

# Oracle 11g RAC环境ASM 故障恢复案例



2016.08.27

# 关于我

- 李信
- 河南辉煌科技 高级DBA

# 事情的经过

- 机房停电之后，Oracle 11g RAC集群启动不了！！！！
- 故障发生的时间为周四，领导下的命令是下周一必须恢复。（.....）
- 多个工程师已经到过现场.....
  - 工程师甲（集成商负责实施的工程师），表示无能为力
  - 工程师乙（维保方找的高手），一阵捣鼓之后，开始谈费用，费用之高甲方无法接受.....
  - 工程师丙也是一阵捣鼓之后，说需要收集日志发到北京总部分析.....
  - 我开始介入的时候已经是周六下午，用户能够提供QQ远程接入.....  
( 没有互联网时代的DBA怎么处理故障的？ )

# 该怎么做？

- 给甲方工程师电话（得不到有用的信息）
- 打电话给工程师甲，他说CRS死活起不来，以前是可以起来的
- 工程师乙，没联系（！！）
- 工程师丙.....
- 然后远程QQ接入处理问题

# 用户现场环境信息

- Redhat linux 6.4 x86\_64

有什么不对吗?

- Oracle 11.2.0.1 RAC+ASM

- 经过确认（一个好消息一个坏消息）：

- **好消息**是此套系统配置有备份一体机和虚拟带库。😊

- **坏消息**是没有配置备份，备份设备根本没有启用！！！！😭

- 另外，数据必须要恢复。

认证搜索

认证结果

操作系统认证

🚫 Oracle Database 11.2.0.1.0 未在 Linux x86 Red Hat Enterprise Linux 6 上通过认证

显示 Oracle Database 11.2.0.1.0 认证 (按 Linux x86 Red Hat Enterprise Linux 6 筛选) 查看认证简讯

分组方式 产品类别 显示 所有认证

认证结果

操作系统认证

🟢 Oracle Database 11.2.0.3.0 已在 Linux x86 Red Hat Enterprise Linux 6 上通过认证

有关简讯和支持信息，请参阅“认证详细资料”。

显示 Oracle Database 11.2.0.3.0 认证 (按 Linux x86 Red Hat Enterprise Linux 6 筛选) 查看认证简讯

分组方式 产品类别 显示 所有认证

# 用户现场环境信息

- 从日志分析，工程师乙应该对OHAS进行了重建
  - 大概按照文档 ID 1422517.1的步骤（ Reconfiguring & Recreating The 11gR2/12cR1 Restart/OHAS/SIHA Stack Configuration (Standalone) / How to Reconfigure Oracle Restart ）
  - 对与错？
- 数据库版本和系统版本不匹配，再加上面前几位工程师的操作，系统已经变得混乱和复杂，RAC的各个组件全部是unknown状态，无法关闭也无法启动。
- 这个时候ASM实例无法启动，使用asmcmd命令直接拷贝ASM磁盘组里面的文件已经不可能。
- AMDU提取数据文件进行恢复。

# AMDU-ASM Metadata Dump Utility

1、ASM元数据导出工具

2、ASM磁盘组中抽取元数据和文件

3、不依赖于ASM实例或者ASM磁盘组的状态

4、11g直接使用，10g需要下载二进制文件

# AMDU使用方法

```
[grid@rhel63 ~]$ amdu -diskstring="/dev/asm-*" -extract DATAVG.264 -output control.264 -noreport -nodir
```

```
AMDU-00204: Disk N0001 is in currently mounted diskgroup DATAVG
```

```
AMDU-00201: Disk N0001: '/dev/asm-data'
```

- diskstring: 使用磁盘的全路径或者是ASM\_DISKSTRING参数值
- extract: 磁盘组名.ASM文件序号
- output:提取的输出文件（默认当前目录下）
- noreport:不输出amdu的执行过程
- nodir:不创建dump目录



# AMDU使用方法

```
[grid@rhel63 ~]$ amdu -diskstring '/dev/asm-*' -extract DATAVG.264
```

```
amdu_2016_08_16_10_11_04/
```

- 简单模式下，会在当前目录下生成一个目录  
amdu\_YYYY\_MM\_DD\_HH24\_MI\_SS
- 目录下有report.txt和DATAVG\_264.f两个文件
- DATAVG\_264.f对应DATAVG.264

