Resolving corruption issue with file_hdrs dump

Filed under: Oracle database internals, corruption, recovery — orainternals @ 2:17 pm
Edit This
Tags: controlf, dump, file_hdrs, ORA-01171, oradebug

One of our client had an interesting and bizarre corruption issue. Intermittently their database is corrupted.

ORA-01171: datafile 178 going offline due to error advancing checkpoint
ORA-01122: database file 178 failed verification check
ORA-01110: data file 178: '/app/u04/oradata/somedb/some_data_01.dbf'
ORA-01251: Unknown File Header Version read for file number 178

Error message printed above indicates that file header is corrupted. This is not a one time issue and recurring many times in the past 45 days or so. Database become unusable and only option was to restore the database from backup and recover it. A painful and business impacting issue. Client's frustration is understandable.

Of course, Client tried normal channels to resolve this problem with out luck. Opening Service tickets with vendors, Relink software stack, update dbid, modify kernel parameters and replace hardware etc. Client has even planned to reinstall OS soon. Still, problem re-occurred every week. That's when he reached out to us.

file_hdrs dump

Realizing that file header is corrupted, First thing, I wanted to do was that to find what exactly is there in those file headers. What type of corruption in those file headers? Is it a block formatted with null characters? or Is it that few fields are corrupted? This will give more clues and direct us in the right path. So, I asked the client to perform a file_hdrs dump when this problem re-occurs. Following commands were sent to the client.

```
oradebug setmypid
oradebug dump file_hdrs 10
exit
```

Above statement is to dump file headers at level 10. oradebug file_hdrs dump will print blocks from the file headers.

```
oradebug setmypid
oradebug dump controlf 10
```

Second dump (controlf dump) should print control file records to the trace file. Idea here is to match the file headers and control file file header section to see if we gain more insights in to this corruption issue.

It is quite possible that this could be a control file corruption and control file corruption can’t be ruled out yet. File headers and files can be corrupted. Essentially, this needs cross verification between control file and file headers to identify the root cause. At least, I was hopeful that this will point us in right direction.

dd of file header block

I also requested the client to take a backup of first few blocks of the corrupted file using ‘dd if=filename of=filehdr.out bs=8192 count=2′ command. In UNIX platform, this dd command will copy first two blocks (of block size 8K) in to a flat file. After every corruption, only workaround is to restore and recover the database and I wanted to collect as much information as possible in one round.

What happens on Friday?

Client also pointed out that this issue happens on Fridays only. Various groups were checking UNIX, SAN layers to see if there is anything special about Friday, looking for maintenance jobs that runs on Friday etc. Nothing of any significance shows up. At this point, we were waiting for the problem to reoccur.

And then it did…

file_hdr and controlf trace files

We received trace files from the client and reviewed it. We will concentrate on just one corrupted data file, even though 68 files were corrupted at this point.

control file section for that file printed below:

```
DATA FILE #32:
(name #41) /app/u04/oradata/somedb/somedata1.dbf
 creation size=12800 block size=8192 status=0x1c head=41 tail=41 dup=1
 tablespace 32, index=33 krfil=32 prev_file=0
 unrecoverable scn: 0x0000.00000000 01/01/1988 00:00:00
 Checkpoint cnt:48 scn: 0x0910.62fd2f34 05/08/2009 10:30:32
 Stop scn: 0x0910.6305098e 05/08/2009 17:42:13
 Creation Checkpointed at scn:  0x0000.000935a6 04/30/2009 21:42:29
```

Nothing jumps out. Checkpoint SCN is recent enough [Of course, this problem happened in May 2009].
Let’s review the file header section for that corrupted file. Clearly , there is a corruption since controlf dump command itself is unable to print correct file header version.

```
File header version cannot be determined due to corruption <<<<
Dump may be suspect
 V10 STYLE FILE HEADER:
        Compatibility Vsn = 169870080=0xa200300
        Db ID=1755378070=0x68a0f196, Db Name='SOMEDB'
        Activation ID=0=0x0
        Control Seq=3032023=0x2e43d7, File size=401792=0x62180
        File Number=18, Blksiz=8192, File Type=3 DATA
```

