

SDCC 2017 | 上海

数据库核心技术与应用实战峰会

CSDN

我对oracle非常规恢复的认识

演讲者:程飞

网名:惜分飞

邮箱: dba@xifenfei.com

技术BLOG: www.xifenfie.com

预备知识

- **控制文件相关scn**
- `v$database.checkpoint_change#`
- `v$datafile.checkpoint_change#`
- **数据文件相关scn**
- `v$datafile_header.checkpoint_change#`
- **数据库干净判断**
- `v$datafile_header.FUZZY`（数据文件头）
- `v$datafile.last_change#`（控制文件）
- **dump分析**
- `alter system dump datafile '/u01/oradata/orcl/system01.dbf' block 60;--dump block`（mount状态）
- `ALTER SESSION SET EVENTS 'immediate trace name file_hdrs level 10';--File Header`
- `ALTER SESSION SET EVENTS 'immediate trace name controlf level 10';--Control file`
- `ALTER SESSION SET EVENTS 'immediate trace name redohdr level 10';--Redo log Header`
- **数据库open过程(10046跟踪)**
- `alter session set events '10046 trace name context forever, level 12';`

Oracle数据库非常规恢复常用招数

- **招式一：**
通过隐含参数,event等方式强制open数据库；
- **招式二：**
通过bbed, kfed, dd, ue等工具欺骗数据库然后open数据库；
- **招式三：**
通过dul之类工具直接绕过数据库验证直接读取文件恢复数据。

隐含参数,event等方式强制open数据库

- **_allow_resetlogs_corruption**

Active/Current redo log 坏块, IO错误, 丢失等因为redo log异常导致数据库不能启动
主要是屏蔽redo前滚, 强制打开数据库, 可能导致redo中数据丢失, 使用需要慎重

- **_offline/corrupted_rollback_segments**

Undo段出现异常无法正常回滚回滚事务,导致数据库无法打开, 例如含回滚事务的回滚段block出现坏块, undo文件丢失, 回滚段和redo前滚信息不一致等

通过设置该参数屏蔽回滚段(该回滚段未提交事务自动提交), 将导致数据不一致, 使用需要慎重

- **常见event**

event='10513 trace name context forever, level 2' --设置10513事件来临时禁止SMON恢复事务

event='10231 trace name context forever, level 10' --跳过全表扫描坏块

举例说明(一)----redo异常

• SQL> startup

ORACLE 例程已经启动。

Total System Global Area 1581916160 bytes

Fixed Size 1336060 bytes

Variable Size 964693252 bytes

Database Buffers 603979776 bytes

Redo Buffers 11907072 bytes

数据库装载完毕。

ORA-00368: 重做日志块中的校验和错误

ORA-00353: 日志损坏接近块 12014 更改 9743799889 时间 12/05/2011 09:21:11

ORA-00312: 联机日志 3 线程 1: 'R:\ORADATA\HZYL\REDO03.LOG'

SQL> select a.group#,a.status,b.member from v\$log a,v\$logfile b

2 where a.group#=b.group#;

| GROUP# | STATUS | MEMBER |
|--------|----------|----------------------------|
| 3 | CURRENT | R:\ORADATA\HZYL\REDO03.LOG |
| 2 | INACTIVE | R:\ORADATA\HZYL\REDO02.LOG |
| 1 | INACTIVE | R:\ORADATA\HZYL\REDO01.LOG |

- SQL> recover database until cancel; ----因为需要的redo损坏，无法继续
- SQL>alter database open resetlogs

ORA-01547: 警告: RECOVER 成功但 OPEN RESETLOGS 将出现如下错误

ORA-01194: 文件 1 需要更多的恢复来保持一致性

ORA-01110: 数据文件 1: 'R:\ORADATA\HZYL\SYSTEM01.DBF'

- 设置 `_allow_resetlogs_corruption=TRUE`
SQL> recover database until cancel; ----cancel
SQL> alter database open resetlogs;
- 导出导入重建数据库

