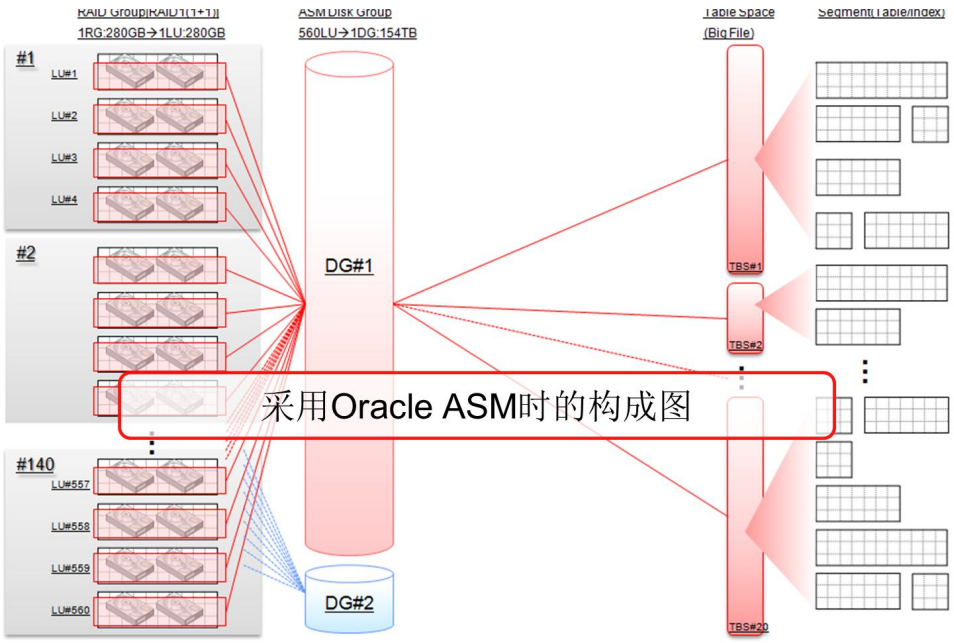
数据库 ASM 管理手册

目录

1. ASM 基础



1.1. ASM 磁盘

Oracle ASM disks are the storage devices that are provisioned to Oracle ASM disk groups. Examples of Oracle ASM disks include:

- A disk or partition from a storage array

  Oracle recommends that you use hardware RAID functionality to create LUNs (Logical Unit Number). Storage hardware RAID 0+1 or RAID5, and other RAID configurations, can be provided to Oracle ASM as Oracle ASM disks.

- An entire disk or the partitions of a disk

  However, the Oracle ASM disk cannot be in a partition that includes the partition table because the partition table would be overwritten.磁盘或分区不能包含磁盘分区表信息。对磁盘来说，只需要 FDISK，不需要 FORMAT。

- Logical volumes

  Logical volume configurations are not recommended by Oracle because they create a duplication of functionality. Oracle also does not recommended using logical volume

managers for mirroring because Oracle ASM provides mirroring. 不建议使用 LVM。LVM 实现的功能与 ASM 有重复。

- Network-attached files (NFS)

An Oracle ASM disk group can be created from NFS files, including Oracle Direct NFS (dNFS), as well as whole disks, partitions, and LUNs.

Direct NFS can be used to store data files, but is not supported for Oracle Clusterware files. To install Oracle Real Application Clusters (Oracle RAC) on Windows using Direct NFS, you must also have access to a shared storage method other than NFS for Oracle Clusterware files.

In a cluster, a disk may be assigned different operating system device names on different nodes, but the disk has the same Oracle ASM disk name on all of the nodes. In a cluster, an Oracle ASM disk must be accessible from all of the instances that share the disk group.

Oracle ASM Dynamic Volume Manager (Oracle ADVM) volumes and Oracle Automatic Storage Management Cluster File System (Oracle ACFS) file systems are currently not supported on disk groups that have been created from NFS or Common Internet File System (CIFS) files.