关键是确认ASM磁盘组名和文件序号！！

# 两种方法

- 从ASM元数据中得到磁盘组名和file number。
- Alert.log->pfile->提取control file->启动数据库到mount->视图v\$datafile, v\$logfile得到file number。 ( 非OMF格式的数据文件不适用 )

```
SQL> select name from v$datafile;
```

```
NAME
```

```
-----  
+DATAVG/orcl/datafile/system.256.801134197  
+DATAVG/orcl/datafile/sysaux.257.801134197  
+DATAVG/orcl/datafile/undotbs1.258.801134197  
+DATAVG/orcl/datafile/users.259.801134197  
+DATAVG/test.dbf
```

# KFOD - Kernel Files OSM Disk

- KFOD 在操作系统层面模拟disk discovery

```
[grid@adg1 ~]$ kfod status=TRUE asm_diskstring='/dev/asm-*' disk=all dscvgroup=TRUE
```

```
-----  
Disk      Size Header Path                Disk Group User  Group  
-----  
1:      4096 Mb MEMBER /dev/asm-crsvg          CRSVG   grid  asmadmin  
2:     20480 Mb MEMBER /dev/asm-data          DATAVG  grid  asmadmin  
3:     20480 Mb MEMBER /dev/asm-fravg         FRAVG   grid  asmadmin
```

```
KFOD-00301: Unable to contact Cluster Synchronization Services (CSS). Return code 2 from kgxgncin.
```

# ASM磁盘组DUMP

```
[grid@rhel63 ~]$ amdu -diskstring '/dev/asm-*' -dump DATAVG -noimage
[grid@rhel63 ~]$ cd amdu_2016_08_16_10_19_41/
[grid@rhel63 amdu_2016_08_16_10_19_41]$ ls
DATAVG.map report.txt
```

```
[grid@rhel63 amdu_2016_08_16_10_19_41]$ more report.txt
```

```
***** AMDU Settings
*****
ORACLE_HOME = /opt/11.2.0/grid
System name: Linux
Node name: rhel63
Release: 2.6.32-279.el6.x86_64
Version: #1 SMP Wed Jun 13 18:24:36 EDT 2012
Machine: x86_64
amdu run: 16-AUG-16 10:19:41
Endianness: 1
***** SCANNING DISKGROUP DATAVG
*****
```

```
Creation Time: 2012/10/17 18:06:54.722000
Disks Discovered: 1
Redundancy: 1
AU Size: 1048576 bytes
Metadata Block Size: 4096 bytes
Physical Sector Size: 512 bytes
Metadata Stride: 113792 AU
Duplicate Disk Numbers: 0
```

```
----- SCANNING DISK N0001 -----
```

```
Disk N0001: '/dev/asm-data'
Allocated AU's: 2398
Free AU's: 18082
AU's read for dump: 63
Block images saved: 13068
Map lines written: 63
Heartbeats seen: 0
Corrupt metadata blocks: 0
Corrupt AT blocks: 0
```

# MAP文件说明

```
[grid@rhel63 amdu_2016_08_16_10_19_41]$ more DATAVG.map
N0001 D0000 R00 A00000000 F00000000 IO E00000000 U00 C00256 S0000 B0000000000
N0001 D0000 R00 A00000001 F00000000 IO E00000000 U00 C00256 S0000 B0000000000
N0001 D0000 R00 A00000002 F00000001 IO E00000000 U00 C00256 S0000 B0000000000
.....
N0001 D0000 R00 A00000047 F00000004 IO E00000001 U00 C00000 S0000 B0000000000
N0001 D0000 R00 A00000048 F00000006 IO E00000000 U00 C00256 S0000 B0000000000
N0001 D0000 R00 A00000049 F00000009 IO E00000000 U00 C00256 S0000 B0000000000
N0001 D0000 R00 A00000050 F00000008 IO E00000000 U00 C00256 S0000 B0000000000
```

