**Automatic Storage Management**

云和恩墨(北京)信息技术有限公司

技术顾问 燕鑫

http://www.enmotech.com

**文档控制：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序** | **版本号** | **更改人** | **日期** | **备注** |
| 1 | 1.0版 | 燕鑫 | 2018-10-9 | 初始版本 |
|  |  |  |  |  |
|  |  |  |  |  |

目录

[1. ASM概要 - 6 -](#_Toc528849376)

[1.1 ASM Instances - 6 -](#_Toc528849377)

[2. Administering Oracle ASM Instances - 6 -](#_Toc528849378)

[2.1 Initialization Parameters - 6 -](#_Toc528849379)

[2.1.1 ASM SPFILE - 6 -](#_Toc528849380)

[2.1.2 配置ASM的初始化参数 - 7 -](#_Toc528849381)

[2.2 管理ASM实例 - 12 -](#_Toc528849382)

[2.2.1 Flex ASM - 12 -](#_Toc528849383)

[2.2.2 Converting to Oracle Flex ASM - 12 -](#_Toc528849384)

[2.2.3 GI for a Standalone Server - 12 -](#_Toc528849385)

[2.2.4 启动ASM实例 - 13 -](#_Toc528849386)

[2.2.5 关闭ASM实例 - 14 -](#_Toc528849387)

[2.2.6 Server Control Utility (SRVCTL) - 15 -](#_Toc528849388)

[2.2.7 Out of Place Upgrades - 16 -](#_Toc528849389)

[2.2.8 Active Session History Sampling - 16 -](#_Toc528849390)

[2.3 Authentication for Accessing - 16 -](#_Toc528849391)

[2.3.1 Using Separate Operating System Groups - 17 -](#_Toc528849392)

[2.3.2 Operating System Authentication - 18 -](#_Toc528849393)

[2.3.3 Password File Authentication - 19 -](#_Toc528849394)

[2.4 Migrating a Database to Use Oracle ASM - 20 -](#_Toc528849395)

[3. Administering Oracle ASM Disk Groups - 21 -](#_Toc528849396)

[3.1 Managing Disk Group Attributes - 21 -](#_Toc528849397)

[3.1.1 Viewing and Setting - 21 -](#_Toc528849398)

[3.1.2 Attributes - 23 -](#_Toc528849399)

[3.2 Creating Disk Groups - 24 -](#_Toc528849400)

[3.2.1 ASMCA创建磁盘组 - 24 -](#_Toc528849401)

[3.2.2 SQLPLUS创建磁盘组。 - 24 -](#_Toc528849402)

[3.2.3 存储OCR和Voting file - 28 -](#_Toc528849403)

[3.3 Altering Disk Groups - 28 -](#_Toc528849404)

[3.3.1 管理ADVM - 29 -](#_Toc528849405)

[3.3.2 增加磁盘 - 30 -](#_Toc528849406)

[3.3.3 替换磁盘 - 30 -](#_Toc528849407)

[3.3.4 重命名磁盘 - 30 -](#_Toc528849408)

[3.3.5 删除磁盘 - 31 -](#_Toc528849409)

[3.3.6 Intelligent Data Placement - 31 -](#_Toc528849410)

[3.3.7 Resizing Disks - 31 -](#_Toc528849411)

[3.3.8 Undropping Disks - 32 -](#_Toc528849412)

[3.3.9 Manually Rebalancing - 32 -](#_Toc528849413)

[3.3.10 Scrubbing - 36 -](#_Toc528849414)

[3.4 ASM Disk Discovery - 37 -](#_Toc528849415)

[3.5 Managing Capacity in Disk Groups - 39 -](#_Toc528849416)

[3.6 冗余模式 - 39 -](#_Toc528849417)

[3.6.1 三种模式 - 39 -](#_Toc528849418)

[3.6.2 如何选择冗余模式 - 40 -](#_Toc528849419)

[3.6.3 Oracle ASM Recovery from Read and Write I/O Errors - 41 -](#_Toc528849420)

[3.6.4 Fast Mirror Resync - 41 -](#_Toc528849421)

[3.6.5 Even Read for Disk Groups - 47 -](#_Toc528849422)

[3.6.6 Preferred Read Failure Groups - 47 -](#_Toc528849423)

[3.7 对于Disk Groups的建议和限制 - 47 -](#_Toc528849424)

[3.8 Disk Group Compatibility - 48 -](#_Toc528849425)

[3.8.1 设置Compatibility - 48 -](#_Toc528849426)

[3.8.2 回退Compatibility - 50 -](#_Toc528849427)

[3.8.3 Features Enabled By Disk Group Compatibility Attribute Settings - 51 -](#_Toc528849428)

[3.9 Managing Oracle ASM File Access Control - 53 -](#_Toc528849429)

[3.10 Maintaining Disk Groups - 55 -](#_Toc528849430)

[3.10.1 Mount/Dismount - 55 -](#_Toc528849431)

[3.10.2 Checking the Internal Consistency of Disk Group Metadata - 56 -](#_Toc528849432)

[3.10.3 Drop - 57 -](#_Toc528849433)

[3.10.4 Rname - 58 -](#_Toc528849434)

[4. Administering Oracle ASM Files, Directories, and Templates - 60 -](#_Toc528849435)

[4.1 ASM支持的文件类型 - 60 -](#_Toc528849436)

[4.2 ASM File Names - 60 -](#_Toc528849437)

[4.2.1 Single File Creation Form - 60 -](#_Toc528849438)

[5. 相关视图 - 61 -](#_Toc528849439)

[6. ASM Data Migration with RMAN - 61 -](#_Toc528849440)

[6.1 从ASM到磁盘 - 62 -](#_Toc528849441)

[6.2 从磁盘到ASM - 65 -](#_Toc528849442)

[6.3 数据库整体迁移到ASM（含重建redo和standbylog脚本） - 66 -](#_Toc528849443)

[7. ASMCA - 69 -](#_Toc528849444)

[8. ASMCMD - 74 -](#_Toc528849445)

[9. ADVM和ACFS - 79 -](#_Toc528849446)

# ASM概要

## ASM Instances

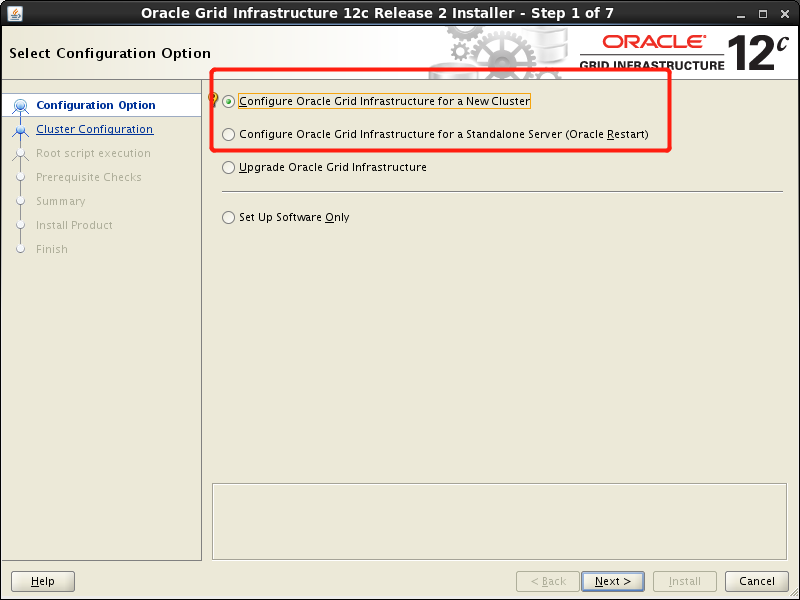
ASM instance和database instance一样，也有SGA和background processes，只是ASM实例的工作量远没有database instance的工作量大，所以所需的SGA不大，后台进程也远没有database instance多。

# Administering Oracle ASM Instances

## Initialization Parameters

### ASM SPFILE

如果装GI的时候选择第一个和第二选项的话，OUI会自动将ASM实例的spfile创建在ASM磁盘组上：



[grid@yanxin ~]$ sqlplus / as sysasm

SQL\*Plus: Release 12.2.0.1.0 Production on Wed Oct 10 17:26:52 2018

Copyright (c) 1982, 2016, Oracle. All rights reserved.

Connected to:

Oracle Database 12c Enterprise Edition Release 12.2.0.1.0 - 64bit Production

SQL> show parameter spfile

NAME TYPE VALUE

---------------- ----------- ------------------------------

spfile string +DATA/ASM/ASMPARAMETERFILE/registry.253.989164131

[grid@yanxin ~]$ asmcmd

ASMCMD> spget

+DATA/ASM/ASMPARAMETERFILE/registry.253.989164131

那么，asm实例在启动时定位spfile的搜索顺序是：

1. GPnP（Grid Plug and Play）profile中记录的spfile的位置
2. 如果上一步没找到，那么就在ASM instance home下先找spfile：$ORACLE\_HOME/dbs/spfile+ASM.ora
3. 如果还没找到，就仍然在该目录下找pfile。

备份、拷贝、移动ASM SPFILE的命令分别位：spbackup、spcopy、spmove

### 配置ASM的初始化参数

#### Automatic Memory Management

跟配置database instance是一样的，要么在pfile里面写好，一般肯定用的都是spfile，那就使用alter system/session set来配置。

ASM会自动选择Automatic Memory Management的内存管理方式。前面说了ASM实例和database实例大同小异，只是相对简单罢了。我们来看asm也有如下的参数：

SQL> show parameter target

NAME TYPE VALUE

------------------------------------ ----------- ------------------------------

memory\_max\_target big integer 1076M

memory\_target big integer 1076M

pga\_aggregate\_target big integer 0

sga\_target big integer 0

自动内存管理是官方推荐的，所以我们也不必再去纠结这些参数。

我们在用OUI装GI的时候，没有像装数据库一样，会有内存管理的选项供我们选择，因为ASM实例的自动内存管理是默认的。

另外，如果所需内存超过了memory\_max\_target，会报ORA-04031同时asm会自动将该参数以及memory\_target调大，调大的值为当前memory\_target的10%或者128M。

Memory target最小值位1G，就是设置为1G以下，也会变为1G，当然可以设置为0，禁用自动内存管理，这是不推荐的。

Linux中，如果/dev/shm太小或者不可用，那么也使用不了自动内存管理。

所以如果我们选择了自动内存管理，那么如下几个SGA中组成部分的参数也就不需要我们手动调整了：db\_cache\_size、large\_pool\_size、shared\_pool\_size

这里需要说一下，如果是手动管理内存，shared\_pool\_size的算法：

先算连接在asm实例上的数据库的所有在asm磁盘上存放的文件大小，假设该数据库把所有数据文件都放在了asm上：

SELECT SUM(bytes)/(1024\*1024\*1024) FROM V$DATAFILE;

SELECT SUM(bytes)/(1024\*1024\*1024) FROM V$LOGFILE a, V$LOG b WHERE a.group#=b.group#;

SELECT SUM(bytes)/(1024\*1024\*1024) FROM V$TEMPFILE WHERE status='ONLINE';

然后计算原则如下：

For disk groups using external redundancy, every 100 GB of space needs 1 MB of extra shared pool plus 2 MB

For disk groups using normal redundancy, every 50 GB of space needs 1 MB of extra shared pool plus 4 MB

For disk groups using high redundancy, every 33 GB of space needs 1 MB of extra shared pool plus 6 MB

#### ASM\_DISKGROUPS

如果ASM的参数文件是spfile，那么创建并挂载磁盘组，ASM就会自动修改该参数，把新建的磁盘组加进去，如果有磁盘组被remove了，那么ASM就会自动把相应的磁盘组从该参数中剔除。

该参数是动态参数：

SQL> alter system set asm\_diskgroups=DATA,FRA;

System altered.

SQL> show parameter asm\_diskgroups

NAME TYPE VALUE

------------------------------------ ----------- ------------------------------

asm\_diskgroups string DATA, FRA

12c之前，ASM最多支持配置63个磁盘组，12c以后，最多可以配置511个磁盘组。

#### ASM\_DISKSTRING

该参数非常关键，就是ASM启动挂载磁盘组时，磁盘所在的路径。ASM会在该参数中的路径逐一的找，如果其中某一个磁盘在俩个路径中都存在，那么该磁盘只会被挂载一次。

路径可以使用通配符，比如？代表一个字符，\*代表一串字符：

SQL> show parameter diskstring

NAME TYPE VALUE

------------------------------------ ----------- ------------------------------

asm\_diskstring string AFD:\*, /dev/asmdisk\*

#### ASM\_POWER\_LIMIT

该值为0，则磁盘组不会自动进行rebalance。该值越大，disk rebalance越快，当然消耗的IO也就越大。

如果COMPATIBLE.ASM的值为11.2.0.2或者更高，那么最大值为1024。如果是11.2.0.2之前，则为11。

SQL> show parameter ASM\_POWER\_LIMIT

NAME TYPE VALUE

------------------------------------ ----------- ------------------------------

asm\_power\_limit integer 1

#### ASM\_PREFERRED\_READ\_FAILURE\_GROUPS

该值将被遗弃。

#### DIAGNOSTIC\_DEST

这个跟dabase instance也是一样的，默认是$ORACLE\_BASE：

SQL> show parameter DIAGNOSTIC\_DEST

NAME TYPE VALUE

------------------------------------ ----------- ------------------------------

diagnostic\_dest string /u01/app/grid

也可以通过如下方式参看各种diag的路径：

SQL> select name,value from v$diag\_info;

NAME VALUE

---------------------- ----------------------------------------------------------------

Diag Enabled TRUE

ADR Base /u01/app/grid

ADR Home /u01/app/grid/diag/asm/+asm/+ASM

Diag Trace /u01/app/grid/diag/asm/+asm/+ASM/trace

Diag Alert /u01/app/grid/diag/asm/+asm/+ASM/alert

Diag Incident /u01/app/grid/diag/asm/+asm/+ASM/incident

Diag Cdump /u01/app/grid/diag/asm/+asm/+ASM/cdump

Health Monitor /u01/app/grid/diag/asm/+asm/+ASM/hm

Default Trace File /u01/app/grid/diag/asm/+asm/+ASM/trace/+ASM\_ora\_75904.trc

Active Problem Count 0

Active Incident Count 0

11 rows selected.