举例说明(二)----undo异常

- 数据库启动alert日志报如下错误

Tue Feb 14 09:34:11 2012

Errors in file d:\oracle\product\10.2.0\admin\interlib\bdump\interlib_smon_2784.trc:

ORA-01595: error freeing extent (2) of rollback segment (3))

ORA-00607: Internal error occurred while making a change to a data block

ORA-00600: internal error code, arguments: [4194], [6], [30], [], [], [], [], []

Tue Feb 14 09:35:34 2012

Errors in file d:\oracle\product\10.2.0\admin\interlib\udump\interlib_ora_2824.trc:

ORA-00603: ORACLE server session terminated by fatal error

ORA-00600: internal error code, arguments: [4193], [2005], [2008], [], [], [], [], []

ORA-00600: internal error code, arguments: [4193], [2005], [2008], [], [], [], [], []

- 设置undo manual管理模式
undo_tablespace=SYSTEM
undo_management=MANUAL
- 数据库未启动成功，依然报ORA-600[4193/4194]
- **_offline_rollback_segments=_SYSSMU3\$**
--(注意11g开始undo segment名称有时间戳)
- 启动数据库成功，创建新undo，删除故障undo
- 导出导入重建数据库

通过bbed, kfed, dd, ue等工具欺骗数据库然后open数据库

- **bbed**

BBED是Oracle的一款内部工具，可以直接修改Oracle数据文件块的内容，在一些极端恢复场景下比较有用。使用起来也很方便，但该工具不受Oracle支持，所以默认是没有生成可执行文件的，在使用前需要编译（11g开始Oracle软件不含编译文件需要从10g中拷贝）

- **kfed**

kfed是一个没有官方文档记录的ASM工具，它可以用来读取和修改ASM的元数据块。它本身是一个独立的工具，独立于ASM实例，因此不管实例是否启动，ASM磁盘组是否mount，它都可以正常使用。kfed最为强大的地方在于它可以修复ASM损坏的元数据。10g需要人工编译，11.2及其以后版本直接使用

- **dd**

主要是为了实现数据文件填充，或者跳过坏块拼接文件，构造坏块等

- **ue**

主要是在某些特殊情况下，用来分析二进制文件，修改二进制文件等

bbed修改block试验

- SQL> create table hr.b(id number,name varchar2(100)) tablespace xff;
- SQL> insert into hr.b values(1,'aaa');
- SQL> Commit

- SQL> select rowid,
2 dbms_rowid.rowid_relative_fno(rowid)rel_fno,
3 dbms_rowid.rowid_block_number(rowid)blockno,
4 dbms_rowid.rowid_row_number(rowid) rowno
5 from hr.b;
- ROWID REL_FNO BLOCKNO ROWNO
- -----
- AAHytAACAAAIAKAAA 2 522 0

- SQL> shutdown immediate

- [oracle@localhost ~]\$ bbed parfile=/tmp/parfile.cnf
- Password:
- BBED: Release 2.0.0.0.0 – Limited Production on Sat Aug 20 17:10:24 2011
- Copyright (c) 1982, 2002, Oracle Corporation. All rights reserved.
- ***** !!! For Oracle Internal Use only !!! *****
- BBED> set dba 2,522
- DBA 0x0080020a (8389130 2,522)

- BBED> find /c aaa
- File: /opt/oracle/oradata/xifenfei/xff01.dbf (2)
- Block: 522 Offsets: 8185 to 8191 Dba:0x0080020a
- -----
- 61616101 067735

- BBED> dump /v offset 8185
- File: /opt/oracle/oradata/xifenfei/xff01.dbf (2)
- Block: 522 Offsets: 8185 to 8191 Dba:0x0080020a
- -----
- 61616101 067735 | aaa..w5

- BBED> modify /c **bbb**
- Warning: contents of previous BIFILE will be lost. Proceed? (Y/N) y
- File: /opt/oracle/oradata/xifenfei/xff01.dbf (2)
- Block: 522 Offsets: **8185** to 8191 Dbf:0x0080020a
- -----
- 62626201 067735