- 第1列：disk逻辑编号，从上面的report中可以看到Disk N0001对应实际的物理卷 '/dev/asm-data'。
- 第2列：asm内部定义的disk number号，DATAVG磁盘组包含1个disk，所以/dev/asm-data是0，
- 第4列：表示的au号。
- 第5列：asm内部定义的file number号。asm里面file number 0~6都是元数据。数据文件的file number都是从256开始的。
- 第9列：表示该au所分配的block大小，我这里block size是4096，由于au size是1m，所以该值基本上都是256。
- F00000006对应的AU里面存放的就是文件号和数据文件的对应关系。

# KFED - Kernel Files metadata EDitor

1、元数据编辑器

2、读取和修改ASM的元数据块

3、不依赖于ASM实例或者ASM磁盘组的状态

4、11g直接使用，10g需要手工编译后才能使用

# KFED使用方法

```
[grid@adg1 amdu_2016_08_14_23_17_08]$ kfed read /dev/asm-data aun=48 aus=1M blkcn=0
kfbh.endian:          1 ; 0x000: 0x01
kfbh.hard:            130 ; 0x001: 0x82
kfbh.type:           11 ; 0x002: KFBTYP_ALIASDIR
kfbh.datfmt:         1 ; 0x003: 0x01
kfbh.block.blk:      0 ; 0x004: blk=0
kfbh.block.obj:      6 ; 0x008: file=6
kfbh.check:          2690161383 ; 0x00c: 0xa0589ae7
kfbh.fcn.base:       4055 ; 0x010: 0x00000fd7
.....
```

- aun-读取的AU号，如果不提供值，默认为AU 0
- aus-AU的大小，默认为1048576字节（1MB），如果磁盘组不是默认的AU大小，那么需要在命令行中显式的指定AU的大小
- blkcn-读取的块号，默认为块0或者是AU的第一个block
- /dev/asm-data 磁盘或设备名称。

# 文件序号和文件名

```
for (( i=0; i<256; i++ ))
do
kfed read /dev/asm-data aun=48 blkn=$i | egrep 'name|fnum'|egrep -v 'length=0|0x00000000'
done
```

```
kfade[0].name:      ORCL ; 0x034: length=4
kfade[0].fnum:     4294967295 ; 0x064: 0xffffffff
kfade[1].name:     TEST.DBF ; 0x080: length=8
kfade[1].fnum:     270 ; 0x0b0: 0x0000010e
kfade[0].name:     DATAFILE ; 0x034: length=8
kfade[0].fnum:     4294967295 ; 0x064: 0xffffffff
kfade[1].name:     CONTROLFILE ; 0x080: length=11
kfade[1].fnum:     4294967295 ; 0x0b0: 0xffffffff
kfade[2].name:     ONLINELOG ; 0x0cc: length=9
kfade[2].fnum:     4294967295 ; 0x0fc: 0xffffffff
kfade[3].name:     TEMPFILE ; 0x118: length=8
kfade[3].fnum:     4294967295 ; 0x148: 0xffffffff
kfade[4].name:     PARAMETERFILE ; 0x164: length=13
kfade[4].fnum:     4294967295 ; 0x194: 0xffffffff
kfade[5].name:     spfileorcl.ora ; 0x1b0: length=14
kfade[5].fnum:     265 ; 0x1e0: 0x00000109
kfade[0].name:     SYSTEM ; 0x034: length=6
kfade[0].fnum:     256 ; 0x064: 0x00000100
kfade[1].name:     SYSAUX ; 0x080: length=6
kfade[1].fnum:     257 ; 0x0b0: 0x00000101
kfade[2].name:     UNDOTBS1 ; 0x0cc: length=8
```

```
kfade[2].fnum:     258 ; 0x0fc: 0x00000102
kfade[3].name:     USERS ; 0x118: length=5
kfade[3].fnum:     259 ; 0x148: 0x00000103
kfade[4].name:     TEST ; 0x164: length=4
kfade[4].fnum:     270 ; 0x194: 0x0000010e
kfade[0].name:     Current ; 0x034: length=7
kfade[0].fnum:     260 ; 0x064: 0x00000104
kfade[0].name:     group_1 ; 0x034: length=7
kfade[0].fnum:     261 ; 0x064: 0x00000105
kfade[1].name:     group_2 ; 0x080: length=7
kfade[1].fnum:     262 ; 0x0b0: 0x00000106
kfade[2].name:     group_3 ; 0x0cc: length=7
kfade[2].fnum:     263 ; 0x0fc: 0x00000107
kfade[3].name:     group_4 ; 0x118: length=7
kfade[3].fnum:     266 ; 0x148: 0x0000010a
kfade[4].name:     group_5 ; 0x164: length=7
kfade[4].fnum:     267 ; 0x194: 0x0000010b
kfade[5].name:     group_6 ; 0x1b0: length=7
kfade[5].fnum:     268 ; 0x1e0: 0x0000010c
kfade[6].name:     group_7 ; 0x1fc: length=7
kfade[6].fnum:     269 ; 0x22c: 0x0000010d
kfade[0].name:     TEMP ; 0x034: length=4
kfade[0].fnum:     264 ; 0x064: 0x00000108
kfade[0].name:     spfile ; 0x034: length=6
kfade[0].fnum:     265 ; 0x064: 0x00000109
```