需要注意，集群中各节点对于的磁盘名可能不一致，但是对应的 ASM 磁盘名称必须一致。这也就是 Linux 需要使用 udev 绑定的原因。

Block or **raw devices** are **not supported** by Oracle Universal Installer (OUI) or Database Configuration Assistant (DBCA). However, manually configured raw or block devices are supported by Oracle, but not recommended.

ASM 磁盘对文件属组的要求为：oracle:dba 或 grid:asmadmin。权限要求为：660。

Linux 下，扫描磁盘大小变化：

echo "- - -" >/sys/class/scsi_host/host{N}/scan

1.1.1. Allocation Units

Every Oracle ASM disk is divided into allocation units (AU). An allocation unit is the fundamental unit of allocation within a disk group. A file extent consists of one or more allocation units. An Oracle ASM file consists of one or more file extents.

When you create a disk group, you can set the Oracle ASM allocation unit size with the AU_SIZE disk group attribute. The values can be 1, 2, 4, 8, 16, 32, or 64 MB, depending on the specific disk group compatibility level. Larger AU sizes typically provide performance advantages for data warehouse applications that use large sequential reads.

AU_SIZE 类似于表空间的 extend，较大的 extend 可以避免频繁扩展带来的开销。 也类似于 LVM 中的 PE Size。11G 支持可变 AU。

10G 调整 AU_SIZE 的隐含参数：

#ASM AU 16MB
_asm_ausize=16777216
#ASM fine grain stripesize 1MB
_asm_stripesize=1048576

1.1.2. Disk Partners

Disk Partnership 是一种基于 2 个磁盘之间的对称关系，存在于 high 或 normal 的 redundancy diskgroup 中。Diskgroup 中的 Disk 与同一个 Diskgroup 内的其他几个 disk 组成结伴关系。ASM 会自动创建和维护这种关系。镜像拷贝数据仅仅在已经与主数据镜像 primary data extent 组成 partners 关系的磁盘上分配。

Disk partnering 用来减少由于同时 2 个磁盘故障导致的数据丢失的概率。原因在于当 ASM 配置中使用了较多的磁盘时(例如上千个)，如果如果数据镜像是随机寻找次级磁盘来存放镜像拷贝，当 2 个磁盘丢失时有较大概率丢失数据。原因是如果采取随机存放镜像数据的话，出现数据的 primary 和镜像数据同时存在于 2 个正好失败的磁盘上的概率是很高的。 如果我们不采取 disk partnering，2 个磁盘失败所造成的数据丢失的概率大大增加。

Disk partnering 策略限制了用来保护某个磁盘数据拷贝的磁盘数目。ASM 为一个磁盘限制了 disk partners 的总数为 8。 这个数目越小，则双磁盘同时失败造成数据丢失概率越小。 但是这个数目越小，也会造成其他不便。所以 ORACLE ASM 研发团队最终选择了 8 这个数字。

ASM 从本 disk 所在 Failure group 之外的 FG 中挑选 partners disk，由于一个 ASM DISK 有多个 partners，所以其多个 partners disk 可能有的在同一个 failure Group 中。

Partners 被尽可能多的选择在不同的 Failure Group 中，这样做的目的也很明确，提高磁盘失败时的容错能力。

1.1.3. ASM metadata

Asm Metadata 是存在于 ASM disk header 用以存放 ASM Diskgroup 控制信息的数据，Metadata 包括了该磁盘组中有哪些磁盘，多少可用的空间，其中存放的 File 的名字，一个文件有哪些 Extent 等等信息。

由于 Asm metadata 就存放在 ASM DISK HEADER，所以 ASM disk group 是自解释的。所有的 metadata 元数据均存放在一个个 metadata block 中(默认 block size 4096)。这些信息包括该 metadata block 的类型以及其逻辑位置。同样有 checksum 信息来确认 block 是否被损坏。所有的 metadata block 均是 4k 大小。实际使用时 ASM 实例会缓存这些 ASm metadata。