- BBED> **sum apply**
- Check value for File 2, Block 522:
- current = 0xa286, required = 0xa286

- SQL> startup
- SQL> select * from hr.b;
- ID NAME
- -----
- 1 **bbb**

通过ue修改oracle二进制文件修改执行计划

- Ue打开oracle二进制文件发现sql语句

```
0459c650h: 3A 35 2C 3A 36 2C 3A 37 29 00 00 00 00 00 00 00 ; :5,:6,:7).....
0459c660h: 73 65 6C 65 63 74 20 6F 2E 6F 77 6E 65 72 23 2C ; select o.owner#,
0459c670h: 6F 2E 6E 61 6D 65 2C 6F 2E 6E 61 6D 65 73 70 61 ; o.name,o.namespace
0459c680h: 63 65 2C 6F 2E 72 65 6D 6F 74 65 6F 77 6E 65 72 ; ce,o.remoteowner
0459c690h: 2C 6F 2E 6C 69 6E 6B 6E 61 6D 65 2C 6F 2E 73 75 ; ,o.linkname,o.sub
0459c6a0h: 62 6E 61 6D 65 2C 6F 2E 64 61 74 61 6F 62 6A 23 ; bname,o.dataobj#
0459c6b0h: 2C 6F 2E 66 6C 61 67 73 20 66 72 6F 6D 20 6F 62 ; ,o.flags from ob
0459c6c0h: 6A 24 20 6F 20 77 68 65 72 65 20 6F 2E 6F 62 6A ; j$ o where o.obj
0459c6d0h: 23 3D 3A 31 00 00 00 00 00 00 00 00 00 00 00 ; #=:1.....
0459c6e0h: 73 65 6C 65 63 74 20 74 79 70 65 23 2C 63 6F 6E ; select type#,con
```

- 数据库启动执行计划

```
PARSING IN CURSOR #2 len=116 dep=2 uid=0 oct=3 lid=0 tim=1345475346332403 hv=854877822 ad='2f2be060'
select o.owner#,o.name,o.namespace,o.remoteowner,o.linkname,o.subname,o.dataobj#,o.flags from obj$ o where o.obj#=:1
END OF STMT
PARSE #2:c=1000,e=1258,p=0,cr=0,cu=0,mis=1,r=0,dep=2,og=4,tim=1345475346332394
BINDS #2:
kkscoacd
Bind#0
oacdt=02 mxl=22(22) mxlc=00 mal=00 scl=00 pre=00
oacflg=08 fl2=0001 frm=00 csi=00 siz=24 off=0
kxsbbfp=b69ef000 bln=22 avl=03 flg=05
value=893
EXEC #2:c=2000,e=1382,p=0,cr=0,cu=0,mis=1,r=0,dep=2,og=4,tim=1345475346333978
WAIT #2: nam='db file sequential read' ela= 17820 file#=1 block#=220 blocks=1 obj#=-1 tim=1345475346351927
FETCH #2:c=1000,e=18054,p=1,cr=3,cu=0,mis=0,r=1,dep=2,og=4,tim=1345475346352100
STAT #2 id=1 cnt=1 pid=0 pos=1 obj=18 op='TABLE ACCESS BY INDEX ROWID OBJ$' (cr=3 pr=1 pw=0 time=18047 us)
STAT #2 id=2 cnt=1 pid=1 pos=1 obj=36 op='INDEX UNIQUE SCAN I_OBJ1' (cr=2 pr=1 pw=0 time=18014 us)
```