重点关注的是asm告警日志的位置：

[grid@yanxin trace]$ ls /u01/app/grid/diag/asm/+asm/+ASM/trace/alert\*

/u01/app/grid/diag/asm/+asm/+ASM/trace/alert\_+ASM.log

注意，database的alert日志所在路径是diag/rdbms，而ASM的则是diag/asm。

#### PROCESSES

Process的配置公式如下，其中n是连接到asm实例的database实例的个数。

For n < 10, PROCESSES = 50\*n + 50

For n >= 10, PROCESSES = 10\*n + 450

默认值是360：

SQL> show parameter process

NAME TYPE VALUE

------------------------------------ ----------- ------------------------------

processes integer 360

#### INSTANCE\_TYPE

该参数就三个值：asm、rdbms、asmproxy（FLEX ASM）。

SQL> show parameter instance\_type

NAME TYPE VALUE

------------------------------------ ----------- ------------------------------

instance\_type string ASM

当然database的就是rdbms。

#### REMOTE\_LOGIN\_PASSWORDFILE

跟database instance一样。决定访问asm实例时是否检查密码文件。

SQL> show parameter REMOTE\_LOGIN\_PASSWORDFILE

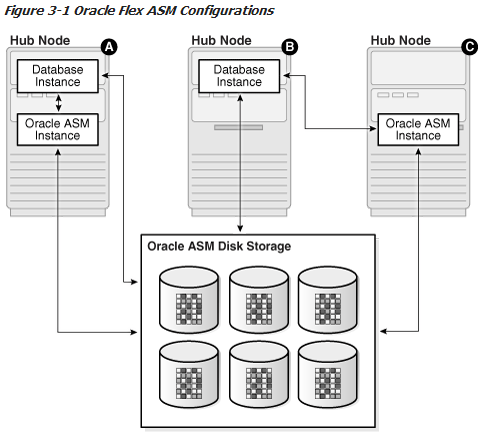
NAME TYPE VALUE

------------------------------------ ----------- ------------------------------

remote\_login\_passwordfile string EXCLUSIVE

## 管理ASM实例

### Flex ASM



### Converting to Oracle Flex ASM

### GI for a Standalone Server

当使用OUI安装，选择GI for standalone sever，就会同时包含Oracle asm和 oracle restart。

当发生故障时，Oracle Restart可以自动重启该服务器上的单实例（non-clustered）数据库、asm实例、监听、服务和其他的进程。

使用SRVCTL可以添加组件（比如ASM实例）到Oracle Restart，也可以移除组件，或者警用Oracle Restart。

### 启动ASM实例

#### Startup

首先配置好ORACLE\_SID和ORACLE\_HOME，然后使用sqlplus / as sysasm连接到ASM实例，然后执行startup。

Startup有如下几个选项：

1. Force：强制重启，相当于先shutdown abort然后再startup
2. Open或者mount：这是默认选项，也就是不用指定，效果是启动ASM实例，并且将磁盘组全部mount。相当于startup nomount+ALTER DISKGROUP ALL MOUNT
3. Nomount：只启动ASM实例，但是不挂载磁盘组
4. RESTRICT：可以同前面几个选项合用，再restrict mode，磁盘组将不能被数据库实例访问。同时磁盘组也不能被别的asm实例mount，只能被本实例mount。用途就是为了修复一些配置问题。  
   该模式下也不能使用alter system …，但是可以进行磁盘组的rebalance。

下面我们来看一下实验：

[grid@yanxin ~]$ sqlplus / as sysasm

SQL\*Plus: Release 12.2.0.1.0 Production on Wed Oct 10 18:25:35 2018

Copyright (c) 1982, 2016, Oracle. All rights reserved.

Connected to:

Oracle Database 12c Enterprise Edition Release 12.2.0.1.0 - 64bit Production

SQL> startup

ASM instance started

Total System Global Area 1140850688 bytes

Fixed Size 8629704 bytes

Variable Size 1107055160 bytes

ASM Cache 25165824 bytes

ASM diskgroups mounted

ASM diskgroups volume enabled

#### Startup时的磁盘组挂载

在startup时，ASM会挂载如下的磁盘组：

1. ASM\_DISKGROUPS中记录的磁盘组
2. ASM实例的SPFILE所在的磁盘组
3. Cluster Synchronization Services (CSS)用来放voting files的磁盘组
4. OCR所在的磁盘组

如果某些磁盘组都不符合上述的任一条件，则ASM在启动的时候不会自动挂载它们，需要手动执行alter diskgroup xx mount来进行挂载。

### 关闭ASM实例

Shutdown命令，也是分normal、immediate/transactional、abort。跟database instance关闭是一样的。

Normal就是等待所有连接到asm实例的user断开连接，才关闭ASM实例，如果有database instance在连接asm实例，则会返回error。

Immediate就是不用等连接到asm实例的user自主断开连接，但是如果有database instance在连接，同样返回error。

因为ASM没有事务这一说，所以transactional跟immediate效果是一样的。

Abort的效果当然就是强制关闭asm实例，带来的后果是，如果还有database instance连接到ASM实例，则database也会被shutdown abort。

我们一般当然是使用shutdown immediate。

这个过程会卸载磁盘组，所以，一定要先将数据库实例关闭、然后如果有ADVM，也要先umount。

如果ocr和voting file也在磁盘组上，那么asm实例停掉，并不会卸载ocr和voting file所在的磁盘组，只有当集群关闭时，才会dismount这些磁盘组。

下面我们简单看一下这个过程：

[grid@yanxin ~]$ sqlplus / as sysasm

SQL\*Plus: Release 12.2.0.1.0 Production on Wed Oct 10 22:02:12 2018

Copyright (c) 1982, 2016, Oracle. All rights reserved.

Connected to:

Oracle Database 12c Enterprise Edition Release 12.2.0.1.0 - 64bit Production

SQL> shutdown immediate

ASM diskgroups volume disabled

ASM diskgroups dismounted

ASM instance shutdown

### Server Control Utility (SRVCTL)

可以用srvctl来配置管理ASM实例和磁盘组。

[grid@yanxin ~]$ srvctl -help

Usage: srvctl add asm [-listener <lsnr\_name>] [-spfile <spfile>] [-pwfile <password\_file\_path>] [-diskstring <asm\_diskstring>]

Usage: srvctl config asm [-all]

Usage: srvctl start asm [-startoption <start\_options>]

Usage: srvctl stop asm [-stopoption <stop\_options>] [-force]

Usage: srvctl status asm [-all] [-verbose]

Usage: srvctl enable asm

Usage: srvctl disable asm

Usage: srvctl modify asm [-listener <lsnr\_name>] [-spfile <spfile>] [-pwfile <password\_file\_path>] [-diskstring <asm\_diskstring>]

Usage: srvctl remove asm [-force]

Usage: srvctl getenv asm [-envs "<name>[,...]"]

Usage: srvctl setenv asm {-envs "<name>=<val>[,...]" | -env "<name>=<value>"}

Usage: srvctl unsetenv asm -envs "<name>[,...]"

Usage: srvctl start diskgroup -diskgroup <dg\_name>

Usage: srvctl stop diskgroup -diskgroup <dg\_name> [-force]

Usage: srvctl status diskgroup -diskgroup <dg\_name> [-all] [-verbose]

Usage: srvctl enable diskgroup -diskgroup <dg\_name>

Usage: srvctl disable diskgroup -diskgroup <dg\_name>

Usage: srvctl remove diskgroup -diskgroup <dg\_name> [-force]

我们来启动一下asm：

[grid@yanxin ~]$ srvctl start asm

### Out of Place Upgrades

目前要想升级GI，只支持out of place，不支持in-place，也就是换个目录。比如说原来的11g的Gi的$ORACLE\_HOME是/u01/app/11.2.0/grid，那么升级为12c，就不能再使用这个$ORACLE\_HOME了。

### Active Session History Sampling

ASM也有自己的ASH，可以通过视图V$ACTIVE\_SESSION\_HISTORY进行查询。当然是在ASM实例中进行查询。

## Authentication for Accessing

因为ASM没有数据字典，所以只能通过三种系统权限连接ASM实例：sysasm、sysdba、sysoper。

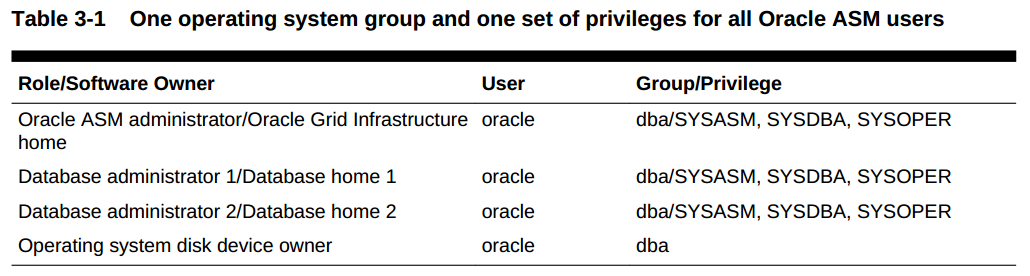
而连接ASM实例的方式跟连接数据库实例是一样的，也分为：

1. Local connection using operating system authentication：本地连接操作系统认证
2. Local connection using password authentication：本地连接密码认证
3. Remote connection by way of Oracle Net Services using password authentication：远程使用oracle net service连接并且密码认证

ASM实例和数据库实例必须具有对asm磁盘组的完全现同的（identical）读写权限。

### Using Separate Operating System Groups

可以只使用一个操作系统用户组，如下：



这个我们就不赘述了，一般情况下我们是使用多个operating system groups，而这些group分别被赋予了不同的管理权限：

1. OSASM group：  
   该组被赋予SYSASM权限，是管理ASM的最高权限，启动关闭啥都能干。  
   这样的组的例子是：asmadmin
2. OSDBA for Oracle ASM group：  
   该组被赋予SYSDBA权限，是OSASM组权限的子集，可以访问asm实例的数据。但不能进行管理操作，比如启动、关闭、mount、dismount等等的操作。但是该权限可以Create and delete files, aliases, directories, and templates。  
   这样的组的例子是：asmdba
3. OSOPER for Oracle ASM group：  
   该组被赋予SYSOPER权限，也是OSASM组权限的子集，可以进行startup、shutdown ASM实例和mount、dismount、check磁盘组的操作  
   这样的组的例子是：asmoper

我们这里看一下sysoper跟sysdba的区别，其实sysasm就是等于sysdba+sysoper。

[grid@yanxin ~]$ sqlplus / as sysoper

SQL> select count(1) from v$active\_session\_history;

select count(1) from v$active\_session\_history

\*

ERROR at line 1:

ORA-01219: database or pluggable database not open: queries allowed on fixed

tables or views only

SQL> exit

[grid@yanxin ~]$ sqlplus / as sysdba

SQL> select count(1) from v$active\_session\_history;

COUNT(1)

----------

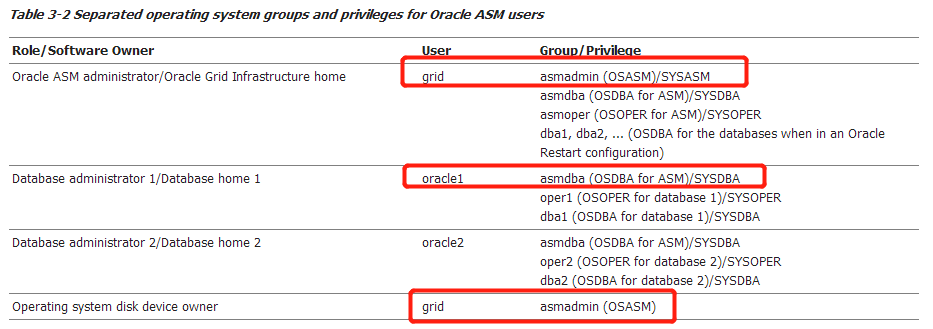
456

而我们知道，我们一般来说，GI+ASM是grid用户，Database是oracle用户。那么它们所属的组就有讲究了。

磁盘组的磁盘的owner必须得跟ASM software的owner是一致的，所以磁盘的所属是grid:asmadmin。

Database想要访问asm实例的数据，比如说要访问磁盘组，那oracle用户就必须得从属于ASM的OSDBA组。所以oracle用户必须得属于asmdba组。

如下是官文上的一个表格示例：



### Operating System Authentication

说白了就是：

sqlplus / as sysasm

sqlplus / as sysdba

sqlplus / as sysoper

[grid@yanxin ~]$ asmcmd

[grid@yanxin ~]$ asmca

### Password File Authentication

密码文件认证，当然就要有密码文件。如果安装GI的时候选择使用ASM存储，则asmca在创建磁盘组的时候会同时为初始用户sys和syssnmp创建密码文件。

那么后续如果想将一个user加入到密码文件中，可以使用create user和grant操作来实现：

[grid@yanxin ~]$ sqlplus / as sysasm

SQL> create user test identified by test;

User created.

SQL> grant sysasm to test;

Grant succeeded.

SQL> revoke sysasm from test;

Revoke succeeded.

SQL> drop user test;

User dropped.

注意，revoke操作并不能将用户从密码文件中移除，只有drop user才可以。

#### Shared Password File

12c的新特性，diskgroup的COMPATIBLE.ASM必须得是12.1以上。

12c以前密码文件不能放磁盘组上，12c以后能了。这样集群中各节点的数据库实例可以共享一个密码文件，ASM实例可以共享一个密码文件。

我们这里直接贴官文的示例。

创建密码文件仍然使用ORAPWD，而管理操作密码文件可以使用ASMCMD和SRVCTL。

为数据库创建密码文件，路径是asm的路径，并且指定dbuniquename：

$ orapwd file='+data/ORCL/orapwdb' dbuniquename='orcl'

Enter password for SYS:

为ASM创建密码文件：

$ orapwd file='+data/ASM/orapwasm' asm=y

Enter password for SYS:

使用之前的密码文件，创建到ASM磁盘组上：

$ orapwd input\_file='/oraclegrid/dbs/orapwasm' file='+data/ASM/orapwasm' asm=y

Enter password for SYS:

Asmcmd的命令有：

pwcreate pwdelete pwcopy pwmove pwset pwget

SRVCTL的相关命令有：

$ srvctl modify asm -pwfile location

$ srvctl modify database -db dbname -pwfile location

$ srvctl config asm

我们实验一下：

[oracle@yanxin dbs]$ orapwd input\_file='/u01/app/oracle/product/12.2.0/dbhome\_1/dbs/orapwPRODCDB' file='+data/prodcdb/orapwPRODCDB' dbuniquename='PRODCDB'

ASMCMD> pwget --dbuniquename PRODCDB

+DATA/prodcdb/orapwprodcdb

## Migrating a Database to Use Oracle ASM

就是将数据库文件从本地磁盘移到asm磁盘组上。

可以使用RMAN完成。这个后面详述。

# Administering Oracle ASM Disk Groups

## Managing Disk Group Attributes

磁盘组的属性是与磁盘组绑定的，无关ASM。有些是只能在磁盘组创建时进行指定，有些是只能在磁盘组创建后进行修改，当然还有一些是无论何时都可以指定和修改。

### Viewing and Setting

查询磁盘组的属性的方式有俩种：

1. 视图：V$ASM\_ATTRIBUTE
2. ASMCMD：lsattr

我们先看下视图V$ASM\_ATTRIBUTE：

SQL> desc V$ASM\_ATTRIBUTE

Name Null? Type

----------------------------------------- -------- ----------------------------

NAME VARCHAR2(256)

VALUE VARCHAR2(256)

GROUP\_NUMBER NUMBER

ATTRIBUTE\_INDEX NUMBER

ATTRIBUTE\_INCARNATION NUMBER

READ\_ONLY VARCHAR2(7)

SYSTEM\_CREATED VARCHAR2(7)

CON\_ID NUMBER

其中，read\_only列如果是Y，就表示该属性只能在磁盘组创建时指定。Group\_number可以通过视图v$asm\_diskgroup来对应到磁盘组得名字：

SQL> select group\_number,name from v$asm\_diskgroup;