1. 通过 kfed 判断磁盘文件头

[grid@rac11g1 ~]$ kfed read /dev/asm-data1 aun=0 blkn=0 text=asm-data1.head

[grid@rac11g1 ~]$ cat asm-data1.head | grep -E 'type|name|size'

```
kfbh.type:                            1 ; 0x002: KFBTYP_DISKHEAD
kfdhdb.dskname:             DATA_DG_0000 ; 0x028: length=12
kfdhdb.grpname:                  DATA_DG ; 0x048: length=7
kfdhdb.fgname:              DATA_DG_0000 ; 0x068: length=12
kfdhdb.capname:                          ; 0x088: length=0
kfdhdb.secsize:                     512 ; 0x0b8: 0x0200
kfdhdb.blksize:                    4096 ; 0x0ba: 0x1000
kfdhdb.ausize:                  1048576 ; 0x0bc: 0x00100000
kfdhdb.dsksize:                    5120 ; 0x0c4: 0x00001400
```

ASM 盘头信息，存放在初始的 4096 字节中。

2. 通过 dd 命令获取磁盘文件头

[grid@oracle11 ~]$ dd if=/dev/asm-data1 bs=1 count=4096 | hexdump -c

```
0000000 001 202 001 001  \0  \0  \0  \0  \0  \0 200 371 276 266 262
0000010  \0  \0  \0  \0  \0  \0  \0  \0  \0  \0  \0  \0  \0  \0  \0
0000020   O   R   C   L   D   I   S   K  \0  \0  \0  \0  \0  \0  \0
```

```
0000030  \0  \0  \0  \0  \0  \0  \0  \0  \0  \0  \0  \0  \0  \0  \0  \0
0000040  \0  \0      \v  \0  \0 001 003   D   A   T   A   D   G   _   0
0000050   0   0   0  \0  \0  \0  \0  \0  \0  \0  \0  \0  \0  \0  \0  \0
```

ORACLE 内部的判断逻辑：

```
if ( buf[32] == 'O' && buf[33] == 'R' && buf[34] == 'C' && buf[35] == 'L'
   && buf[36] == 'D' && buf[37] == 'I' && buf[38] == 'S' && buf[39] == 'K')
{
    if (buf[71] != 4)
    {
        printf("This disk %s still used by ASM\n",argv[1]);
    }
    else
    {
        printf("This disk %s has been used by ASM\n",argv[1]);
    }
}
```

1.1.4. ASM 磁盘配置

- Linux

建议使用磁盘分区进行配置(FDISK 对磁盘进行分区，只分一个区，并且不进行格式化)。使用 UDEV 进行绑定。

- AIX

AIX 会为加入卷组的磁盘分配物理卷标识(PVID)。该标识会存放在物理磁盘和 ODM 中，PVID 信息存放在磁盘的前 4K 中，通过 lspv 命令可以查看到。

```
# /usr/sbin/chdev -l hdisk1 -a pv=clear
```

针对 RAC 环境的磁盘属性设置：

- EMC 存储

```
# /usr/sbin/chdev -l hdiskn -a reserve_lock=no
```

- IBM 存储

```
# /usr/sbin/chdev -l hdiskn -a reserve_policy=no_reserve
```

- HDS 存储

/usr/sbin/chdev -l hdiskn -a dlmrsvlevel=no_reserve

- HP

  没有特殊要求，使用 /dev/rdisk/disk* 设备即可。

- Solaris

  Note that slices 0 and 2 cannot be used as ASM disks because these slices include the Solaris VTOC. 使用 format 命令进行格式化。

  Notice that slice 4 is created and that it skips four cylinders, offsetting past the VTOC.

[root@racnode1]# ls -l /dev/rdsk/c0t2d0s4

```
lrwxrwxrwx 1 root root 45 Feb 24 07:14 c0t2d0s4 -> ../../devices/pci@1f,4000/scsi@3/sd@2,0:e,raw
```