- 通过ue修改后语句

```

0459c650h: 3A 35 2C 3A 36 2C 3A 37 29 00 00 00 00 00 00 00 ; :5,:6,:7).....
0459c660h: 73 65 6C 65 63 74 20 6F 2E 6F 77 6E 65 72 23 2C ; select o.owner#,
0459c670h: 6F 2E 6E 61 6D 65 2C 6F 2E 6E 61 6D 65 73 70 61 ; o.name,o.namespa
0459c680h: 63 65 2C 6F 2E 72 65 6D 6F 74 65 6F 77 6E 65 72 ; ce,o.remoteowner
0459c690h: 2C 6F 2E 6C 69 6E 6B 6E 61 6D 65 2C 6F 2E 73 75 ; ,o.linkname,o.su
0459c6a0h: 62 6E 61 6D 65 2C 6F 2E 64 61 74 61 6F 62 6A 23 ; bname,o.dataobj#
0459c6b0h: 2C 6F 2E 66 6C 61 67 73 20 66 72 6F 6D 20 6F 62 ; ,o.flags from ob
0459c6c0h: 6A 24 20 6F 20 77 68 65 72 65 20 6F 62 6A 23 2B ; j$ o where obj#
0459c6d0h: 30 3E 3A 31 00 00 00 00 00 00 00 00 00 00 00 ; 0=:1.....
0459c6e0h: 73 65 6C 65 63 74 20 74 79 70 65 23 2C 63 6F 6E ; select type#,con
    
```

- 修改之后的执行计划

```

PARSING IN CURSOR #2 len=116 dep=2 uid=0 oct=3 lid=0 tim=1345475781593851 hv=493726595 ad='2f2ba76c'
select o.owner#,o.name,o.namespace,o.remoteowner,o.linkname,o.subname,o.dataobj#,o.flags from obj$ o where obj#+0=:1
END OF STMT
PARSE #2:c=1000,e=1095,p=0,cr=0,cu=0,mis=1,r=0,dep=2,og=4,tim=1345475781593840
BINDS #2:
kkscoacd
  Bind#0
    oacdty=02 mxl=22(22) mxlc=00 mal=00 scl=00 pre=00
    oacflg=08 fl2=0001 frm=00 csi=00 siz=24 off=0
    kxsbbbfp=b5a86000 bln=22 avl=03 flg=05
    value=893
EXEC #2:c=2999,e=1603,p=0,cr=0,cu=0,mis=1,r=0,dep=2,og=4,tim=1345475781595754
WAIT #2: nam='db file sequential read' ela= 18417 file#=1 block#=121 blocks=1 obj#=-1 tim=1345475781614344
WAIT #2: nam='db file sequential read' ela= 1000 file#=1 block#=123 blocks=1 obj#=-1 tim=1345475781615726
WAIT #2: nam='db file sequential read' ela= 1462 file#=1 block#=124 blocks=1 obj#=-1 tim=1345475781617720
WAIT #2: nam='db file sequential read' ela= 492 file#=1 block#=125 blocks=1 obj#=-1 tim=1345475781618452
WAIT #2: nam='db file sequential read' ela= 1358 file#=1 block#=126 blocks=1 obj#=-1 tim=1345475781620361
WAIT #2: nam='db file sequential read' ela= 500 file#=1 block#=127 blocks=1 obj#=-1 tim=1345475781621039
WAIT #2: nam='db file sequential read' ela= 63132 file#=1 block#=128 blocks=1 obj#=-1 tim=1345475781684316
WAIT #2: nam='db file sequential read' ela= 989 file#=1 block#=5233 blocks=1 obj#=-1 tim=1345475781685559
WAIT #2: nam='db file sequential read' ela= 792 file#=1 block#=5234 blocks=1 obj#=-1 tim=1345475781686583
FETCH #2:c=7999,e=90994,p=9,cr=13,cu=0,mis=0,r=1,dep=2,og=4,tim=1345475781686865
STAT #2 id=1 cnt=1 pid=0 pos=1 obj=18 op='TABLE ACCESS FULL OBJ$' (cr=13 pr=9 pw=0 time=90977 us)
    
```