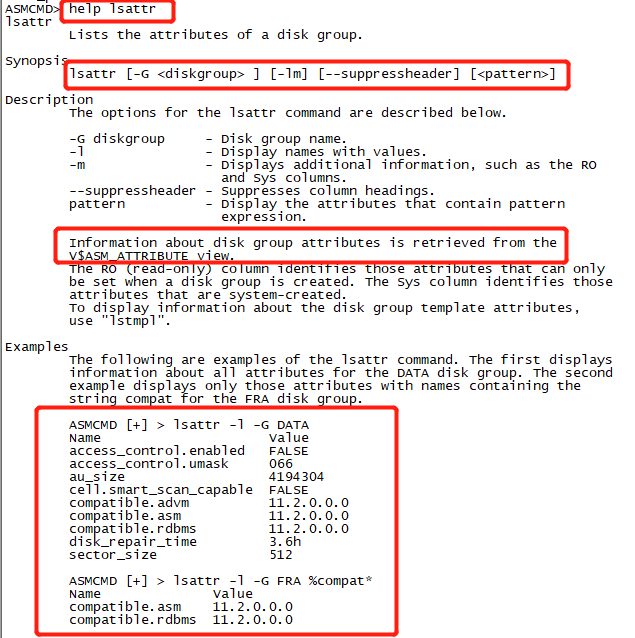
GROUP\_NUMBER NAME

------------ --------------------

1 DATA

2 FRA

然后再看下lsattr：



我们看到，lattr也是从v$ASM\_ATTRIBUTE中得来得信息。

[grid@yanxin ~]$ asmcmd

ASMCMD> lsattr -l -G DATA %compat\*

Name Value

compatible.advm 12.2.0.1.0

compatible.asm 12.2.0.1.0

compatible.rdbms 10.1.0.0.0

设置的方式有三种：

1. SQL：使用ALTER DISKGROUP 或者 CREATE DISKGROUP的ATTRIBUTE子句
2. ASMCMD：setattr和mkdg
3. ASMCA

### Attributes

下面我们来看一些主要得属性：

1. ACCESS\_CONTROL.ENABLED：设置oracle asm文件的访问控制的，只能通过alter diskgroup来设置。
2. ACCESS\_CONTROL.UMASK：同上
3. AU\_SIZE：设置allocation unit的大小，只能在create diskgroup时设置
4. CELL.SMART\_SCAN\_CAPABLE：Exadata（一体机）的参数
5. COMPATIBLE.ASM：
6. COMPATIBLE.RDBMS
7. COMPATIBLE.ADVM
8. CONTENT.CHECK
9. CONTENT.TYPE：  
   可以是data、recovery、system这三个值其中之一。  
   该参数一体机常用，redundancy必须是normal或者high才有效。  
   这个值说的是与自己最近的一个相邻的failure group的磁盘的距离，说白了就是自己与mirror们之间的间隔。Data是1，recovery是3，system是5。  
   为了防止一个区域的磁盘都坏了而导致一个磁盘组坏了。  
   该参数create或者alter diskgroup的能指定，但是如果使用alter diskgroup，那么必须手动执行rebalance，才能生效。
10. DISK\_REPAIR\_TIME：  
    磁盘被修复并被重新online的时间，超了这个时间，才可以删除磁盘  
    只有normal和high冗余的磁盘组才能设置该参数  
    该参数只能使用alter来修改
11. FAILGROUP\_REPAIR\_TIME：  
    同上，只不过是修复磁盘组的时间，默认是24h，但是如果配置了disk\_repair\_time，那么会覆盖该值。
12. IDP.BOUNDARY and IDP.TYPE：一体机的参数
13. PHYS\_META\_REPLICATED：
14. SECTOR\_SIZE
15. STORAGE.TYPE
16. THIN\_PROVISIONED

## Creating Disk Groups

### ASMCA创建磁盘组

见后面第7章ASMCA部分。

### SQLPLUS创建磁盘组。

#### 概念

创建磁盘组的SQL为：CREATE DISKGROUP。

我们需要指定如下内容：

1. diskgroup的名字：  
   这个名字时唯一的，指定的时候无所谓大小写，磁盘组建立时，最终会将该名字存为大写。
2. redundancy level：  
   NORMAL、HIGH、FLEX、EXTERNAL  
   当磁盘组的冗余为normal或者high的时候，可以使用alter diskgroup的CONVERT REDUNDANCY TO FLEX子句migrate为FLEX REDUNDANCY。前提必须是磁盘组至少有3个failure group，并且必须在mount restricted下进行migrate。  
   external磁盘组不能转成flex冗余。  
   如果想normal、high、external这些互相转，那就必须创建新的合适的redundancy磁盘组，然后把文件移过去。
3. 指定disk：  
   可以使用通配符指定一个路径。  
   可以使用NAME子句为disk指定asm中的名称，如果不指定，则默认使用系统分配的名称
4. 指定分配failure group（可选）：  
   默认是1个磁盘一个failure group。为了维护space balance和mirror data的均匀分布，Failure group们的大小必须相同
5. Failure group的类型（可选）：  
   regular和quorum
6. Diskgroup的其他一些属性：  
   比如compatibility、AU size

一个磁盘组中的所有磁盘大小必须一致，如果磁盘大小本身不一致，可以使用size子句，让每个磁盘用于构成磁盘组的部分大小一致。

前面说过可以使用NAME子句来指定磁盘的名字。ASMCMD中的stamp命令可以在create diskgroup使用磁盘们之前provision这些磁盘们（没弄明白）。如下是示例：

ASMCMD [+] > stamp --dscstr "/dev/sd\*" --site "SALTLAKE-CENTER" --failgroup "fg1" --disk "DISK1"

当从一个磁盘组drop一个磁盘时，如果是使用了force，那么再次将该磁盘添加到新的磁盘组时，也需要使用force。如果drop的时候没有使用force（即默认的noforce），那么添加的时候也不用使用force。

当使用force添加一个磁盘到新的磁盘组时，该磁盘所在的原磁盘组不能处于mount状态。

如果使用的是spfile，在创建了磁盘组之后，会将新的磁盘组自动添加到asm\_diskgroups中。如果是pfile，而且磁盘组又不满足前面提到的自动挂起的条件，我们又想让磁盘组在asm实例启动的时候自动挂起，则我们需要手动将新的磁盘组添加到asm\_diskgroups中。

#### 基本示例

首先，我们可以通过v$asm\_disk，看到目前配置的diskstring下有哪些disk：

SQL> show parameter diskstring

NAME TYPE VALUE

------------------------------------ ----------- ------------------------------

asm\_diskstring string AFD:\*, /dev/asmdisk\*

set lines 200

set pagesize 5000

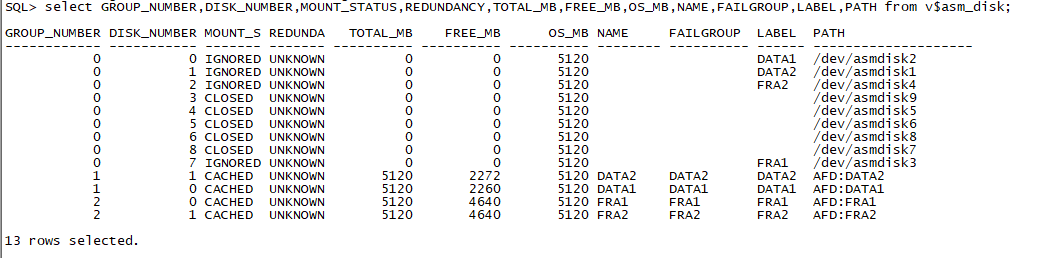
col path for a20

col label for a6

col faligroup for a11

col name for a8

SQL> select GROUP\_NUMBER,DISK\_NUMBER,MOUNT\_STATUS,REDUNDANCY,TOTAL\_MB,FREE\_MB,OS\_MB,NAME,FAILGROUP,LABEL,PATH from v$asm\_disk;



我们看到：

1. 我们有5块mount\_status是closed状态的磁盘，并且group\_number为0，那么这5块磁盘我们可以用来进行新的磁盘组的创建。
2. 如果磁盘还没创建total\_mb是没有值的，我们可以通过os\_mb看到，每一个磁盘都是5G。

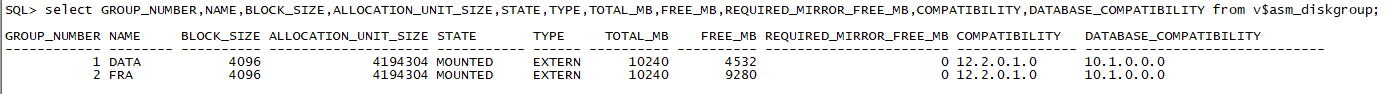
我们再来通过v$asm\_diskgroup看下磁盘组的信息：

col DATABASE\_COMPATIBILITY for a30

col COMPATIBILITY for a15

col FAILGROUP for a11

SQL> select GROUP\_NUMBER,NAME,BLOCK\_SIZE,ALLOCATION\_UNIT\_SIZE,STATE,TYPE,TOTAL\_MB,FREE\_MB,REQUIRED\_MIRROR\_FREE\_MB,COMPATIBILITY,DATABASE\_COMPATIBILITY from v$asm\_diskgroup;



我们可以看到：

1. 我们有俩次磁盘组都是mount的，且类型都是extern（外部冗余）。
2. Compatibility是12.2.0.1.0，database\_compatibility是10.1.0.0.0

现在我们来创建一个磁盘组，使用下面的sql：

CREATE DISKGROUP data02 NORMAL REDUNDANCY

FAILGROUP controller1 DISK

'/dev/asmdisk5' NAME disk5,

'/dev/asmdisk6' NAME disk6

FAILGROUP controller2 DISK

'/dev/asmdisk7' NAME disk7,

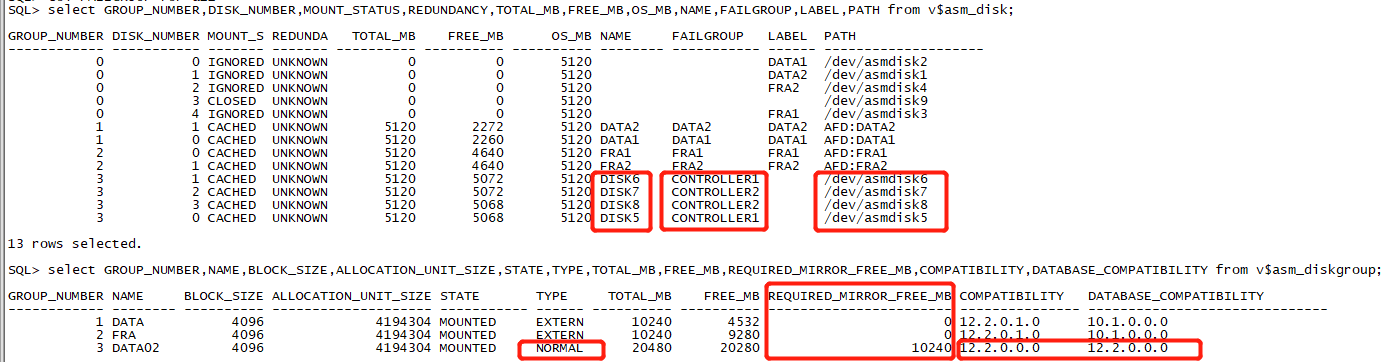
'/dev/asmdisk8' NAME disk8

ATTRIBUTE 'au\_size'='4M',

'compatible.asm' = '12.2',

'compatible.rdbms' = '12.2';

然后我们再来看上面俩个查询：



我们跟之前的进行一下比对，可以看出：

1. 由于我们指定了NAME子句，现在磁盘的name便不是系统指定的了。
2. 因为我们没有指定使用AFD，所以，我们看到路径就是磁盘本身的路径
3. Failgroup列说明了6和5是一个failgroup，7和8是一个failgroup
4. 我们看到磁盘组的type现在是normal，required\_mirror\_mb正好是磁盘组total\_mb的一半。
5. 由于我们指定了asm和数据库的compatibility，所以这里不再是之前的默认值

### 存储OCR和Voting file

强烈建议将OCR和voting file单独放在一个normal冗余的磁盘组。如果做不到，那就跟datafiles放在一个磁盘组。另外OCR的备份也可以放在asm磁盘组上。

有一个比较重要的概念就是quorum failure group。看一下官方示例：

CREATE DISKGROUP mgmt\_dg NORMAL REDUNDANCY

FAILGROUP fg1 DISK '/devices/diskg1'

FAILGROUP fg2 DISK '/devices/diskg2'

QUORUM FAILGROUP fg3 DISK '/devices/diskg3'

ATTRIBUTE 'compatible.asm' = '12.2.0.0.0';

Failgroup有俩种：regular和quorum。能放用户数据的，比如数据文件呀等等的failure group就是regular的。

所以着重说一下quorum failgroup的特点：

1. 不放user data
2. 存放有ASM metadata
3. 磁盘组必须是normal或者high冗余
4. 存放voting files的mirror copies
5. Quorum faliure group可以用来确定当磁盘组丢失一个或多个failure group时，还能否mount
6. 创建normal或high冗余磁盘组时，quorum failure group不算在内，因为它不能存放数据文件。说白了就是如果有俩个磁盘，本来是可以做成normal冗余，但如果指定其中一个是quorum failure group，那么这俩个磁盘就不能创建normal冗余的磁盘组，想要创建至少需要再加一块disk。

## Altering Disk Groups

Alter diskgroup会修改磁盘组的配置。该操作有如下几个需要注意的事项：

1. 只要修改磁盘组配置，就要进行rebalance。所以为了减少rebalance的次数，最好将要做的操作和改变放在一条alter diskgroup指令中完成。
2. 执行alter diskgroup指令会立马返回执行结果，而不会等待操作完成，事实上只有当rebalance结束才是真正的操作完成。这个过程可以通过v$asm\_operation进行监控和观察。
3. 可以加上REBALANCE WAIT子句，效果就是直到rebalance完成才返回执行结果。这个对于脚本来说很有用，避免磁盘组还在rebalance就开始进行脚本的下一步，可能导致脚本失败。
4. 如果是加了REBALANCE WAIT子句，那么在rebalance的时候就要等待，如果不想等待，可以直接使用CTRL+C，会返回ORA-01013: user requested cancel of current operation报错。该操作什么都不会中断，只是将界面交换用户，rebalance依然在后台异步继续进行。
5. 如果想要调整rebalance的速度，可以使用REBALANCE POWER子句，指定power，覆盖初始化参数中的配置。

### 管理ADVM

使用ADVM需要COMPATIBLE.ASM and COMPATIBLE.ADVM必须为11.2以上。

直接看官文示例：

SQL> ALTER DISKGROUP data ADD VOLUME volume1 SIZE 10G;

Diskgroup altered.

SQL> ALTER DISKGROUP data RESIZE VOLUME volume1 SIZE 15G;

Diskgroup altered.

SQL> ALTER DISKGROUP data DISABLE VOLUME volume1;

Diskgroup altered.

SQL> ALTER DISKGROUP data ENABLE VOLUME volume1;

Diskgroup altered.

SQL> ALTER DISKGROUP ALL DISABLE VOLUME ALL;

Diskgroup altered.

SQL> ALTER DISKGROUP data DROP VOLUME volume1;

Diskgroup altered.

如果ADVM上还有ACFS，就不能使用这种方式修改ADVM。需要使用acfsutil。

### 增加磁盘

主要就是注意，只有asm\_diskstring中配置的路径下的磁盘才能进行增加。

ALTER DISKGROUP data1 ADD DISK '/devices/diska\*';

ALTER DISKGROUP data1 ADD DISK

'/devices/diska5' NAME diska5,

'/devices/diska6' NAME diska6,

'/devices/diska7' NAME diska7,

'/devices/diska8' NAME diska8;

ALTER DISKGROUP data1 ADD DISK '/devices/diskd\*' EBALANCE POWER 5 WAIT;

ALTER DISKGROUP data2 ADD DISK '/devices/diskc3' FORCE;

使用force的情况，是将一个已经是其他磁盘组的磁盘强行加入另一个磁盘组，原磁盘组必须不能被mount。或者就是drop该disk的时候是用force进行drop的。

### 替换磁盘

就是磁盘损坏了，需要替换个磁盘来补位。Replace操作要比先drop再add更高效。

示例如下：

SQL> ALTER DISKGROUP data2 REPLACE DISK diskc7 WITH '/devices/diskc18' POWER 3;

### 重命名磁盘

如果要重命名磁盘，可以同时命名一个磁盘组下的多个磁盘，但是一定要在diskgroup在mount restricted的模式下才能进行重命名磁盘的操作。

SQL> ALTER DISKGROUP fra2 MOUNT RESTRICTED;

SQL> ALTER DISKGROUP fra2 RENAME DISK 'FRA1\_0001' TO 'FRA2\_0001', 'FRA1\_0002' TO 'FRA2\_0002';

还可以是使用ALTER DISKGROUP diskgroupname RENAME DISKS ALL重命名磁盘组下所有的磁盘。效果就是：名字不符合diskgroupname\_number规则的，都按该模板改，已经符合该模板的，就不改。

### 删除磁盘

Drop disk后，ASM会把disk中的extent移动剩余的磁盘中进行rebalance，如果剩余磁盘空间不够，该操作会失败。如果可以接受reduce reduancy，则可以使用force强制drop。

ALTER DISKGROUP data1 DROP DISK diska5;

所以为了保证足够的剩余空间，并且减少rebalance的时间，最好是将drop和add写在一条指令中执行，如下：

ALTER DISKGROUP data1 DROP DISK diska5

ADD FAILGROUP failgrp1 DISK '/devices/diska9' NAME diska9;