[root@racnode1]# chown oracle:dba ../../devices/pci@1f,4000/scsi@3/sd@2,0:e,raw

- Windows

C:\windows> diskpart

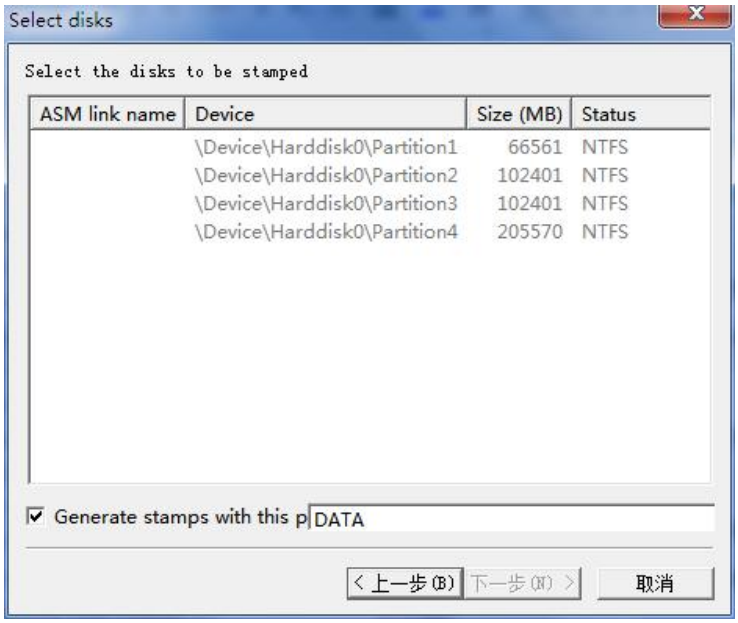
DISKPART> LIST DISK

```
  磁盘 ###  状态           大小     可用     Dyn  Gpt
  --------  -------------  -------  -------  ---  ---
  磁盘 0    联机              465 GB  4096 KB
```

C:\windows> asmtool -list

```
NTFS                            \Device\Harddisk0\Partition1            66561M
NTFS                            \Device\Harddisk0\Partition2           102401M
NTFS                            \Device\Harddisk0\Partition3           102401M
NTFS                            \Device\Harddisk0\Partition4           205570M
```

C:\windows> asmtoolg



1.2. ASM 磁盘组

When you create a disk group, you specify an Oracle ASM disk group type based on one of the following three redundancy levels:

- Normal for 2-way mirroring
- High for 3-way mirroring
- External to not use Oracle ASM mirroring, such as when you configure hardware RAID for redundancy

The redundancy level controls how many disk failures are tolerated without dismounting the disk group or losing data.

1.2.1. Failure Group

ASM 提供冗余，failure group 用来保证单点错误不会造成同一数据的多份拷贝同时不可用。 如果 ASM 使用的多个 ASM DISK LUN 属于同一硬件 例如同一磁盘阵列，该硬件故障会导致这多个盘均不可用，则该硬件的失败应当被容错， 在 ASM 中一般将这些盘规划到同一个 failure group 中。多份冗余拷贝不会存放在同一个 failure group 的磁盘中，换句话说一个 failure group 中只有一份数据的拷贝，不会有第二份。

由于 Failure Group 的配置很大程度上与用户的本地规划有关，所以 ASM 允许用户自己指定 Failure group 的规划、组成。 但是如果用户自己没有指定 Failure Group 的规划，那么 ASM 会自动分配磁盘到必要的 Failure Group。

使用 External Redundancy 的 Diskgroup 没有 Failure Group。Normal redundancy Disk Groups 要求至少 2 个 Failure Group，High Redundancy Disk Groups 要求 3 个 Failure Group。

如果 Normal redundancy Disk Groups 中有多于 2 个的 Failure Group，例如 Failure Group A、B、C，则一个 Virtual Extent 会自动在 A、B、C 之间找 2 个 Failure Group 存放 2 个 mirror extent，不会存放 3 份拷贝。