通过dul之类的工具恢复

- **dul是oracle internal工具，在使用一切方法无法恢复数据库的情况下，直接读取数据文件恢复数据内容**
- **dul主要用于以下场景恢复**
 - 1) 使用前面方法无法open的数据库
 - 2) drop table 恢复
 - 3) truncate table 恢复
 - 4) 丢失system表空间/数据文件恢复
 - 5) 删除表空间（对应数据文件还存在）
 - 6) 误删除用户
 - 7) asm无法正常mount情况下，恢复数据文件和表中数据(amdudump能从不能mount的asm中抽取数据文件)
-

dul的不足之处

- 无法直接恢复pl/sql(包,存储过程,函数,触发器,视图)
- 无法直接恢复index
- 无法直接恢复表的约束关系,主外键,默认值等
- 无法直接恢复同义词,dblink等
- Clob,nvarchar2,nclob可能乱码
- 恢复出来数据,不能保证一致性(只读取数据文件,无redo和undo)
- Tde加密数据无法恢复
- Hcc压缩无法恢复
- Exadata中的数据直接恢复
- 11g secure file lobs中的加密功能无法恢复

dul不是拯救数据的万能工具，在一些特殊场景或者误操作的情况下，可能导致dul也无能为力(或者恢复工作量巨大,超过了很多企业的允许等待时间)，再次呼吁数据备份

dul使用基本演示--init.dul文件配置

- `osd_big_endian_flag=false`
- `osd_dba_file_bits=10`
- `osd_c_struct_alignment=32`
- `osd_file_leader_size=1`
- `osd_word_size = 32`
- `dc_columns=2000000`
- `dc_tables=10000`
- `dc_objects=1000000`
- `dc_users=400`
- `dc_segments=100000`
- `Buffer=10485760`
- `control_file = control.txt`
- `db_block_size=8192`
- `export_mode=true`
- `compatible=10`

dul使用基本演示--control文件配置

- 通过启动数据库到mount执行select ts#,rfile#,name from v\$datafile获得
- [oracle@xifenfei dul]\$ more control.txt
0 1 /u01/oracle/oradata/system01.dbf
1 2 /u01/oracle/oradata/undotbs01.dbf
2 3 /u01/oracle/oradata/sysaux01.dbf
4 4 /u01/oracle/oradata/users01.dbf
6 5 /u01/oracle/oradata/datfttuser.dbf

dul使用基本演示—unload 数据

- [oracle@xifenfei dul]\$./dul
-
- Data UnLoader: 11.2.0.1.1 - Internal Only - on Sat Jan 07 07:02:34 2017
- with 64-bit io functions and the decompression option
-
- Copyright (c) 1994 2016 Bernard van Duijnen All rights reserved.
-
- Strictly Oracle Internal Use Only
-
- Found db_id = 3426707456
- Found db_name = xifenfei
-
- 加载数据字典
- DUL> **BOOTSTRAP;**
-
- Unload table
- DUL> **UNLOAD TABLE hr.test;**

ORA-600[kcrf_resilver_log_1]分享

- Unpublished Bug 9056657: BOX REBOOT DURING UPGRADE CAUSED ORA-600 [KCRF_RESILVER_LOG_1]
- **There has been a lost write to the online redolog as a result of the crash.**
- The fix for this bug will raise a more meaning log corruption error rather than an ORa-00600 error.
- **Instance recovery is not possible - restore the database and do point in time recovery to the most recent archive log.**
- 参考：ORA-00600 [kcrf_resilver_log_1] on restart after system crash (文档 ID 1227666.1)

ORA-600[kcrf_resilver_log_1]案例一

- **alert日志报错**

- Sat Mar 01 18:40:44 2014
- alter database open
- Beginning crash recovery of 1 threads
- parallel recovery started with 3 processes
- Started redo scan
- Errors in file f:\app\administrator\diag\rdbms\orcl\orcl\trace\orcl_ora_6432.trc (incident=61360):
- **ORA-00600: 内部错误代码, 参数: [kcrf_resilver_log_1], [0x7FF61C56E30], [2], [], [], [], [], [], [], [], []**
- Incident details in: f:\app\administrator\diag\rdbms\orcl\orcl\incident\incdir_61360\orcl_ora_6432_i61360.trc
- Aborting crash recovery due to error 600
- Errors in file f:\app\administrator\diag\rdbms\orcl\orcl\trace\orcl_ora_6432.trc:
- **ORA-00600: 内部错误代码, 参数: [kcrf_resilver_log_1], [0x7FF61C56E30], [2], [], [], [], [], [], [], [], []**
- Errors in file f:\app\administrator\diag\rdbms\orcl\orcl\trace\orcl_ora_6432.trc:
- **ORA-00600: 内部错误代码, 参数: [kcrf_resilver_log_1], [0x7FF61C56E30], [2], [], [], [], [], [], [], [], []**
- ORA-600 signalled during: alter database open...