### Intelligent Data Placement

IDP在12.2被启用了，未来可能就不支持了，就不说了。列个示例：

ALTER DISKGROUP data MODIFY FILE '+data/orcl/datafile/users.259.679156903' ATTRIBUTE (HOT MIRRORHOT);

### Resizing Disks

因为磁盘组中磁盘大小必须一致，所以如果要resize磁盘，就必须resize磁盘组中的所有磁盘。可以使用如下sql：

ALTER DISKGROUP data1 RESIZE ALL SIZE 100G;

该sql的效果是：

1. 如果原来每个磁盘大小比100g小，且磁盘本身的大小大于等于100g，则ASM会把新的大小（100g）写入到磁盘头，并且立即将空间分配出来。  
   如果磁盘本身大小不足100g，则sql失败。
2. 如果原来每个磁盘大小比100g大，则磁盘组会进行rebalance，重新分配extent，如果剩余空间足够重新分配extent，则sql执行成功  
   如果剩余空间不足以relocate extents，则sql失败
3. 如果不指定size，则默认使用磁盘本身的大小。

### Undropping Disks

Sql如下：

ALTER DISKGROUP data1 UNDROP DISKS;

该操作只能还原还没进行完的drop diskgroup的操作。

如果drop diskgroup已经完成或者是使用force进行的drop，则该操作无效果。

### Manually Rebalancing

#### Rebalance的基本使用

前面说过，正常来说，修改了磁盘组配置，ASM就会自动进行rebalance。那么有俩种情况需要手动执行rebalance：

1. ASM\_POWER\_LIMIT设置为0，把rebalance禁了
2. 上一次rebalance失败了

Sql示例如下：

SQL> alter diskgroup data02 rebalance;

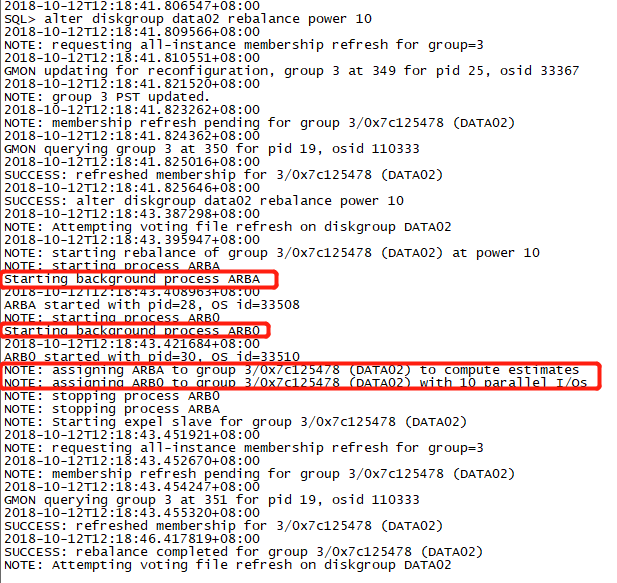
Diskgroup altered.

我们可以指定power子句来覆盖ASM\_POWER\_LIMIT的设置（默认值为1）。

SQL> alter diskgroup data02 rebalance power 10;

Diskgroup altered.

我们来看alert日志：



可以看到：

1. 相关的俩个进程为ARBA和ARB0
2. Power 10就是分配了10个IO并行

那么如果要终止正在进行的rebalance，可以使用：

SQL> alter diskgroup data02 rebalance power 0;

Diskgroup altered.

如果要中途修改rebalance的速率，可以使用：

ALTER DISKGROUP data2 REBALANCE MODIFY POWER 10; ----把当前power改成10

ALTER DISKGROUP data2 REBALANCE MODIFY POWER; ------把当前power改成ASM\_POWER\_LIMIT

当然，对于wait和nowait的使用，wait时的ctrl+c，都跟alter diskgroup中的使用是一样的。

#### Rebalance的各个阶段

还有可以指定with或者without的子句，来只执行rebalance或不执行rebalance的某几个阶段。默认是执行rebalance的所有阶段。

Rebalance的各个阶段如下：

1. RESTORE：restore阶段是必须经历的，不可以去除的，也就是说指定without restore也没用  
   restore有份3个操作：RESYNC、RESILVER、REBUILD  
   RESYNC：同步当前在线的disk中的陈旧的extent  
   RESILVER：一体机中才有的操作  
   REBUILD：只恢复forcing disks的redundancy。Forcing disks就是用force选项drop的磁盘们
2. BALANCE：restore所有磁盘组中磁盘的redundancy，并且balance所有disk上的extent
3. PREPARE：只对flex和extended冗余模式的磁盘组有效，COMPATIBLE.ASM必须是12.2以上
4. COMPACT：整理碎片并且压缩extents

如下是官文示例：

ALTER DISKGROUP data2 REBALANCE WITH BALANCE COMPACT;

ALTER DISKGROUP data3 REBALANCE WITHOUT BALANCE;

#### Rebalance其他的注意事项

还有如下注意事项需要记住：

1. 如果是RAC环境，diskgroup的rebalance操作只会在一个节点上执行。
2. 仍然是RAC环境，如果一个节点上连续执行几条、分别是对于不同磁盘组的rebalance，如果有可用节点，则会将后面的rebalance放到其他节点上去并行执行。如果没有，就一个等一个串行执行。  
   所以，建议不同磁盘组的rebalance，可以直接到不同的节点去分别执行。
3. 如果一个磁盘组的rebalance正在执行，那么仍然可以修改该磁盘组的配置，rebalance会立即重新开始。
4. 可以通过V$ASM\_OPERATION来观察监控reblance操作。如果有阶段执行失败，则会在rebalance后，将ORA-59000记录在视图中。
5. Rebalance时进程的状态和过程，可以通过alert日志观测。

#### 调优rebalance

Power越大，rebalance越快，但是消耗的IO也越多。ASM尽量保证多少power，就开多少IO平行。每一个IO都需要消耗PGA来存放要被relocate的extent。所以power越大，对PGA的消耗也越大。

可以使用explain work来预估rebalance的工作量，预估结果会存放在视图V$ASM\_ESTIMATE中。

如下是实验：

SQL> EXPLAIN WORK FOR ALTER DISKGROUP data02 drop disk disk8;

EXPLAIN WORK FOR ALTER DISKGROUP data02 drop disk disk8

\*

ERROR at line 1:

ORA-15032: not all alterations performed

ORA-15411: Failure groups in disk group DATA02 have different number of disks.

SQL> SELECT est\_work FROM V$ASM\_ESTIMATE;

no rows selected

SQL> EXPLAIN WORK FOR ALTER DISKGROUP data02 drop disk disk8 force;

Explained.

SQL> select \* from V$ASM\_ESTIMATE;

GROUP\_NUMBER STATEMENT\_ID TIMESTAMP EST\_WORK CON\_ID

------------ --------------- --------- ---------- ----------

3 12-OCT-18 20 0

SQL> EXPLAIN WORK SET STATEMENT\_ID='online' FOR ALTER DISKGROUP data02 drop disk disk8 force;

Explained.

SQL> select \* from V$ASM\_ESTIMATE;

GROUP\_NUMBER STATEMENT\_ID TIMESTAMP EST\_WORK CON\_ID

------------ --------------- --------- ---------- ----------

3 12-OCT-18 20 0

3 online 12-OCT-18 20 0

我们可以看到如果该sql不能执行，会直接报错，并且不会进行预估。

预估结果中的est\_work是rebalance中要被移动的AU的数目的预估。

可以使用如下sql进行rebalance时的观测：

SQL> SELECT GROUP\_NUMBER, PASS, STATE FROM V$ASM\_OPERATION;

GROUP\_NUMBER PASS STAT

------------ --------- ----

2 RESYNC WAIT

2 REBALANCE WAIT

2 COMPACT WAIT

### Scrubbing

Sql示例如下：

SQL> ALTER DISKGROUP data SCRUB POWER LOW;

SQL> ALTER DISKGROUP data SCRUB FILE '+DATA/ORCL/DATAFILE/example.266.806582193' REPAIR POWER HIGH FORCE;

SQL> ALTER DISKGROUP data SCRUB DISK DATA\_0005 REPAIR POWER HIGH FORCE;

下面来解释一下各个子句：

1. 如果没有指定REPAIR，则指检查指定磁盘组或磁盘或文件的corruption并报告出来；  
   如果指定了REPAIR，则如果是normal或者high冗余，ASM会自动使用mirror修复检查出来的corruption。
2. POWER有4个值：AUTO，LOW，HIGH，MAX。默认是AUTO。就是根据系统目前的状况自动选择SCRUB的速率。
3. 同样可以使用wait和nowait子句
4. 如果使用force，则不管当前系统性能如何，立即进行SCRUB；  
   如果不使用force，操作会进入SCRUB等待队列，轮到它的时候再执行
5. 可以与rebalance同时使用，这样可以节省IO资源
6. SCRUB的过程中，可以通过V$ASM\_OPERATION观测。
7. 同样会把过程记录在alert日志中

## ASM Disk Discovery

当ASM实例去找disk的时候，一定是到asm\_diskstring中配置的路径中找。

那么ASM实例在什么时候回去找disk呢？

1. ASM实例启动初始化的时候。
2. 执行如下SQL：  
   ALTER DISKGROUP MOUNT  
   ALTER DISKGROUP ONLINE DISK  
   ALTER DISKGROUP...ADD DISK  
   CREATE DISKGROUP  
   ALTER DISKGROUP...RESIZE DISK
3. 查询ASM\_DISKGROUP和ASM\_DISK视图时
4. 使用ASMCMA和ASMCMD完成上述行为时

如果一个disk被ASM成功发现了，那么从v$asm\_disk就可以查到该disk。

v$asm\_disk的header\_status标明了该磁盘的状态：

1. MEMBER：已属于某磁盘组，如果要将该状态的磁盘加入到其他磁盘组，需要使用force
2. CANDIDATE：可以被用来加入新的磁盘组，不需要使用force
3. FORMER：之前属于某个磁盘组，现在已经不属于了，而且是被drop cleanly（drop干净的），说白了就是这样状态的磁盘也加入新的磁盘组也不用force
4. FOREIGN：通过force drop的，这样的磁盘再次加入新的磁盘组需要使用force

SQL> SELECT name, header\_status, path FROM V$ASM\_DISK;

NAME HEADER\_STATU PATH

------------------------------ ------------ ---------------

MEMBER /dev/asmdisk2

MEMBER /dev/asmdisk1

MEMBER /dev/asmdisk4

CANDIDATE /dev/asmdisk9

MEMBER /dev/asmdisk3

DATA2 MEMBER AFD:DATA2

DATA1 MEMBER AFD:DATA1

FRA1 MEMBER AFD:FRA1

FRA2 MEMBER AFD:FRA2

DISK6 MEMBER /dev/asmdisk6

DISK7 MEMBER /dev/asmdisk7

DISK8 MEMBER /dev/asmdisk8

DISK5 MEMBER /dev/asmdisk5

13 rows selected.

磁盘mount操作会失败的情况：

1. Multiple paths to the same disk：配置多路径的磁盘，需要在asm\_diskstring中配置不能发现同一个磁盘的多个路径，或者使用多路径软件给磁盘取个伪名（pseudo-device name）。否则磁盘组不能成功挂载。
2. Multiple Oracle ASM disks with the same disk header：当一次磁盘copy到另一个磁盘时，会导致俩个不同的磁盘拥有相同的disk header，这时磁盘组是无法成功挂载的。

精确的的设置asm\_diskstring中的路径的值提升磁盘发现的时间（这很好理解）。

## Managing Capacity in Disk Groups

我们先来看如下的查询：

SQL> SELECT name, type, total\_mb, free\_mb, required\_mirror\_free\_mb,usable\_file\_mb FROM V$ASM\_DISKGROUP;

NAME TYPE TOTAL\_MB FREE\_MB REQUIRED\_MIRROR\_FREE\_MB USABLE\_FILE\_MB

---------- ------ ---------- ---------- ----------------------- --------------

DATA EXTERN 10240 4532 0 4532

FRA EXTERN 10240 9280 0 9280

DATA02 NORMAL 20480 20280 10240 5020

对于空间的管理其实说的主要就是这几个列。External的就不说了，我们以normal为例，说一下各个列。

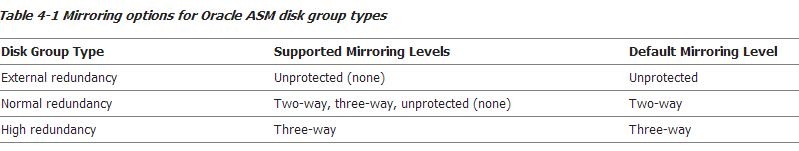
1. Total\_mb是该磁盘组的所有磁盘的大小之和。
2. Free\_mb是该磁盘组剩余的空间大小。有可能出现free\_mb有足够的空间，但是建文件报没有剩余空间的错误，这是因为磁盘组目前处于imbalance，其中某个磁盘空间耗尽了，还没有进行rebalance。
3. REQUIRED\_MIRROR\_FREE\_MB该值说明了如果有一个或多个磁盘顺坏，如果仍然想要保证目前的冗余模式所需要的空间的大小。  
   如果在发生磁盘损坏后，剩余空间不足以保证所有文件仍然保持原有的冗余模式，说白了就是，空间不够放一些文件的mirror了，这种情况就叫做redundancy lowered。  
   那么在v$asm\_file视图的列REDUNDANCY\_LOWERED会指示出该文件是否是redundancy lowered，不过该列目前已经被弃用，该列的值永远为U（unknown）。
4. usable\_file\_mb这个才是真正可用的空间大小  
   该值的计算为：(FREE\_MB - REQUIRED\_MIRROR\_FREE\_MB) / 2 （Normal冗余下）  
   所以该值很有可能为负值，如果该值为负值，那么就要添加磁盘来增大磁盘组空间。

## 冗余模式

### 三种模式

Flex这里不说。

三种冗余模式如下：



说白了就是磁盘组可以容忍多少磁盘损坏或丢失而不会dismount。External就是有一个磁盘坏了，那整个磁盘组都不能mount。

Normal冗余至少有俩个failure group，high至少有3个。

那normal举例，对于normal冗余的磁盘组，ASM会为一个extent产生俩份copy：a primary copy and a secondary copy。俩份copy放在会放在不同的failure group中。如果创建磁盘时没有指定falure group，则默认一个磁盘一个failure group。

磁盘失败的后会有俩种情况发生：

1. 失败的磁盘被dismount后被自动drop，磁盘组继续正常服务，因为有mirror可用。在disk drop之后，ASM会进行rebalance，将失败磁盘的数据恢复到现有磁盘，意思就是用现有的磁盘继续保持原本的冗余模式。
2. 失败的磁盘中的数据没有mirror，则整个磁盘组dismount

使用Failure group有如下建议和限制：

1. 一个磁盘只能属于一个磁盘组的一个failure group
2. 一个磁盘组的的Failure groups的大小最好是完全相等的
3. Normal冗余至少2个failure group，high冗余至少3个failuer group

### 如何选择冗余模式

ORACLE建议尽量将磁盘冗余使用raid来实现，磁盘组本身使用external冗余。

如果没有raid，则可以选择normal冗余。

### Oracle ASM Recovery from Read and Write I/O Errors

对于read error，即无法访问某个磁盘的上的某个extent。有可能是磁盘某些扇区（sector）可能发生损坏。Oracle ASM会尝试将该extent的好的copy拿来修复该损坏的部分。

对于write error，即当写一个extent到一个新的AU上时，无法写入，则ASM会从本磁盘分配一个新的AU来完成写入。一直这样尝试下去，如果仍然写入失败，则磁盘会被offline。

### Fast Mirror Resync

使用该特性，需要diskgroup的compatibility在11.1以上。

#### DISK\_REPAIR\_TIME

在一次短暂的（transient）磁盘访问失败后进行恢复redundancy是非常耗时的！那么fast mirror resync技术就是，在磁盘访问失败后，等待故障修复，如果实在修复不了，在磁盘被drop了以后，才会进行rebalance及restore redundancy。

这样的故障，包括磁盘坏了、线缆坏了、路径突然失灵等等，但凡导致磁盘暂时不能访问的都属于上述的“故障”。

而上述这个所谓“短暂的”其实就是一段预先指定的时间长度。

这个时间长度其实是磁盘组的一个属性：disk\_repair\_time。创建磁盘组时如果不指定，默认时3.6h。指定它可以使用m或者h，m是分钟，h是小时。

SQL> select name,value,group\_number from v$asm\_attribute where name ='disk\_repair\_time';

NAME VALUE GROUP\_NUMBER

------------------------------ ---------- ------------

disk\_repair\_time 3.6h 1

disk\_repair\_time 3.6h 2

disk\_repair\_time 3.6h 3

SQL> alter diskgroup data02 set attribute 'disk\_repair\_time'='4.8h';

Diskgroup altered.