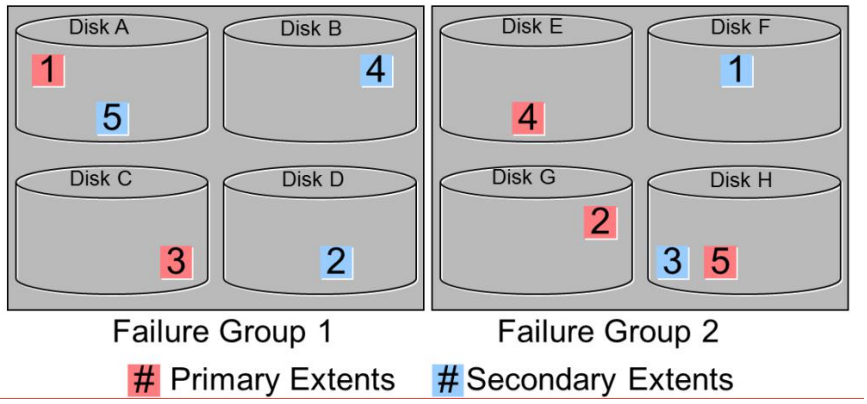
实际应用中 Normal/High 一般会和多个存储控制器 Controller 结合来分配 failure group，或者存在多路存储可用。

1.2.2. ASM mirror 保护

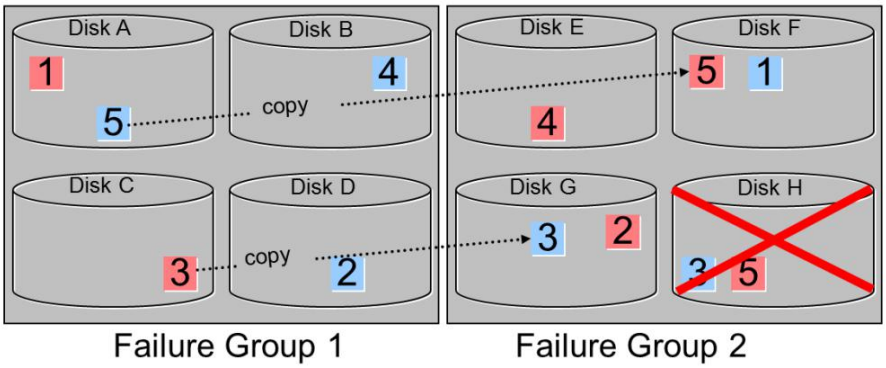
ASM mirror 镜像保护可避免因丢失个别磁盘而丢失数据。每一个文件均有自己的 ASM 镜像策略属性， 对于该文件所辖的所有 virtual extent 来说同样基于该镜像策略。文件创建时会设置这个镜像策略属性，今后都无法修改。 ASM 镜像要比操作系统镜像磁盘要来的灵活一些，至少它可以在文件级别指定需要的冗余度。

ASM mirror 区分镜像 extent 的 primary 和 secondary 拷贝，但在更新 extent 时会同时写所有的拷贝镜像。 ASM 总是先尝试读取 primary 拷贝，仅当 primary 拷贝不可用时去读取 secondary 拷贝。

以下为一个示例， 一个 normal redundancy 的 Diskgroup 中存在 8 个 Disk，并使用 2 个 Failure Group：



当磁盘 Disk H 失败，这个失败要求在失败磁盘上所有的 Extent 均被修复， Extent 3 和 5 会从现存的拷贝中复制到 Failgroup 2 中可用的区域。在此例子中，Extent 5 被从 Disk A 拷贝到 Disk F，extent 3 从 Disk C 拷贝到 Disk G，最后还会将失败的磁盘从 Diskgroup 中 drop 出去。



1.2.3. Disk Group Mount

在数据库实例可以用 Diskgroup 上的文件之前，需要 ASM 实例去 mount 这个本地 diskgroup。 Mount Diskgroup 牵扯到发现所有的磁盘并找到上面已经有 metadata 数据的 disk，并尝试将对应到一个 diskgroup 的 DISK mount 起来。 能 mount 起来的前提还需要验证 metadata 来确保现在已经有足够数量的磁盘在哪里，例如使用 3 个 DISK 创建的 external diskgroup ，当前 OS 下面只挂了 2 个 Disk，则显然不能 mount 这个 diskgroup。之后还需要初始化 SGA 以便之后更新和管理这些 metadata。