# 恢复损坏的ASM磁盘组步骤

## How to Restore the Database Using AMDU after Diskgroup Corruption (文档 ID 1597581.1)

### SOLUTION

1. Create pfile of lost database , from database alert.log startup messages.

2. Startup database in nomount

3. Get the controlfile number from db alert log , it will show while starting the database here in this eg. its 256  
eg., control\_files='+DATA/orcl/controlfile/current.256.709676643'

If you have controlfile backup already in non-asm location edit the pfile to point to location of non-asm and then mount the database  
If you dont have controlfile backup then go to step 4 after determining the file# from step 3

4. \$ amdu -diskstring <asm\_diskstring> -extract DATA.256

5. shutdown the database and change the control\_file location to point to the extracted file location

6. startup mount the database

7. once mounted, get the datafile file numbers using "select name from v\$datafile"  
And get online redo logfile from "select \* from v\$logfile".

8. Extract all datafiles and redolog files in similar manner

9. alter database rename <datafile 1> to < newly extracte location>

10. open the database

# 需要考虑的几点

- 数据库大小和磁盘空间问题
- 本地恢复还是异地恢复
  - 对原有环境是否有影响
  - 异地恢复的操作系统、数据库版本等问题。
- 数据库恢复之后，原环境的重建问题。
- 数据迁移问题
- 客户端连接问题
- 过度设备的性能问题

# 手动编辑pfile

## Alert.log中内容

```
Processes= 3000
Sessions = 4608
Memory_target= 51456M
Control_files= "+DATA/devdb/controlfile/current.260.875804915"
Db_block_size= 8192
Compatible = "11.2.0.0.0"
cluster_database = TRUE
db_create_file_dest= "+DATA"
db_recovery_file_dest= "+FLASH"
db_recovery_file_dest_size= 2097036M
thread = 1
undo_tablespace= "UNDOTBS1"
instance_number= 1
remote_login_passwordfile= "EXCLUSIVE"
dispatchers= "(PROTOCOL=TCP) (SERVICE=devdbXDB)"
remote_listener= "scan-cluster.localdomain:1521"
audit_file_dest= "/u01/app/oracle/admin/devdb/adump"
audit_trail= "DB"
db_name= "devdb"
open_cursors = 300
diagnostic_dest= "/u01/app/oracle"
```

## 编辑之后内容

```
processes= 3000
sessions = 4608
memory_target= 51456M
control_files= "/oradata/current.260"
db_block_size= 8192
compatible = "11.2.0.0.0"
cluster_database = false
db_create_file_dest= "/oradata"
db_recovery_file_dest= "/oradata"
db_recovery_file_dest_size= 2097036M
undo_tablespace= "UNDOTBS1"
remote_login_passwordfile= "EXCLUSIVE"
dispatchers= "(PROTOCOL=TCP) (SERVICE=devdbXDB)"
audit_file_dest=
"/u01/app/oracle/admin/devdb/adump"
db_name= "devdb"
open_cursors = 300
diagnostic_dest= "/u01/app/oracle"
```

# 确认文件序号

- 根据pfile，提取控制文件  
current.260.875804915
- 启动数据库到mount状态，查询  
v\$datafile,v\$logfile等，得到文件序号。