SQL> select name,value,group\_number from v$asm\_attribute where name ='disk\_repair\_time';

NAME VALUE GROUP\_NUMBER

------------------------------ ---------- ------------

disk\_repair\_time 3.6h 1

disk\_repair\_time 3.6h 2

disk\_repair\_time 4.8h 3

意思很简单，就是在磁盘不可访问时，磁盘会被offline，然后ASM会开始计时，当offline的时间超过disk\_repair\_time后，磁盘被drop，并且ASM开始进行rebalance。

当磁盘被offline后，v$asm\_disk中的repair\_timer列就会显示还剩多长时间磁盘会被drop。

我们来实验一下：

SQL> alter diskgroup data02 offline disk disk8;

Diskgroup altered.

SQL> select group\_number,MOUNT\_STATUS,header\_status,name,failgroup,repair\_timer from v$asm\_disk;

GROUP\_NUMBER MOUNT\_S HEADER\_STATU NAME FAILGROUP REPAIR\_TIMER

------------ ------- ------------ -------- --------------- ------------

0 IGNORED MEMBER 0

0 IGNORED MEMBER 0

0 IGNORED MEMBER 0

0 CLOSED CANDIDATE 0

0 CLOSED MEMBER 0

0 IGNORED MEMBER 0

3 MISSING UNKNOWN DISK8 CONTROLLER2 17280

1 CACHED MEMBER DATA2 DATA2 0

1 CACHED MEMBER DATA1 DATA1 0

2 CACHED MEMBER FRA1 FRA1 0

2 CACHED MEMBER FRA2 FRA2 0

3 CACHED MEMBER DISK6 CONTROLLER1 0

3 CACHED MEMBER DISK7 CONTROLLER2 0

3 CACHED MEMBER DISK5 CONTROLLER1 0

14 rows selected.

SQL> select 17280/3600 from dual;

17280/3600

----------

4.8

SQL> select group\_number,MOUNT\_STATUS,header\_status,name,failgroup,repair\_timer from v$asm\_disk;

GROUP\_NUMBER MOUNT\_S HEADER\_STATU NAME FAILGROUP REPAIR\_TIMER

------------ ------- ------------ -------- --------------- ------------

0 IGNORED MEMBER 0

0 IGNORED MEMBER 0

0 IGNORED MEMBER 0

0 CLOSED CANDIDATE 0

0 IGNORED MEMBER 0

1 CACHED MEMBER DATA2 DATA2 0

1 CACHED MEMBER DATA1 DATA1 0

2 CACHED MEMBER FRA1 FRA1 0

2 CACHED MEMBER FRA2 FRA2 0

3 CACHED MEMBER DISK6 CONTROLLER1 0

3 CACHED MEMBER DISK7 CONTROLLER2 0

3 CACHED MEMBER DISK8 CONTROLLER2 17096

3 CACHED MEMBER DISK5 CONTROLLER1 0

13 rows selected.

SQL> /

GROUP\_NUMBER MOUNT\_S HEADER\_STATU NAME FAILGROUP REPAIR\_TIMER

------------ ------- ------------ -------- --------------- ------------

0 IGNORED MEMBER 0

0 IGNORED MEMBER 0

0 IGNORED MEMBER 0

0 CLOSED CANDIDATE 0

0 IGNORED MEMBER 0

1 CACHED MEMBER DATA2 DATA2 0

1 CACHED MEMBER DATA1 DATA1 0

2 CACHED MEMBER FRA1 FRA1 0

2 CACHED MEMBER FRA2 FRA2 0

3 CACHED MEMBER DISK6 CONTROLLER1 0

3 CACHED MEMBER DISK7 CONTROLLER2 0

3 CACHED MEMBER DISK8 CONTROLLER2 0

3 CACHED MEMBER DISK5 CONTROLLER1 0

13 rows selected.

我们看到，当磁盘被重新mount后，在磁盘组进行完fast mirror resync后，repair\_timer才被重置为0。

我们可以通过在offline时显式指定时间来覆盖该属性设定的指：

SQL> alter diskgroup data02 offline disk disk8 drop after 120m;

Diskgroup altered.

SQL> select group\_number,MOUNT\_STATUS,header\_status,name,failgroup,repair\_timer from v$asm\_disk where name='DISK8';

GROUP\_NUMBER MOUNT\_S HEADER\_STATU NAME FAILGROUP REPAIR\_TIMER

------------ ------- ------------ -------- --------------- ------------

3 MISSING UNKNOWN DISK8 CONTROLLER2 7200

如果对磁盘组修改该属性，则只会影响仍在先的磁盘及后续新加入的磁盘，对于已经offline的磁盘不会生效，也就是说如果一个磁盘offline了，你想通过修改disk\_repair\_time属性来延长修复时间是徒劳的，但是可以通过再次执行alter diskgroup xx offline disk xx drop after xxh来重定时间。

如果一个磁盘被offline后，是不能执行alter diskgroup drop disk的。如果确定磁盘无法修复，想要在disk\_repair\_time到达之前将磁盘组drop，可以再次执行alter diskgroup xx offline disk xx drop after 0m/0h。

如果在滚动升级（rolling upgrade）过程中一个磁盘被offline了，那么计时器会停止，直到滚动升级结束。磁盘仍是offline的，但是计时器开始计时。

在ALTER DISKGROUP ONLINE 时，可以通过V$ASM\_OPERATION查看当前disk resync，rebalance，rebuild和data compaction的进度。

SQL> alter diskgroup data02 offline disk disk8;

Diskgroup altered.

SQL> alter diskgroup data02 online disk disk8;

Diskgroup altered.

SQL> SELECT GROUP\_NUMBER, PASS, STATE FROM V$ASM\_OPERATION;

GROUP\_NUMBER PASS STAT

------------ --------- ----

3 COMPACT WAIT

3 REBALANCE WAIT

3 REBUILD WAIT

3 RESYNC RUN

SQL> /

no rows selected

注意：offline的时候，是不会使v$asm\_operation产生记录的。

#### FAILGROUP\_REPAIR\_TIME

跟disk\_repair\_time同理，只不过就是变成了留给failure group的进行repair的时间。默认是24h：

SQL> select name,value,group\_number from v$asm\_attribute where name ='failgroup\_repair\_time';

NAME VALUE GROUP\_NUMBER

--------------------- ---------- ------------

failgroup\_repair\_time 24.0h 1

failgroup\_repair\_time 24.0h 2

failgroup\_repair\_time 24.0h 3

#### 常用指令

You can use ALTER DISKGROUP to set the DISK\_REPAIR\_TIME attribute to a specified hour or minute value, such as 4.5 hours or 270 minutes. For example:

ALTER DISKGROUP data SET ATTRIBUTE 'disk\_repair\_time' = '4.5h'

ALTER DISKGROUP data SET ATTRIBUTE 'disk\_repair\_time' = '270m'

The following example takes disk DATA\_001 offline and drops it after five minutes.

ALTER DISKGROUP data OFFLINE DISK DATA\_001 DROP AFTER 5m;

The next example takes the disk DATA\_001 offline and drops it after the time period designated by DISK\_REPAIR\_TIME elapses:

ALTER DISKGROUP data OFFLINE DISK DATA\_001;

This example takes all of the disks in failure group FG2 offline and drops them after the time period designated by DISK\_REPAIR\_TIME elapses. If you used a DROP AFTER clause, then the disks would be dropped after the specified time:

ALTER DISKGROUP data OFFLINE DISKS IN FAILGROUP FG2;

The next example brings all of the disks in failure group FG2 online:

ALTER DISKGROUP data ONLINE DISKS IN FAILGROUP FG2;

This example brings only disk DATA\_001 online:

ALTER DISKGROUP data ONLINE DISK DATA\_001;

This example brings all of the disks in disk group DATA online:

ALTER DISKGROUP data ONLINE ALL;

### Even Read for Disk Groups

数据读会平均分配到磁盘组的所有磁盘上。该功能在12.1以上且非一体机的情况下，是自动打开的。

### Preferred Read Failure Groups

就是可以进行配置，使实例读自己节点最近的extent，这个extent甚至可能是secondary copy，而不是primary copy。

也就是说实例默认优先读取primary copy，不管primary copy离自己的节点远不远。而Preferred Read Failure Group的配置就是为了解决这个问题。

ASM\_PREFERRED\_READ\_FAILURE\_GROUPS就是来配置该特性的。

SQL> show parameter ASM\_PREFERRED\_READ\_FAILURE\_GROUPS

NAME TYPE VALUE

------------------------------------ ----------- ------------------------------

asm\_preferred\_read\_failure\_groups string

该特性主要对extended cluster有用，所以先不多说了。

## 对于Disk Groups的建议和限制

性能方面主要建议就是新的快的磁盘用来放数据文件（DATA），老的慢的用来做FRA。言外之意还有DATA和FRA最好区分开。

ASM存储方面有限制，这里只说目前12c以上的限制：

1. 最多511个磁盘组
2. 最多10000个ASM disk。注意是ASM disk最多10000个，并不是真实磁盘最多10000个。多个磁盘可以绑成LUN对应一个ASM disk。
3. 每个磁盘组最多放一百万个文件
4. 对于磁盘大小的限制：  
   4 PB maximum storage for each Oracle ASM disk with the allocation unit (AU) size equal to 1 MB  
   8 PB maximum storage for each Oracle ASM disk with the AU size equal to 2 MB  
   16 PB maximum storage for each Oracle ASM disk with the AU size equal to 4 MB  
   32 PB maximum storage for each Oracle ASM disk with the AU size equal to 8 MB  
   320 exabytes (EB) maximum for the storage system

如果COMPATIBLE.RDBMS是10.1（默认值），那么在AU是1M的情况下，可存放的文件大小限制会更小：

1. External redundancy: 16 TB
2. Normal redundancy: 5.8 TB
3. High redundancy: 3.9 TB

## Disk Group Compatibility

### 设置Compatibility

就是三个属性：

1. COMPATIBLE.ASM：必须是这3个属性中最大的！
2. COMPATIBLE.RDBMS
3. COMPATIBLE.ADVM

可以通过v$asm\_attribute查询：

SQL> select group\_number,name,value from v$asm\_attribute where name like 'compatible%';

GROUP\_NUMBER NAME VALUE

------------ ------------------------- -------------

1 compatible.asm 12.2.0.1.0

1 compatible.rdbms 10.1.0.0.0

1 compatible.advm 12.2.0.1.0

2 compatible.asm 12.2.0.1.0

2 compatible.rdbms 10.1.0.0.0

2 compatible.advm 12.2.0.1.0

3 compatible.asm 12.2.0.0.0

3 compatible.rdbms 12.2.0.0.0

8 rows selected.

可以通过asmcmd的setattr，或者alter diskgroup xx set attribute来重新设置：

SQL> ALTER DISKGROUP data set ATTRIBUTE 'compatible.rdbms' = '12.2';

Diskgroup altered.

SQL> select group\_number,name,value from v$asm\_attribute where name like 'compatible%' and group\_number=1;

GROUP\_NUMBER NAME VALUE

------------ ------------------------- -------------

1 compatible.asm 12.2.0.1.0

1 compatible.rdbms 12.2

1 compatible.advm 12.2.0.1.0

使用asmcmd：

[grid@yanxin ~]$ asmcmd

ASMCMD> setattr

usage: setattr -G <diskgroup> <attribute\_name> <attribute\_value>

help: help setattr

ASMCMD> setattr -G fra compatible.rdbms 12.2

ASMCMD> lsattr -l -G FRA %compat\*

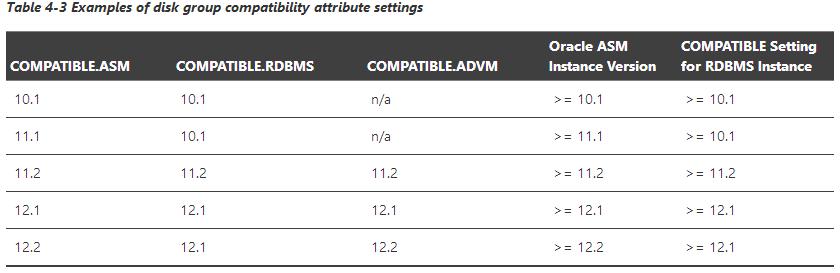
Name Value

compatible.advm 12.2.0.1.0

compatible.asm 12.2.0.1.0

compatible.rdbms 12.2

官文建议配置如下：



### 回退Compatibility

该修改不可逆！也就是说该大可以，改小不行：

SQL> ALTER DISKGROUP data set ATTRIBUTE 'compatible.rdbms' = '12.1';

ALTER DISKGROUP data set ATTRIBUTE 'compatible.rdbms' = '12.1'

\*

ERROR at line 1:

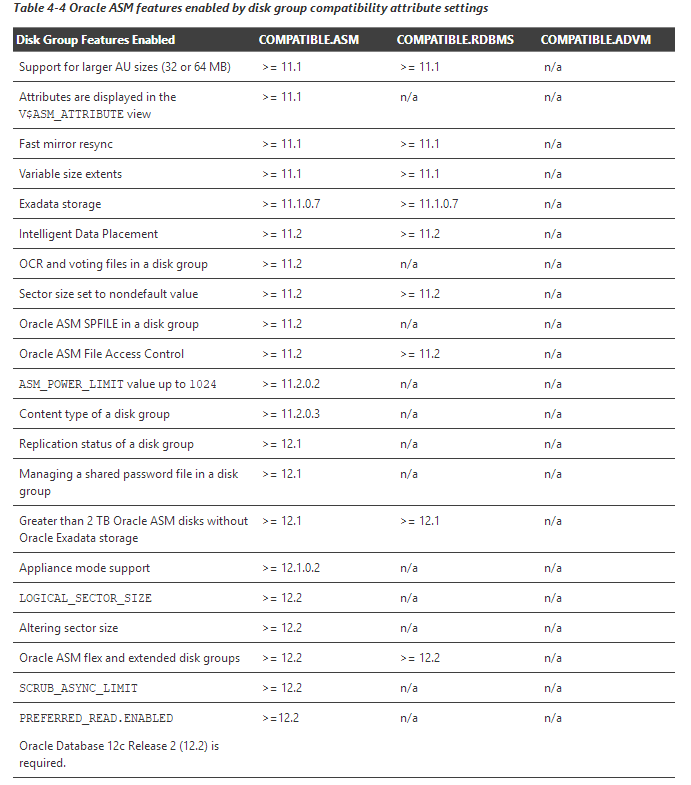
ORA-15032: not all alterations performed