| DBID | NAME | OPEN MODE | CREATED | OPEN MODE | LOG MODE | CHECKPOINT CHANGE# | CTL_TYP | CTL_CREATED | CTL_CHANGE# |
|------------|------|-----------|------------------------|-----------|--------------|--------------------|---------|------------------------|-------------|
| 1350877926 | ORCL | MOUNTED | 2013-08-10 14:06:30 | MOUNTED | NOARCHIVELOG | 16551515 | CURRENT | 2013-08-10 14:06:30 | 16556135 |

- 数据文件SCN(来自控制文件v\$datafile)

| TS# | FILE# | FILE SIZE G | STATUS | ENABLED | CREATION TIME | SCN | STOP SCN |
|-----|-------|---------------|--------|------------|---------------------|----------|----------|
| 0 | 1 | .72265625 | SYSTEM | READ WRITE | 2010-03-30 10:07:48 | 16574746 | 16574746 |
| 1 | 2 | .634765625 | ONLINE | READ WRITE | 2010-03-30 10:07:52 | 16574746 | 16574746 |
| 2 | 3 | 3.3447265625 | ONLINE | READ WRITE | 2010-03-30 11:07:21 | 16574746 | 16574746 |
| 4 | 4 | .020751953125 | ONLINE | READ WRITE | 2010-03-30 10:08:04 | 16574746 | 16574746 |
| 6 | 5 | .09765625 | ONLINE | READ WRITE | 2013-08-10 14:07:57 | 16574746 | 16574746 |
| 7 | 8 | 9.765625 | ONLINE | READ WRITE | 2014-01-19 02:07:22 | 16574746 | 16574746 |
| 7 | 10 | 9.765625 | ONLINE | READ WRITE | 2014-02-28 14:15:30 | 16574746 | 16574746 |
| 7 | 11 | 7.8125 | ONLINE | READ WRITE | 2014-02-28 16:11:59 | 16574746 | 16574746 |
| 7 | 12 | 6.8359375 | ONLINE | READ WRITE | 2014-02-28 16:34:56 | 16574746 | 16574746 |
| 7 | 13 | 4.8828125 | ONLINE | READ WRITE | 2014-03-01 09:04:05 | 16574746 | 16574746 |
| 7 | 14 | 1.953125 | ONLINE | READ WRITE | 2014-03-01 09:04:48 | 16574746 | 16574746 |
| 8 | 6 | 4.8828125 | ONLINE | READ WRITE | 2014-01-17 09:48:35 | 16574746 | 16574746 |
| 8 | 7 | 9.765625 | ONLINE | READ WRITE | 2014-01-18 20:23:30 | 16574746 | 16574746 |
| 8 | 9 | .48828125 | ONLINE | READ WRITE | 2014-02-17 17:27:17 | 16574746 | 16574746 |

- 数据文件头SCN(来自数据文件头v\$datafile header)