# 提取文件

```
amdu -diskstring '/dev/oracleasm/disks/*' -extract DATA.256
amdu -diskstring '/dev/oracleasm/disks/*' -extract DATA.257
amdu -diskstring '/dev/oracleasm/disks/*' -extract DATA.258
amdu -diskstring '/dev/oracleasm/disks/*' -extract DATA.259
amdu -diskstring '/dev/oracleasm/disks/*' -extract DATA.261
amdu -diskstring '/dev/oracleasm/disks/*' -extract DATA.262
amdu -diskstring '/dev/oracleasm/disks/*' -extract DATA.264
amdu -diskstring '/dev/oracleasm/disks/*' -extract DATA.265
amdu -diskstring '/dev/oracleasm/disks/*' -extract DATA.266
amdu -diskstring '/dev/oracleasm/disks/*' -extract DATA.268
amdu -diskstring '/dev/oracleasm/disks/*' -extract DATA.270
amdu -diskstring '/dev/oracleasm/disks/*' -extract DATA.272
amdu -diskstring '/dev/oracleasm/disks/*' -extract DATA.271
amdu -diskstring '/dev/oracleasm/disks/*' -extract DATA.273
amdu -diskstring '/dev/oracleasm/disks/*' -extract DATA.274
amdu -diskstring '/dev/oracleasm/disks/*' -extract DATA.275
```

```
mv DATA_256.f ../system.256
mv DATA_257.f ../sysaux.257
mv DATA_258.f ../undotbs1.258
mv DATA_259.f ../users.259
mv DATA_262.f ../group_2.262
mv DATA_261.f ../group_1.261
mv DATA_264.f ../undotbs2.264
mv DATA_265.f ../group_3.265
mv DATA_266.f ../group_4.266
mv DATA_268.f ../szyjk.268
mv DATA_270.f ../sxml_data.270
mv DATA_272.f ../trsovinfo.272
mv DATA_271.f ../dat_swjczx.271
mv DATA_273.f ../trswcmv7plugins.273
mv DATA_274.f ../trsmasv6.274
mv DATA_275.f ../trswwechat.275
```

*System*表空间  
为什么这么大?

提取出来的数据文件：**system**表空间对应的数据文件**25G**,其他文件共：**40G**

# RENAME数据文件

- 在mount状态下，修改数据文件路径

```
sql>alter database rename file '+DATA/devdb/datafile/system.256.875804775' to  
'/oradata/system.256' ;
```

- 可能会遇到ORA-03113错误,虽然日志报错，实际已经修改成功。

```
ERROR: slave communication error with ASM; terminating process 12010
```

- 依次把所有文件全部rename file。

# 关闭闪回数据库

- 尝试open数据库
- 提示闪回数据库启动失败

Errors in file

/u01/app/oracle/diag/rdbms/devdb/devdb/trace/devdb\_ora\_63177.trc:

ORA-38760: This database instance failed to turn on flashback database

ORA-38760 signalled during: alter database open

- Mount状态下关闭闪回数据库功能

```
sql> alter database flashback off
```

# 关闭块更改跟踪功能

- 尝试open数据库
- 提示块更改跟踪文件无法找到

```
alter database rename file alter database open
```

```
CHANGE TRACKING is enabled for this database, but the  
change tracking file can not be found. Recreating the file.
```

```
ERROR: slave communication error with ASM; terminating process 30516
```

```
Errors in file /u01/app/oracle/diag/rdbms/devdb/devdb/trace/devdb_ora_30516.trc:
```

```
ORA-17503: ksfdopn:2 Failed to open file +DATA/devdb/changetracking/ctf.269.892571711
```

```
ORA-15001: diskgroup "DATA" does not exist or is not mounted
```

```
ORA-15077: could not locate ASM instance serving a required diskgroup
```

- 直接关闭此功能

```
sql>alter database disable block change tracking;
```



# 打开数据库

- 漫长的等待，做好了进一步恢复的准备，结果.....
- 数据库正常open了

# 后续

- 由于数据量不大，直接迁移到单机先恢复了业务
- RAC环境，进行了重新安装
- 用户部署了备份系统
- 周二上级领导到用户单位视察，各业务恢复正常

**The End**

Thanks For Your Attention!