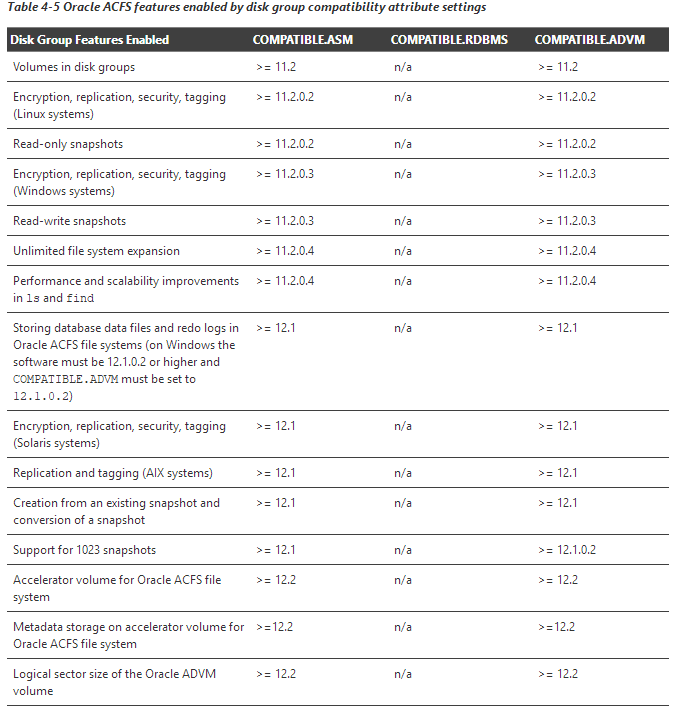
ORA-15242: could not set attribute compatible.rdbms

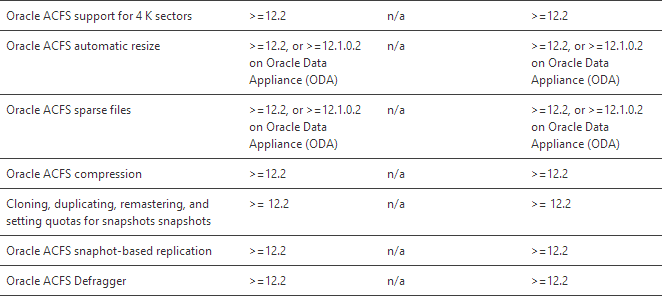
ORA-15244: new compatibility setting less than current [12.2.0.0.0]

如果一定要使用旧的compatibility，则必须重新创建磁盘组（属性为旧的compatibility），然后将文件move到新建的磁盘组上，最后把拥有较高的compatibility的磁盘组drop掉。

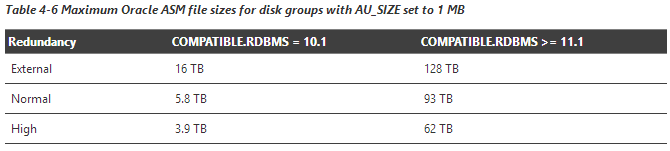
### Features Enabled By Disk Group Compatibility Attribute Settings

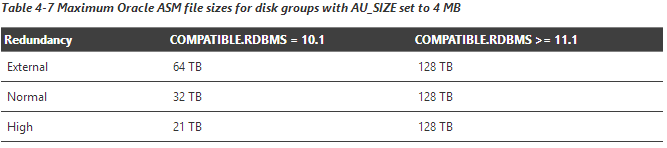






对于COMPATIBLE.RDBMS的设置和数据库可放的数据文件的大小关系如下：





## Managing Oracle ASM File Access Control

就是控制数据库实例对ASM磁盘组中文件的访问。

这个访问控制其实就是跟操作系统对文件的访问控制一模一样。只是把这一套搞到ASM实例中了。也有owner用户、同用户组用户、用户组外其他用户之分，对于3个分类，又分别分出了读写、只读、不可访问。

想要实现上面的控制，首先我们要了解磁盘组的俩个属性：

SQL> select group\_number,name,value from v$asm\_attribute where name like 'access\_control%';

GROUP\_NUMBER NAME VALUE

------------ ------------------------- -------------

1 access\_control.enabled FALSE

1 access\_control.umask 066

2 access\_control.enabled FALSE

2 access\_control.umask 066

3 access\_control.enabled FALSE

3 access\_control.umask 066

6 rows selected.

access\_control.enabled默认是false，只有为ture的时候，才能控制asm file的访问。所以这是总开关，这个如果未false，配置成啥都不好使。

access\_control.umask就是我们上面说的3种访问权限了，但是这个跟操作系统有点儿反，它的权限也由3个数字组成{0|2|6} {0|2|6} {0|2|6}。

只是这次就用求和了，很简单，0表示没限制，2表示限制写，6表示限制读写（让访问）。

默认是066，翻译过来就是，只有asm file的拥有者用户才能读写该文件，同组的其他用户、不同组的其他用户都不能访问该磁盘组的文件。

修改示例如下：

ALTER DISKGROUP data1 SET ATTRIBUTE 'access\_control.enabled' = 'true';

ALTER DISKGROUP data1 SET ATTRIBUTE 'access\_control.umask' = '026';

然后就是添加用户、用户组。

我们列出官文上的示例，当然，这些操作的效果同样也都可以使用asmcmd实现。

ALTER DISKGROUP ADD USERGROUP WITH MEMBER：

SQL> SELECT group\_number, os\_name FROM V$ASM\_USER;

GROUP\_NUMBER OS\_NAME

------------ ----------------------------------------------------------------

1 oracle1

1 oracle2

SQL> ALTER DISKGROUP data ADD USERGROUP 'test\_grp1' WITH MEMBER 'oracle1','oracle2';

ALTER DISKGROUP DROP USERGROUP：

SQL> ALTER DISKGROUP data DROP USERGROUP 'test\_grp1';

ALTER DISKGROUP MODIFY USERGROUP ADD MEMBER：

SQL> ALTER DISKGROUP data MODIFY USERGROUP 'test\_grp2' ADD MEMBER 'oracle2';

ALTER DISKGROUP MODIFY USERGROUP DROP MEMBER：

SQL> ALTER DISKGROUP data MODIFY USERGROUP 'test\_grp2' DROP MEMBER 'oracle2';

ALTER DISKGROUP ADD USER：

SQL> ALTER DISKGROUP DATA ADD USER 'oracle1';

ALTER DISKGROUP DROP USER：如果加了CASCADE则会连同该user所拥有的文件一同删除。

SQL> ALTER DISKGROUP DATA DROP USER 'oracle1';

ALTER DISKGROUP REPLACE USER：

SQL> ALTER DISKGROUP disk\_group REPLACE USER oracle1 WITH oracle2;

ALTER DISKGROUP SET PERMISSION：

SQL> ALTER DISKGROUP data SET PERMISSION OWNER=read write, GROUP=read only, OTHER=none FOR FILE '+data/controlfile.f';

ALTER DISKGROUP SET OWNERSHIP：

SQL> ALTER DISKGROUP data SET OWNERSHIP OWNER='oracle1', GROUP='test\_grp1' FOR FILE '+data/controlfile.f';

## Maintaining Disk Groups

### Mount/Dismount

首先，当磁盘组mount的时候，才会分配给磁盘组一个group\_number，也就是说这个group\_number在每次mount后是可能与之前不同的，因为没有一个永久的结构来记录group\_number。那么本次mount分配的group number可以通过v$asm\_xx查看：

SQL> select GROUP\_NUMBER,name,state from v$asm\_diskgroup;

GROUP\_NUMBER NAME STATE

------------ ------------------------- -----------

1 DATA MOUNTED

2 FRA MOUNTED

3 DATA02 MOUNTED

然后，说一下force选项。

Dismount force就不用说了，当数据文件是open状态时，diskgroup是不能dismount的，但是加了force就可以强制dismount。

那么mount force呢？当normal或者high冗余的磁盘组缺失磁盘时，直接mount会失败。如果磁盘组中的所有extent都至少有一份copy可以找到，那么可以使用mount force来强制挂载起来。

风险就是因为磁盘组空间不足，可能会出现reduce redundancy。

那如果存在完全缺失的extent，那mount force也没用。

当磁盘组被mount force起来后，损坏的磁盘依旧保持offline，同时计时器开始计时，达到disk\_repair\_time就会把磁盘永久删除。

Sql语句示例：

SQL> ALTER DISKGROUP ALL DISMOUNT;

SQL> ALTER DISKGROUP data1 MOUNT;

SQL> ALTER DISKGROUP data1 MOUNT FORCE;

### Checking the Internal Consistency of Disk Group Metadata

首先，该操作必须时磁盘组是mount的情况下才可以进行。

其次，该操作可以指定REPAIR | NOREPAIR子句，意思就是查到错误，ASM是自动修复还不修复，默认是NOREPAIR。

然后，会检查如下项目：

1. 磁盘的一致性
2. 所有file extent maps和allocation tables的一致性
3. alias metadata directory和file directory are linked correctly
4. alias directory tree is linked correctly
5. Oracle ASM metadata directories有没有不可访问的块

最后，检查的结果会记录在alert日志中。

实验如下：

SQL> ALTER DISKGROUP data CHECK ALL;

Diskgroup altered.

Alert日志中的记录如下：

SQL> ALTER DISKGROUP data CHECK ALL

2018-10-12T09:13:17.062684+08:00

NOTE: starting check of diskgroup DATA

2018-10-12T09:13:17.142550+08:00

GMON querying group 1 at 332 for pid 28, osid 19509

GMON checking disk 0 for group 1 at 333 for pid 28, osid 19509

GMON querying group 1 at 334 for pid 28, osid 19509

GMON checking disk 1 for group 1 at 335 for pid 28, osid 19509

2018-10-12T09:13:17.264620+08:00

SUCCESS: check of diskgroup DATA found no errors

2018-10-12T09:13:17.264715+08:00

SUCCESS: ALTER DISKGROUP data CHECK ALL

### Drop

删除磁盘组有如下需要注意的：

1. ASM实例必须启动，同时磁盘组必须mount且磁盘组中没有文件被访问。
2. 如果磁盘组有文件，就必须使用INCLUDING CONTENTS，否则会报错，这样设计是视图从语法上对误删除有文件的磁盘组做的一个保护措施。
3. 磁盘组被删除时，ASM会先dismount磁盘组。
4. 磁盘组被删除后，如果使用的是spfile，asm\_diskgroups参数会自动将对应磁盘组从参数中去除，否则，则需要手动去除。
5. 如果磁盘组不能mount起来，则需要通过FORCE选项来强制删除。原理是直接将磁盘头删除。
6. 如果使用FORCE选项，则ASM不会确认磁盘组中还有没有文件，会直接删除，所以慎用。
7. DROP DISKGROUP的操作会等待磁盘组被成功删除后才会返回结果。
8. 可以使用ASMCA进行删除磁盘组操作

示例如下：

SQL> DROP DISKGROUP data1;

SQL> DROP DISKGROUP data1 INCLUDING CONTENTS;

SQL> DROP DISKGROUP data1 FORCE;

### Rname

使用renamedg来进行修改磁盘组名。

该操作分俩个阶段：

1. 阶段一：生成一个给阶段二用的配置文件
2. 阶段二：使用阶段一生成的配置文件修改磁盘组名

如下是renamedg的语法：

[grid@yanxin ~]$ renamedg -help

Parsing parameters..

phase Phase to execute,

(phase=ONE|TWO|BOTH), default BOTH

dgname Diskgroup to be renamed

newdgname New name for the diskgroup

config intermediate config file

check just check-do not perform actual operation,

(check=TRUE/FALSE), default FALSE

confirm confirm before committing changes to disks,

(confirm=TRUE/FALSE), default FALSE

clean ignore errors,

(clean=TRUE/FALSE), default TRUE

asm\_diskstring ASM Diskstring (asm\_diskstring='discoverystring',

'discoverystring1' ...)

verbose verbose execution,

(verbose=TRUE|FALSE), default FALSE

keep\_voting\_files Voting file attribute,

(keep\_voting\_files=TRUE|FALSE), default FALSE

下面是示例：

第一个示例是使用phase的默认值both，直接一条指令完成俩个阶段。

$ renamedg dgname=fra1 newdgname=fra2 asm\_diskstring='/devices/disk\*' verbose=true

第二个示例是用俩条指令，分别完成俩个阶段：

$ renamedg phase=one dgname=fra1 newdgname=fra2 asm\_diskstring='/devices/disk\*' config=/tmp/fra2.conf verbose=true

$ renamedg phase=two dgname=fra1 newdgname=fra2 config=/tmp/fra2.conf verbose=true

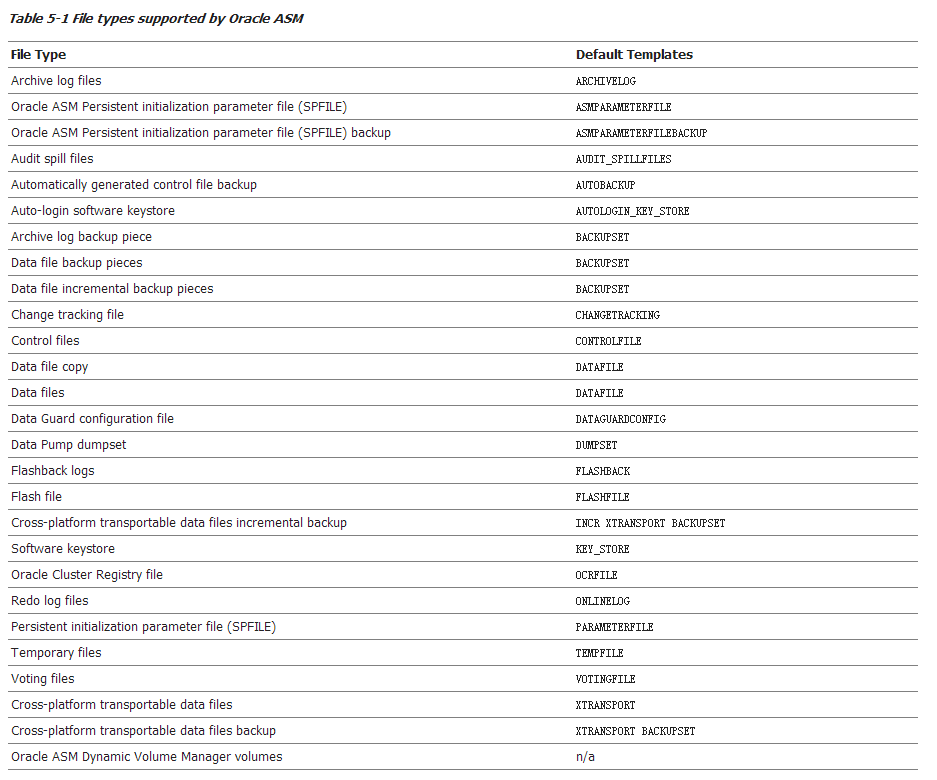
在完成renamedg后，老的资源并不会被自动删除，使用crsctl status res -t依然可以查到，需要使用srvctl手动删除。

然后还需要使用sqlplus将磁盘组下的磁盘名进行修改来匹配新的磁盘组名：

SQL> ALTER DISKGROUP fra2 RENAME DISKS ALL;

# Administering Oracle ASM Files, Directories, and Templates

## ASM支持的文件类型

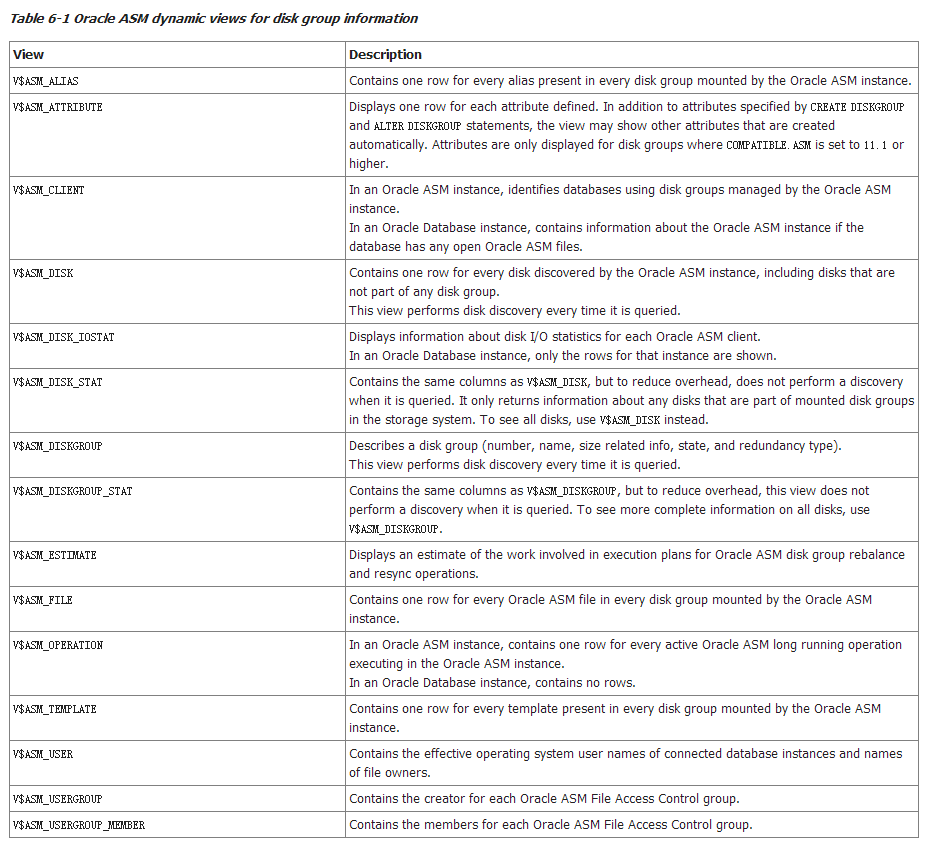


Trace files, audit files, alert logs, export files, tar files, core files这些管理类的文件不支持放在ASM上。但是可以放在ACFS上。