可以显示地去 dismount 一个 diskgroup，但是如果 diskgroup 上的文件正在被 client (例如 DB)使用则 dismount 会报错。如果在 ASM 冗余算法容错能力内丢失磁盘，则不会导致 diskgroup 被强制 dismount。但是如果超出了容错能力则会被强制 dismount。 这种强制 dismount 会导致使用其上文件的 DB instance 被 kill。

1.3. ASM 文件

ORACLE RDBMS Kernel 内核与 ASM 在高层交互是基于 ASM 中存放的文件即 ASM FILE。 这和 ORACLE RDBMS 去使用文件系统或其他逻辑卷的方式没有什么区别。 ASM 中可以存放数据文件，日志文件，控制文件，归档日志等等，对于数据库文件的存放基本和文件系统没什么区别。

一个 ASM FILE 的名字一般以一个”+”和 DiskGroup 名字开头。 当 ORACLE RDBMS KERNEL 内核的文件 I/O 层碰到一个以”+”开头的文件时，就会走到相关 ASM 的代码层中 而不是调用依赖于操作系统的文件系统 I/O。 仅仅在 File I/O 层面才会认识到这是一个 ASM 中的文件，而其上层的内核代码看来 ASM FILE 和 OS FILE 都是一样的。

ASM 对 ROWID 和 SEGMENT 等 RDBMS 元素没有影响，不过是数据文件存放在 ASM 中，ASM 并不会打破 ORACLE 数据库中的这些经典元素。

在一个 ASM Diskgroup 中仅仅允许存放已知的 ORACLE 文件类型。假设一个文件通过 FTP 拷贝到 ASM Diskgroup 中，则该文件的第一个块将被检验以便确认其类型，以及收集其他信息来构建这个文件的完整 ASM 文件名。 如果其文件头无法被识别，则该文件在 DiskGroup 中的创建将会报错。

每一个文件，在 ASM 中都有一个专门的索引号，也就是编号，ASM 文件索引号从 1 开始。其中，前 255 个，也就是 1 至 255 号文件，都是元文件。256 之后的是其他各种文件。 元文件中包含了各种 ASM 的配置、各类数据文件信息还有目录、别名等等信息，都是在元文件中的。所有 V$ASM_开头视图的信息，都来自元文件中。

```
SQL> SELECT FILE_NUMBER, BYTES, BLOCK_SIZE, TYPE FROM V$ASM_FILE
ORDER BY 1;
```

FILE_NUMBER	BYTES	BLOCK_SIZE	TYPE
253	1536	512	ASMPARAMETERFILE
256	786440192	8192	DATAFILE
257	608182272	8192	DATAFILE
258	94380032	8192	DATAFILE
259	5251072	8192	DATAFILE
260	9748480	16384	CONTROLFILE
261	52429312	512	ONLINELOG
262	52429312	512	ONLINELOG
263	52429312	512	ONLINELOG
264	30416896	8192	TEMPFILE
265	2560	512	PARAMETERFILE

其中，1 号文件包含所有文件的磁盘占用信息，包括元文件、甚至 1 号文件自身的空间分布信息，也都是在 1 号文件内部。每个文件在它里面占用一个块（4096 字节，元数据块大小为 4K）的空间。

从 256 号文件开始，是数据库的各类文件。假设你放在 ASM 上的第一个文件是一个控制文件 A，第二个文件是一个数据文件 B。哪么控制文件 A 在 ASM 中的索引号是 256，数据文件 B 的索引号是 257。

1 号文件总是开始在 0 号磁盘 2 号 AU，记住这个位置：0 号盘 2 号 AU。这是 ASM 中定位文件的起点。它的作用，有点相当于磁盘上的引导区，在电脑开机后负责将 OS 启动起来。

1 号文件在最少情况下，至少有两个 AU。上面我们提到过了，在 1 号文件中，每个文件占用一个元数据块，存放自身的空间分布信息。每个元数据块大小是 4K，一个 AU 是 1M，哪么，每个 AU 中，可以存储 256 个文件的空间分布信息。这其中，0 号盘 2 号 AU 中，全是元文件的信息。再具体一点，0 号盘 2 号 AU，第一个元数据块被系统占用，从第二个块开始，到 255 为止，共 255 个元数据块，对应索引号 1 至 255 的文件。其实，也就是全部的元文件了。也就是说 0 号盘 2 号 AU，保存了全部元文件的空间分布信息。