If the file header is filled with null then above section starting with ‘V10 STYLE FILE HEADER:’ can not be printed. oradebug command is complaining that file header version can not be determined, while also printing fields from file header. Confusing and that doesn’t make sense: File header version is corrupt but fields from file headers can be printed. This means that we might need to review the output of dd command. But, before going that route , decided to compare corrupted file header output with another uncorrupted file in the control file trace file to see if we can spot any obvious issue(s).

controlf trace file for an uncorrupted file

Following first two sections prints the control file record section for a uncorrupted file. Checkpoint scn time line is matching for corrupted and uncorrupted data file records. This gives us a clue that control file record for these files may not be corrupted. Genuinely, there must be a corruption in the file header.

```
DATA FILE #33:
  (name #42) /app/u03/oradata/qnoldv01/ANOTHER_GOOD_FILE_01.dbf
creation size=12800 block size=8192 status=0xe head=42 tail=42 dup=1
 tablespace 33, index=34 krfil=33 prev_file=0
 unrecoverable scn: 0x0000.00000000 01/01/1988 00:00:00
 Checkpoint cnt:53 scn: 0x0910.641e926d 05/13/2009 16:25:26
 Stop scn: 0xffff.ffffffff 05/02/2009 13:41:42
 Creation Checkpointed at scn:  0x0000.000935c5 04/30/2009 21:42:31
 thread:1 rba:(0x3.e3bbd.10)
```

Let’s also review the file header section of an uncorrupted data file.

```
 Hot Backup end marker scn: 0x0000.00000000
 aux_file is NOT DEFINED
 V10 STYLE FILE HEADER:
        Compatibility Vsn = 169870080=0xa200300
        Db ID=1908337442=0x71beeb22, Db Name='SOMEDB'
        Activation ID=0=0x0
        Control Seq=34895=0x884f, File size=2434616=0x252638
        File Number=33, Blksiz=8192, File Type=3 DATA
```

Comparison of corrupted and uncorrupted file is not showing any insights either. OR Is it?

Not so fast…

There is a difference in the file header section between these these two file headers. Let me reprint both these file headers side-by-side to see if the difference can be spotted easily.

Corrupted file:

```
File header version cannot be determined due to corruption
Dump may be suspect
 V10 STYLE FILE HEADER:
        Compatibility Vsn = 169870080=0xa200300
        Db ID=1755378070=0x68a0f196, Db Name='SOMEDB'
        Activation ID=0=0x0
        Control Seq=3032023=0x2e43d7, File size=401792=0x62180
        File Number=18, Blksiz=8192, File Type=3 DATA
```

Uncorrupted file:

```
 Hot Backup end marker scn: 0x0000.00000000
 aux_file is NOT DEFINED
 V10 STYLE FILE HEADER:
        Compatibility Vsn = 169870080=0xa200300
        Db ID=1908337442=0x71beeb22, Db Name='SOMEDB'
        Activation ID=0=0x0
        Control Seq=34895=0x884f, File size=2434616=0x252638
        File Number=33, Blksiz=8192, File Type=3 DATA
```

Ahh.. Right there. DBID is different between these two files! How can the DBID be different in this situation? Client assured me that there is no transportable tablespace or read only tablespaces shared between databases going on here. How can the dbids different between two files in the same database? These tablespaces are both opened with read and write activity. We need to follow this clue.

```
Corrupted file:     :Db ID=1755378070=0x68a0f196 , Db Name='SOMEDB'
...
Uncorruped file     :Db ID=1908337442=0x71beeb22 , Db Name='SOMEDB'
```

After pointing out that dbid is different between uncorrupted and corrupted files, client dumped control files from other development and test databases to see if they can locate a db with dbid 1755378070. No databases had the same dbid.

Finally..

With dbid difference in mind, few hours later, client DBA recollected something. When they migrated from old database server to new database server, they used scp to move the database files from old server to new server. Of course, client DBA did not want scp connection to die when his VPN connection dies and so wrote a script to scp the files from cron entry as any good DBA will do.

After migration, system admins changed oracle password. Machine was supposed to be shutdown and kept in pristine condition for few weeks before returning the hardware. Unfortunately, it was not down.

I am sure, you figured out where I am going with it. There was a cron entry scheduled in that old DB server and that was quietly scp'ing the files from old database server to new database server corrupting the database. Client disabled that cron entry and also created new ssh key to avoid these issues.

I heard that client is able to enjoy his Fridays ☺

PS: Thanks to the client for allowing me to blog about this issue.