## ASM File Names

### Single File Creation Form

# 相关视图



# ASM Data Migration with RMAN

使用RMAN进行数据文件的ASM与其他存储设备之间的迁移方便快捷。而且，操作系统的cp命令并不能读写ASM磁盘的文件。

原来其实就是数据文件用backup as copy，做个image copy到asm磁盘组上，然后再switch to copy。

## 从ASM到磁盘

我们选一个数据文件来做实验：

SQL> select file\_name,file\_id,tablespace\_name,AUTOEXTENSIBLE from dba\_data\_files where file\_id=12;

FILE\_NAME FILE\_ID TABLESPACE\_NAME AUT

-------------------------------------------------------- ---------- -------------------- ---

+DATA/PRODCDB/77DD456794621D0FE053140CA8C06A4E/DATAFILE/ 12 YX YES

yx.277.989334975

使用RMAN备份12号文件的image copy到磁盘：

RMAN> backup as copy datafile 12 format '/home/oracle/oradata/yx.dbf';

Starting backup at 2018-10-12 15:50:59

using target database control file instead of recovery catalog

allocated channel: ORA\_DISK\_1

channel ORA\_DISK\_1: SID=746 device type=DISK

channel ORA\_DISK\_1: starting datafile copy

input datafile file number=00012 name=+DATA/PRODCDB/77DD456794621D0FE053140CA8C06A4E/DATAFILE/yx.277.989334975

output file name=/home/oracle/oradata/yx.dbf tag=TAG20181012T155101 RECID=1 STAMP=989337061

channel ORA\_DISK\_1: datafile copy complete, elapsed time: 00:00:01

Finished backup at 2018-10-12 15:51:02

然后switch datafile xx to copy：

RMAN> switch datafile 12 to copy;

RMAN-00571: ===========================================================

RMAN-00569: =============== ERROR MESSAGE STACK FOLLOWS ===============

RMAN-00571: ===========================================================

RMAN-03002: failure of switch to copy command at 10/12/2018 15:51:37

RMAN-06572: database is open and datafile 12 is not offline

我们看到，显然，datafile 12必须要offline，才能进行switch的操作。

我们这里先做些事务，然后再离线：

SQL> alter session set container=pdbprod1;

Session altered.

SQL> create user yx identified by yx default tablespace yx;

User created.

SQL> grant dba to yx;

Grant succeeded.

SQL> conn yx/yx@pdbprod1

Connected.

SQL> create table t1(id number);

Table created.

SQL> insert into t1 values(1);

1 row created.

SQL> commit;

Commit complete.

我们建了个用户，并且建了个表，还插了行数据。现在离线datafile 12：

SQL> alter database datafile 12 offline;

Database altered.

现在我们来再来switch：

RMAN> switch datafile 12 to copy;

datafile 12 switched to datafile copy "/home/oracle/oradata/yx.dbf"

现在成功了。

但是我们要想再online起来datafile 12需要进行实例恢复：

RMAN> recover datafile 12;

Starting recover at 2018-10-12 16:03:48

using channel ORA\_DISK\_1

starting media recovery

media recovery complete, elapsed time: 00:00:00

Finished recover at 2018-10-12 16:03:49

然后我们把datafile 12进行online：

SQL> alter database datafile 12 online;

Database altered.

我们来查一下刚才做的事务：

SQL> select file\_id,file\_name,tablespace\_name,online\_status,AUTOEXTENSIBLE from dba\_data\_files where file\_id=12;

FILE\_ID FILE\_NAME TABLESPACE\_NAME ONLINE\_ AUT

---------- ------------------------------ -------------------- ------- ---

12 /home/oracle/oradata/yx.dbf YX ONLINE YES

SQL> select \* from t1;

ID

----------

1

都没问题。

## 从磁盘到ASM

我们再做个事务：

SQL> insert into t1 values(2);

1 row created.

SQL> commit;

Commit complete.

SQL> select \* from t1;

ID

----------

1

2

我们可以还按上面的步骤backup as copy，然后switch。这次我们不这样，我们直接使用原来的数据文件。

先离线：

SQL> alter database datafile 12 offline;

Database altered.

然后使用alter database rename file，其实跟switch一个道理，改控制文件中数据文件记录的俩种不同的方式而已：

SQL> alter database rename file '/home/oracle/oradata/yx.dbf' to '+DATA/PRODCDB/77DD456794621D0FE053140CA8C06A4E/DATAFILE/yx.277.989334975';

Database altered.

然后就是recover、online、检查：

SQL> recover datafile 12;

Media recovery complete.

SQL> alter database datafile 12 online;

Database altered.

SQL> select \* from t1;

ID

----------

1

2

SQL> select file\_id,file\_name,tablespace\_name,online\_status,AUTOEXTENSIBLE from dba\_data\_files where file\_id=12;

FILE\_ID FILE\_NAME TABLESPACE\_NAME ONLINE\_ AUT

---------- ------------------------------ -------------------- ------- ---

12 +DATA/PRODCDB/77DD456794621D0F YX ONLINE YES

E053140CA8C06A4E/DATAFILE/yx.2

77.989334975

## 数据库整体迁移到ASM（含重建redo和standbylog脚本）

其实可以当作异机恢复来做。

官文给出的办法如下：

第一步把spfile迁移上ASM。

第二步把备库的日志应用停了，如果要迁移的数据库是备库的话：

SQL> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE CANCEL;

第三步做个0级copy，copy到ASM上：

RUN

{

ALLOCATE CHANNEL dev1 DEVICE TYPE DISK;

ALLOCATE CHANNEL dev2 DEVICE TYPE DISK;

ALLOCATE CHANNEL dev3 DEVICE TYPE DISK;

ALLOCATE CHANNEL dev4 DEVICE TYPE DISK;

BACKUP AS COPY INCREMENTAL LEVEL 0 DATABASE FORMAT '+DATA' TAG 'ORA\_ASM\_MIGRATION';

}

第四步做1级copy，开了块跟踪的会比较快一些：

RUN

{

ALLOCATE CHANNEL dev1 DEVICE TYPE DISK;

ALLOCATE CHANNEL dev2 DEVICE TYPE DISK;

ALLOCATE CHANNEL dev3 DEVICE TYPE DISK;

ALLOCATE CHANNEL dev4 DEVICE TYPE DISK;

BACKUP INCREMENTAL LEVEL 1 FOR RECOVER OF COPY WITH TAG 'ORA\_ASM\_MIGRATION' DATABASE;

}

第五步，开了block change tracking要关、开了闪回要关、有guaranteed point要删：

RMAN> SQL "ALTER DATABASE DISABLE BLOCK CHANGE TRACKING";

RMAN> SQL "ALTER DATABASE FLASHBACK OFF";

RMAN> SQL "DROP RESTORE POINT Q106";

第六步，关数据库：

RMAN> SHUTDOWN IMMEDIATE;

第七步，用ASM上的spfile把数据库nomount：

SQL> STARTUP FORCE NOMOUNT;

第八步，修改参数—配置OMF、配置FRA、修改

SQL> ALTER SYSTEM SET DB\_CREATE\_FILE\_DEST='+DATA' SID='\*';

SQL> ALTER SYSTEM SET DB\_RECOVERY\_FILE\_DEST\_SIZE=100G SID='\*';

SQL> ALTER SYSTEM SET DB\_RECOVERY\_FILE\_DEST='+FRA' SID='\*';

SQL> ALTER SYSTEM SET CONTROL\_FILES='+DATA','+FRA' SCOPE=SPFILE SID='\*';

第九步，重启数据库，恢复控制文件到ASM上，启动到mount：

RMAN> STARTUP FORCE NOMOUNT;

RMAN> RESTORE CONTROLFILE FROM 'original\_cf\_name';

RMAN> ALTER DATABASE MOUNT;

第十步，恢复数据库，先switch，再recover：

SWITCH DATABASE TO COPY;

RUN

{

ALLOCATE CHANNEL dev1 DEVICE TYPE DISK;

ALLOCATE CHANNEL dev2 DEVICE TYPE DISK;

ALLOCATE CHANNEL dev3 DEVICE TYPE DISK;

ALLOCATE CHANNEL dev4 DEVICE TYPE DISK;

RECOVER DATABASE;

}

第十一步，开数据库或者恢复备库日志应用

SQL> ALTER DATABASE OPEN;

或：

SQL> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE;

第十二步，重建temp表空间，建到ASM上：

SQL> ALTER DATABASE TEMPFILE 'tempfile\_name' DROP;

SQL> ALTER TABLESPACE temp\_tbs\_name ADD TEMPFILE;

第十三步，重建redolog或者standby log到ASM上：

SET SERVEROUTPUT ON;

DECLARE

CURSOR rlc IS

SELECT GROUP# GRP, THREAD# THR, BYTES, 'NO' SRL

FROM V$LOG

UNION

SELECT GROUP# GRP, THREAD# THR, BYTES, 'YES' SRL

FROM V$STANDBY\_LOG

ORDER BY 1;

stmt VARCHAR2(2048);

BEGIN

FOR rlcRec IN rlc LOOP

IF (rlcRec.srl = 'YES') THEN

stmt := 'ALTER DATABASE ADD STANDBY LOGFILE THREAD ' ||

rlcRec.thr || ' SIZE ' || rlcRec.bytes;

EXECUTE IMMEDIATE stmt;

stmt := 'ALTER DATABASE DROP STANDBY LOGFILE GROUP ' || rlcRec.grp;

EXECUTE IMMEDIATE stmt;

ELSE

stmt := 'ALTER DATABASE ADD LOGFILE THREAD ' ||

rlcRec.thr || ' SIZE ' || rlcRec.bytes;

EXECUTE IMMEDIATE stmt;

BEGIN

stmt := 'ALTER DATABASE DROP LOGFILE GROUP ' || rlcRec.grp;

DBMS\_OUTPUT.PUT\_LINE(stmt);

EXECUTE IMMEDIATE stmt;

EXCEPTION

WHEN OTHERS THEN

EXECUTE IMMEDIATE 'ALTER SYSTEM SWITCH LOGFILE';

EXECUTE IMMEDIATE 'ALTER SYSTEM CHECKPOINT GLOBAL';

EXECUTE IMMEDIATE stmt;

END;

END IF;

END LOOP;

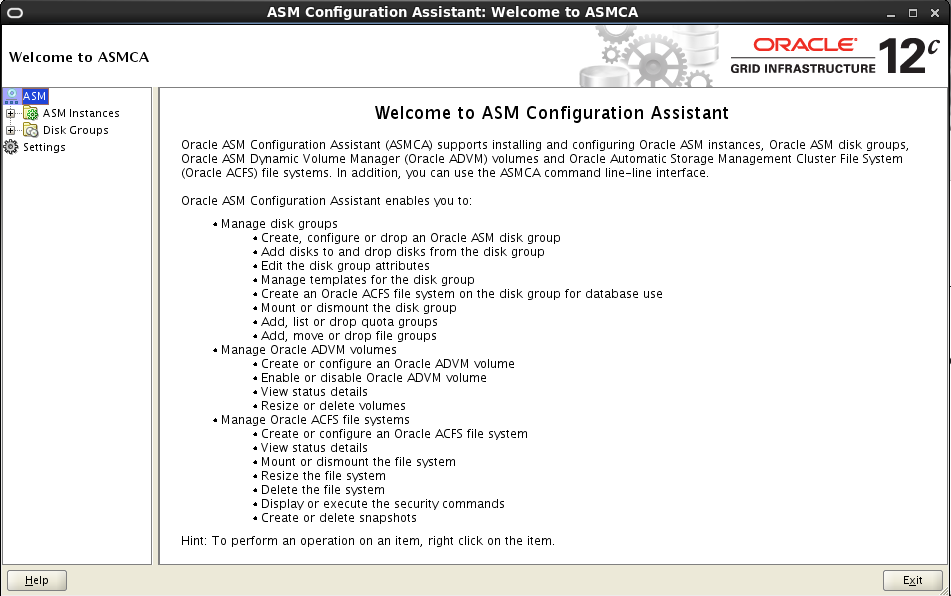
END;

