK\_SEGMENTS=(RBS\_1, RBS\_2)

DB\_BLOCK\_COMPUTE\_CHECKSUM

This parameter is normally used to debug corruption’s that happen on disk.

The following V$ views contain information about blocks marked logically corrupt:

V$ BACKUP\_CORRUPTION, V$COPY\_CORRUPTION

When this parameter is set, while reading a block from disk to catch, oracle will compute the checksum again and compares it with the value that is in the block.

If they differ, it indicates that the block is corrupted on disk. Oracle makes the block as corrupt and signals an error. There is an overhead involved in setting this parameter.

DB\_BLOCK\_CACHE\_PROTECT=‘TRUE’

Oracle will catch stray writes made by processes in the buffer catch.

Oracle 9i new RMAN futures:

Obtain the datafile numbers and block numbers for the corrupted blocks. Typically, you obtain this output from the standard output, the alert.log, trace files, or a media management interface. For example, you may see the following in a trace file:

ORA-01578: ORACLE data block corrupted (file # 9, block # 13)

ORA-01110: data file 9: '/oracle/dbs/tbs\_91.f'

ORA-01578: ORACLE data block corrupted (file # 2, block # 19)

ORA-01110: data file 2: '/oracle/dbs/tbs\_21.f'

$rman target =rman/rman@rmanprod

RMAN> run {

2> allocate channel ch1 type disk;

3> blockrecover datafile 9 block 13 datafile 2 block 19;

4> }

Recovering Data blocks Using Selected Backups:

# restore from backupset

BLOCKRECOVER DATAFILE 9 BLOCK 13 DATAFILE 2 BLOCK 19 FROM BACKUPSET;

# restore from datafile image copy

BLOCKRECOVER DATAFILE 9 BLOCK 13 DATAFILE 2 BLOCK 19 FROM DATAFILECOPY;

# restore from backupset with tag "mondayAM"

BLOCKRECOVER DATAFILE 9 BLOCK 13 DATAFILE 2 BLOCK 199 FROM TAG = mondayAM;

# restore using backups made before one week ago

BLOCKRECOVER DATAFILE 9 BLOCK 13 DATAFILE 2 BLOCK 19 RESTORE

UNTIL 'SYSDATE-7';

# restore using backups made before SCN 100

BLOCKRECOVER DATAFILE 9 BLOCK 13 DATAFILE 2 BLOCK 19 RESTORE UNTIL SCN 100;

# restore using backups made before log sequence 7024

BLOCKRECOVER DATAFILE 9 BLOCK 13 DATAFILE 2 BLOCK 19 RESTORE

UNTIL SEQUENCE 7024;