| TS# | FILE# | TABLESPACE NAME | STATUS | ERROR | FORMAT | REC | FUZ | CREATE TIME | SCN |
|-----|-------|-----------------|--------|-------|--------|-----|-----|---------------------|----------|
| 0 | 1 | SYSTEM | ONLINE | | 10 | NO | NO | 2010-03-30 10:07:48 | 16574746 |
| 1 | 2 | SYSAUX | ONLINE | | 10 | NO | NO | 2010-03-30 10:07:52 | 16574746 |
| 2 | 3 | UNDOTBS1 | ONLINE | | 10 | NO | NO | 2010-03-30 11:07:21 | 16574746 |
| 4 | 4 | USERS | ONLINE | | 10 | NO | NO | 2010-03-30 10:08:04 | 16574746 |
| 6 | 5 | EXAMPLE | ONLINE | | 10 | NO | NO | 2013-08-10 14:07:57 | 16574746 |
| 7 | 8 | BIGDATADBT | ONLINE | | 10 | NO | NO | 2014-01-19 02:07:22 | 16574746 |
| 7 | 10 | BIGDATADBT | ONLINE | | 10 | NO | NO | 2014-02-28 14:15:30 | 16574746 |
| 7 | 11 | BIGDATADBT | ONLINE | | 10 | NO | NO | 2014-02-28 16:11:59 | 16574746 |
| 7 | 12 | BIGDATADBT | ONLINE | | 10 | NO | NO | 2014-02-28 16:34:56 | 16574746 |
| 7 | 13 | BIGDATADBT | ONLINE | | 10 | NO | NO | 2014-03-01 09:04:05 | 16574746 |
| 7 | 14 | BIGDATADBT | ONLINE | | 10 | NO | NO | 2014-03-01 09:04:48 | 16574746 |
| 8 | 6 | CENTERDBT | ONLINE | | 10 | NO | NO | 2014-01-17 09:48:35 | 16574746 |
| 8 | 7 | CENTERDBT | ONLINE | | 10 | NO | NO | 2014-01-18 20:23:30 | 16574746 |
| 8 | 9 | CENTERDBT | ONLINE | | 10 | NO | NO | 2014-02-17 17:27:17 | 16574746 |

- SQL> **recover database using backup controlfile until cancel;**
- ORA-00279: 更改 16574746 (在 03/01/2014 13:10:11 生成) 对于线程 1 是必需的
- ORA-00289: 建议:
- F:\APP\ADMINISTRATOR\FLASH_RECOVERY_AREA\ORCL\ARCHIVELOG\2014_03_01\O1_MF_1_1510_%U_.ARC
- ORA-00280: 更改 16574746 (用于线程 1) 在序列 #1510 中

- 指定日志: {<RET>=suggested | filename | AUTO | CANCEL}
- **F:\APP\ADMINISTRATOR\ORADATA\ORCL\REDO01.LOG**
- 已应用的日志。
- 完成介质恢复。
- SQL> **alter database open resetlogs;**
-
- 数据库已更改。

ORA-600[kcrf_resilver_log_1]案例二

- Tue Mar 04 09:19:22 2014
- ALTER DATABASE OPEN
- Beginning crash recovery of 1 threads
- parallel recovery started with 32 processes
- Started redo scan
- Errors in file /opt/oracle/diag/rdbms/orcl/ORCL/trace/ORCL_ora_4093.trc (incident=13393):
- **ORA-00600: internal error code, arguments: [kcrf_resilver_log_1], [0x7C0E59B40], [2], [], [], [], [], [], [], [], [], []**
- Incident details in: /opt/oracle/diag/rdbms/orcl/ORCL/incident/incdir_13393/ORCL_ora_4093_i13393.trc
- Aborting crash recovery due to error 600
- Errors in file /opt/oracle/diag/rdbms/orcl/ORCL/trace/ORCL_ora_4093.trc:
- **ORA-00600: internal error code, arguments: [kcrf_resilver_log_1], [0x7C0E59B40], [2], [], [], [], [], [], [], [], [], []**
- Errors in file /opt/oracle/diag/rdbms/orcl/ORCL/trace/ORCL_ora_4093.trc:
- **ORA-00600: internal error code, arguments: [kcrf_resilver_log_1], [0x7C0E59B40], [2], [], [], [], [], [], [], [], [], []**
- ORA-600 signalled during: ALTER DATABASE OPEN...