/

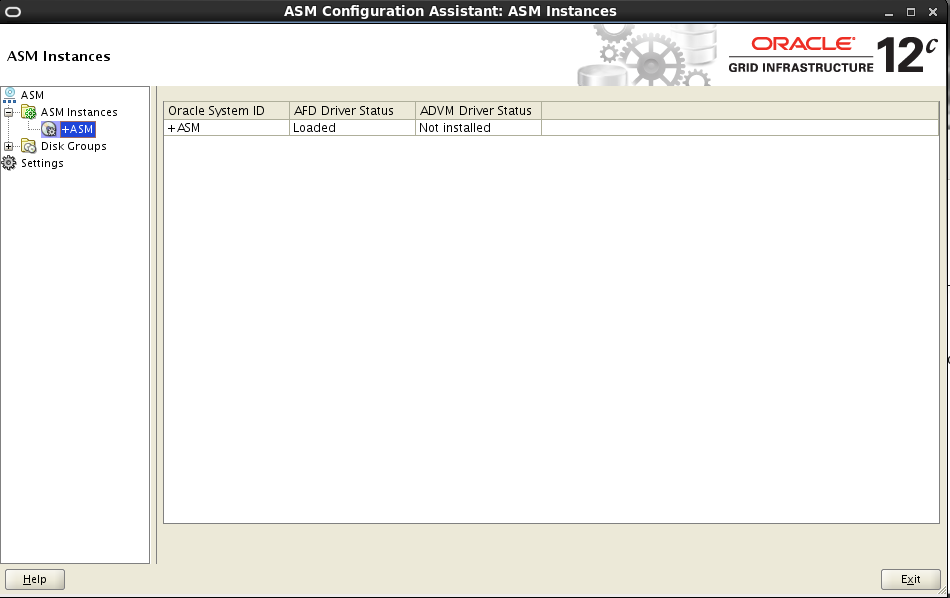
# ASMCA

进入asmca：

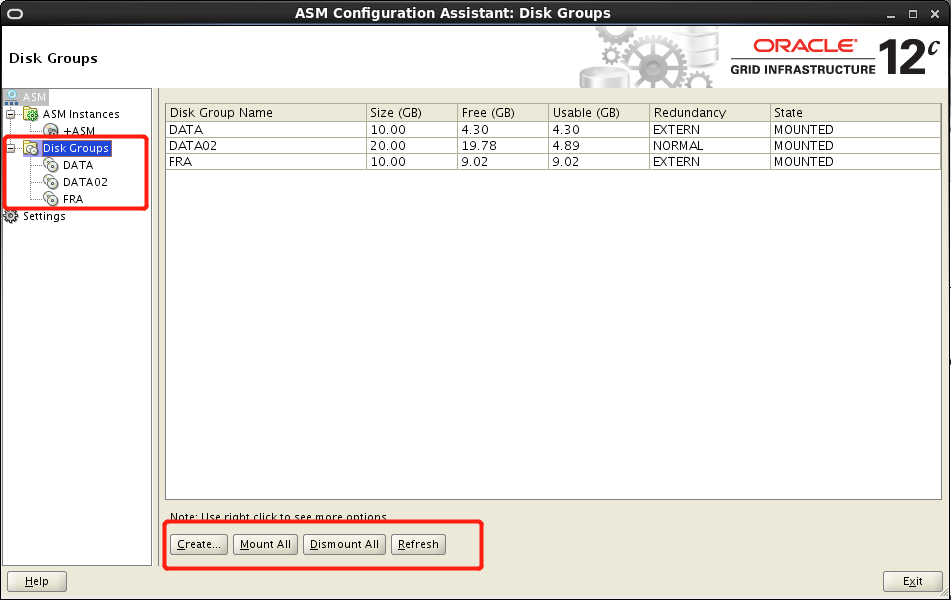
[grid@yanxin ~]$ asmca



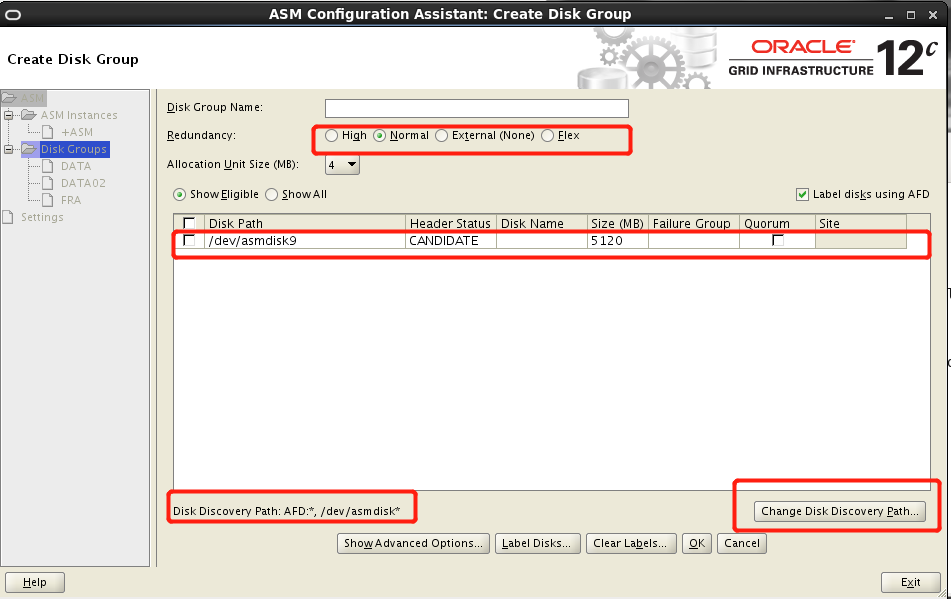
如下是ASM实例的情况：



如下是磁盘组的情况：

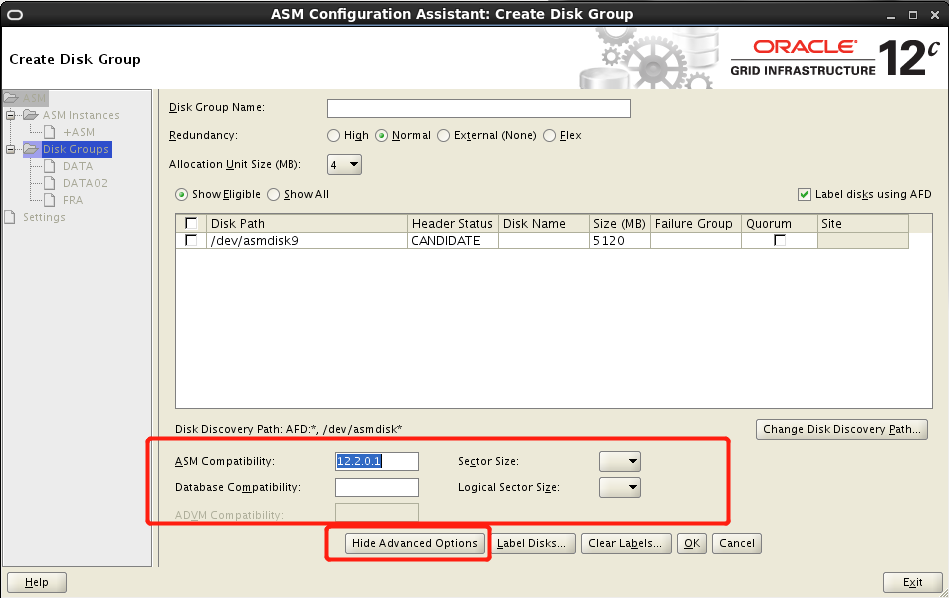


创建磁盘组就点create，进入创建界面：



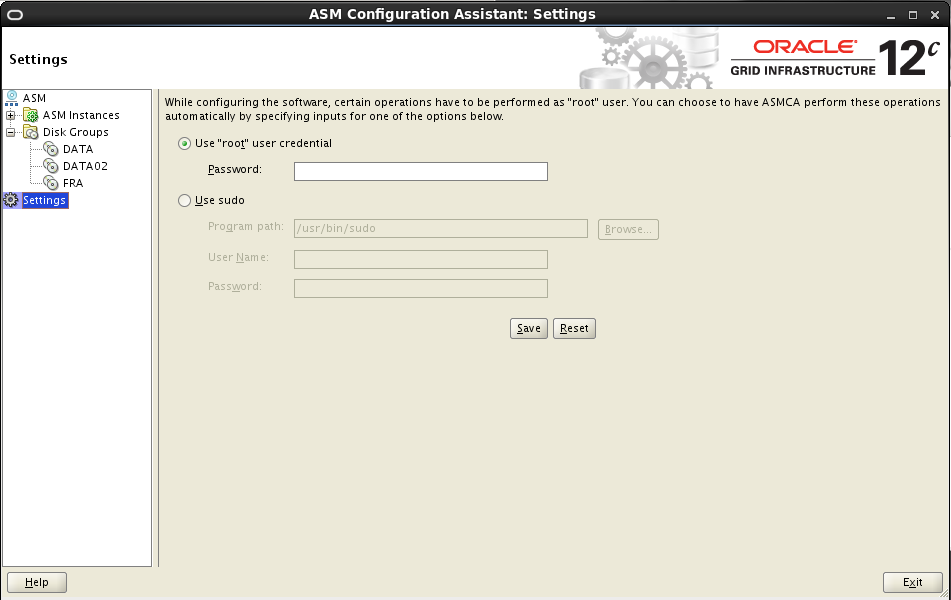
需要给磁盘组取个名字，选择冗余模式，选择AU的大小，选择使用的磁盘，选择是否使用AFD来label磁盘，选择磁盘（当然在选择磁盘之前，应该先确认/修改disk discovery path）。

点Show Advanced Options可以配置兼容性、扇区：

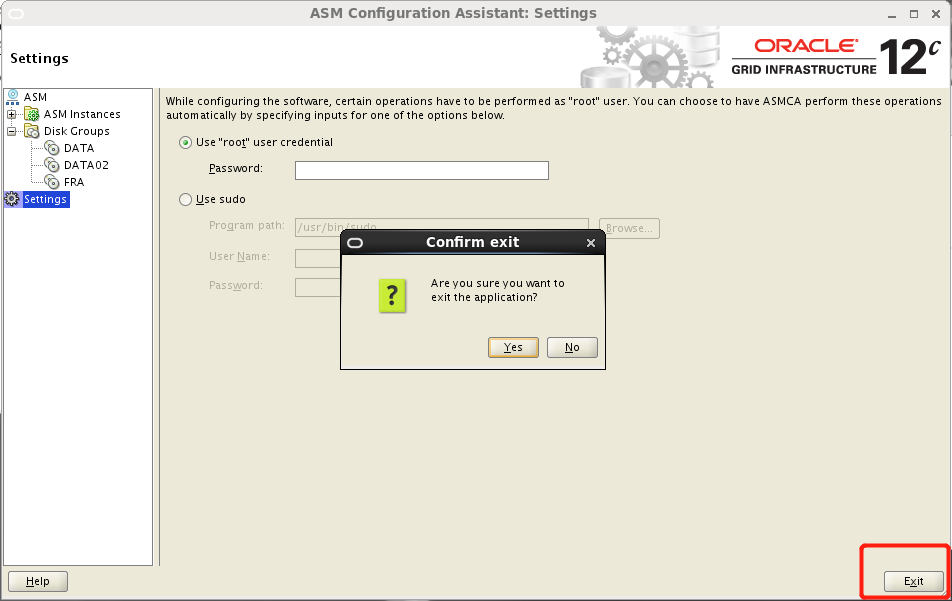


点Cancel可以退出创建界面，如果创建号磁盘组，就点OK来确认创建磁盘组。

如下是setting选项，就是万一某些操作需要root权限，在这里配置好，它就可以自动执行，不用再人为手动输入密码或者root用户执行脚本之类的了：



然后点exit，退出asmca：



# ASMCMD

可以使用在asmcmd的交互窗口中，使用help来查看所有命令：

[grid@yanxin ~]$ asmcmd

ASMCMD> help

asmcmd

Starts asmcmd or executes the command

Synopsis

asmcmd [-V] [--nocp] [-v {errors | warnings | normal | info | debug} ] [--privilege {sysasm | sysdba} ] [-p] [--inst <instance\_name>] [--discover][<command>]

Description

The environment variables ORACLE\_HOME and ORACLE\_SID determine the

instance to which the program connects, and ASMCMD establishes a

bequeath connection to it, in the same manner as a SQLPLUS / AS

SYSASM. The user must be a member of the OSASM group.

If Flex ASM is enabled, the ASMCMD connects to any one of the ASM

instances running in the cluster. The connection to ASM instance does

not depend on the environment variables ORACLE\_HOME and ORACLE\_SID.

The ASMCMD alert log shows to which instance ASMCMD is connected to.

If the user wants to connect to a specific ASM instance, --inst option

should be used to specify the instance name.

Specifying the -V option prints the asmcmd version number and

exits immediately.

Specifying the --nocp option disables connection pooling feature for

ASMCMD.

Specifying the -v option prints extra information that can help

advanced users diagnose problems.

Specify the --privilege option to choose the type of connection. There are

only two possibilities: connecting as SYSASM or as SYSDBA.

The default value if this option is unspecified is SYSASM.

Specifying the -p option allows the current directory to be displayed

in the command prompt, like so:

ASMCMD [+DATA/ORCL/CONTROLFILE] >

Specifying the --discover option uses discovery string obtained from

Listener, does not use ORACLE\_SID.

[command] specifies one of the following commands, along with its

parameters.

Type "help [command]" to get help on a specific ASMCMD command.

commands:

--------

afd\_configure, afd\_deconfigure, afd\_di, afd\_dsget, afd\_dsset

afd\_filter, afd\_label, afd\_lsdsk, afd\_lslbl, afd\_refresh

afd\_scan, afd\_state, afd\_unlabel

md\_backup, md\_restore

amdu\_extract

lsattr, setattr

audcleanaudittrail, audclearproperty, audcleartimestamp

audcreatejob, auddropjob, audloaduniauditfiles, audsetdebug

audsetjobinterval, audsetjobstatus, audsetproperty, audsettimestamp

audsettraillocation, audshowtimestamp, lsaudcleanevents

lsaudcleanupjobs, lsaudconfigparams

cd, cp, du, find, help, ls, lsct, lsdg, lsof, mkalias

mkdir, pwd, rm, rmalias, setsparseparent, showclustermode

showclusterstate, showpatches, showversion

chblk, lsblk, mapau, mapblk, mapextent

chdg, chkdg, dropdg, iostat, lsdsk, lsod, mkdg, mount

offline, online, rebal, remap, stamp, stamplist, umount

chfg, lsfg, mkfg, rmfg

pwcopy, pwcreate, pwdelete, pwget, pwmove, pwset

chqg, lsqg, mkqg, mvfg, rmqg

dsget, dsset, lscc, lsop, mkcc, rmcc, shutdown, spbackup

spcopy, spget, spmove, spset, startup

chtmpl, lstmpl, mktmpl, rmtmpl

chgrp, chmod, chown, groups, grpmod, lsgrp, lspwusr, lsusr

mkgrp, mkusr, orapwusr, passwd, rmgrp, rmusr, rpusr

volcreate, voldelete, voldisable, volenable, volinfo

volresize, volset, volstat

我们列几个常用的：

显式asmcmd的版本：

[grid@yanxin ~]$ asmcmd -V

asmcmd version 12.2.0.1.0

在asmcmd中显示全部路径：

[grid@yanxin ~]$ asmcmd -p

ASMCMD [+] > cd data

ASMCMD [+data] > cd prodcdb

ASMCMD [+data/prodcdb] >

查看当前进行的rebalance的操作，该指令的结果集就是v$asm\_operation：

ASMCMD [+data/prodcdb] > lsop

Group\_Name Pass State Power EST\_WORK EST\_RATE EST\_TIME

DATA02 COMPACT WAIT 0

DATA02 RESYNC WAIT 0

DATA02 REBUILD WAIT 0

DATA02 REBALANCE WAIT 0

SQL> alter diskgroup data02 rebalance;

Diskgroup altered.

ASMCMD [+data/prodcdb] > lsop

Group\_Name Pass State Power EST\_WORK EST\_RATE EST\_TIME

查看各磁盘组的空间使用、状态等：

ASMCMD [+data/prodcdb] > lsdg

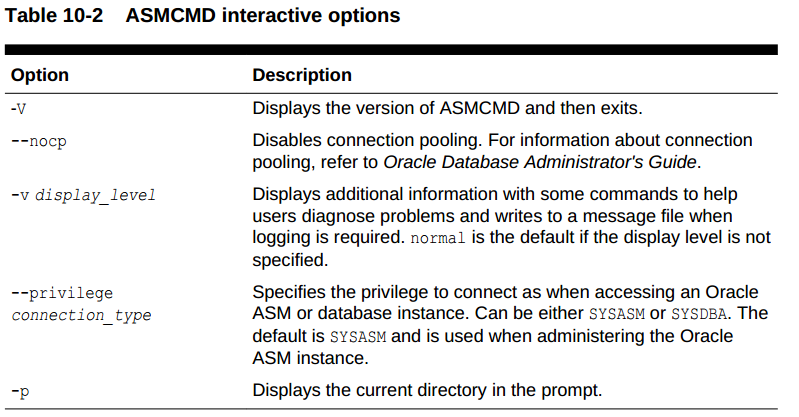
State Type Rebal Sector Logical\_Sector Block AU Total\_MB Free\_MB Req\_mir\_free\_MB Usable\_file\_MB Offline\_disks Voting\_files Name

MOUNTED EXTERN N 512 512 4096 4194304 10240 4408 0 4408 0 N DATA/

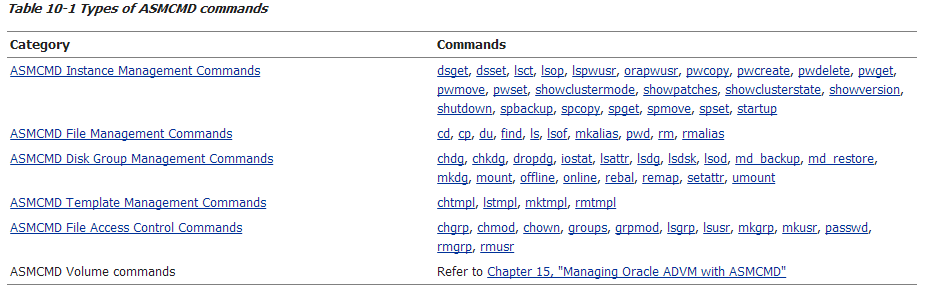
MOUNTED NORMAL N 512 512 4096 4194304 20480 20256 10240 5008 0 N DATA02/

MOUNTED EXTERN N 512 512 4096 4194304 10240 9240 0 9240 0 N FRA/

如下是asmcmd交互选项的总结：



如下是amscmd中指令总结：



# ADVM和ACFS