- 数据库SCN(v\$database)

| DBID | NAME | OPEN_MODE | CREATED | OPEN_MODE | LOG_MODE | CHECKPOINT_CHANGE# | CTL_TYP | CTL_CREATED |
|------------|------|-----------|---------------------|-----------|--------------|--------------------|---------|---------------------|
| 1366059051 | ORCL | MOUNTED | 2014-01-19 17:18:35 | MOUNTED | NOARCHIVELOG | 48503067 | CURRENT | 2014-01-19 17:18:35 |

- 数据文件SCN(v\$datafile)

| TS# | FILE# | FILE_SIZE_G | STATUS | ENABLED | CREATION_TIME | SCN | STOP_SCN |
|-----|-------|--------------|--------|------------|---------------------|----------|----------|
| 0 | 1 | 10.576171875 | SYSTEM | READ WRITE | 2009-08-15 00:16:48 | 48503067 | |
| 1 | 2 | .751953125 | ONLINE | READ WRITE | 2009-08-15 00:17:00 | 48503067 | |
| 2 | 3 | 1.025390625 | ONLINE | READ WRITE | 2009-08-15 00:52:15 | 48503067 | . |
| 4 | 4 | .0048828125 | ONLINE | READ WRITE | 2009-08-15 00:17:30 | 48503067 | |
| 6 | 5 | 15 | ONLINE | READ WRITE | 2014-01-19 19:05:25 | 48503067 | |

- 数据文件头SCN(v\$datafile_header)

| TS# | FILE# | TABLESPACE_NAME | STATUS | ERROR | FORMAT | REC | FUZ | CREATE_TIME | SCN |
|-----|-------|-----------------|--------|-------|--------|-----|-----|---------------------|----------|
| 0 | 1 | SYSTEM | ONLINE | | 10 | NO | YES | 2009-08-15 00:16:48 | 48503067 |
| 1 | 2 | SYSAUX | ONLINE | | 10 | NO | YES | 2009-08-15 00:17:00 | 48503067 |
| 2 | 3 | UNDOTBS1 | ONLINE | | 10 | NO | YES | 2009-08-15 00:52:15 | 48503067 |
| 4 | 4 | USERS | ONLINE | | 10 | NO | YES | 2009-08-15 00:17:30 | 48503067 |
| 6 | 5 | ZNKK | ONLINE | | 10 | NO | YES | 2014-01-19 19:05:25 | 48503067 |

- 直接应用redo恢复失败
- 增加_allow_resetlogs_corruption参数
- ORA-00600: internal error code, arguments: [2662], [0], [48503075], [0], [48508238], [12583040], [], [], [], [], [], []
- 推进SCN(event,隐含参数,bbed,oradebug等等)
- ORA-01595: error freeing extent (3) of rollback segment (1))
- ORA-00600: internal error code, arguments: [4194], [], [], [], [], [], [], [], [], [], [], []
- 使用_offline_rollback_segments屏蔽回滚段
- 数据库open成功，逻辑导出导入重建库

故障后现场备份

- **FS条件允许**
 - 对数据文件,redo文件, 控制文件全备
- **FS条件不允许**
 - 备份控制文件, system数据文件, 其他文件头, redo文件
- **ASM磁盘组可以mount备份**
 - 使用相关v\$或者x\$表备份文件头(类似FS条件不允许情况下备份)
- **ASM不能mount条件允许**
 - 直接对asm disk进行镜像(dd,存储复制)
- **ASM无法mount备份条件不允许**
 - 使用dd备份磁盘头前100M

数据库异常恢复注意事项

1. 了解现场(这个故障是怎么发生的, 为什么会发生这样的问题, 触发这类问题的原因, 正常运行时库是怎么样的)
2. 备份现场(进行操作之后, 不管什么后果都可以回退到初始介入状态)
3. 能够为自己的操作负责(一个命令下去, 需要知道后果, 了解原理, 最好的效果是什么, 最坏的后果是什么)
4. 通过非常规方法打开库之后, 建议逻辑方式重建库(redo, undo, 推scn, bbed修改事务槽/文件头等)
5. 尽可能最大限度恢复数据和恢复时间的抉择(比如能够open的库, 一般不使用dul), 尽快恢复业务, 当两者有冲突的时候给多种恢复方案由客户决定)

SDCC 2017 | 上海

数据库核心技术与应用实战峰会

CSDN

T H A N K S