SQL> **SELECT GROUP_NUMBER,NAME FROM V$ASM_DISKGROUP;**

```
GROUP_NUMBER NAME
------------ ------------------------------------------------------------

           1 DATADG

           2 GRIDDG
```

SQL> **SELECT GROUP_NUMBER,DISK_NUMBER,PATH FROM V$ASM_DISK ORDER BY 1, 2;**

```
GROUP_NUMBER DISK_NUMBER PATH
------------ ----------- ------------------------------------------------

           0           0 /dev/asm-data3

           0           6 /oracle/asm/fakeasm1

           1           0 /dev/asm-data1 --> 磁盘组 0 号磁盘

           1           1 /dev/asm-data2

           2           0 /dev/asm-grid1

           2           1 /dev/asm-grid2

           2           2 /dev/asm-grid3
```

使用 kfed 直接读取 0 号磁盘，2 号 AU，1 号元数据块。0 号元数据块是 1 号文件自身留作文件头的。1 号元数据块呢，是 1 号文件的 AU 分布，2 号元数据块，是 2 号文件的 AU 分布。等等。

[root@oracle11 ~]# **kfed read /dev/asm-data1 aun=2 blkn=1 | more**

```
kfbh.endian:                          1 ; 0x000: 0x01
```

kfbh.endian 的十进值为 1，0x000 是指它开始自第 0 个字节处，最后的 0x01 是十六进制值形式。此域的意义是主机的大小端。0 是大端，1 是小端。此处值为 1，说明主机是小端。

```
kfbh.hard:                          130 ; 0x001: 0x82
kfbh.type:                            4 ; 0x002: KFBTYP_FILEDIR
kfbh.datfmt:                          1 ; 0x003: 0x01
```

```
kfbh.block.blk:                       1 ; 0x004: blk=1
--kfbh.block.obj，它代表此数据块属于哪个文件
kfbh.block.obj:                       1 ; 0x008: file=1
.................
kfffde[0].xptr.au:                    2 ; 0x4a0: 0x00000002          --> 2 号 AU
kfffde[0].xptr.disk:                  0 ; 0x4a4: 0x0000              --> 0 号磁盘
kfffde[0].xptr.flags:                 0 ; 0x4a6: L=0 E=0 D=0 S=0     --> 标志位
kfffde[0].xptr.chk:                  40 ; 0x4a7: 0x28                --> 校验码
kfffde[1].xptr.au:                   58 ; 0x4a8: 0x0000003a --> 58 号 AU (58 对应 16 进制 3a)
kfffde[1].xptr.disk:                  0 ; 0x4ac: 0x0000
kfffde[1].xptr.flags:                 0 ; 0x4ae: L=0 E=0 D=0 S=0
kfffde[1].xptr.chk:                  16 ; 0x4af: 0x10
kfffde[2].xptr.au:           4294967295 ; 0x4b0: 0xffffffff --> 该标号标识没有这个 AU
kfffde[2].xptr.disk:               65535 ; 0x4b4: 0xffff
kfffde[2].xptr.flags:                 0 ; 0x4b6: L=0 E=0 D=0 S=0
kfffde[2].xptr.chk:                  42 ; 0x4b7: 0x2a
kfffde[3].xptr.au:           4294967295 ; 0x4b8: 0xffffffff
kfffde[3].xptr.disk:               65535 ; 0x4bc: 0xffff
```

kfffde，是结构数组。kfffde[0]的数据元素，存放了 1 号文件第一个 AU 的位置。kfffde[1]存放了 1 号文件第二个 AU 位置，等等，依次类推。

[root@oracle11 ~]# **dd if=/dev/asm-data1 bs=1 count=48 | hexdump -c**

```
0000000 001 202 001 001  \0  \0  \0  \0  \0  \0  \0 200 344 323 271 223
0000010 316  \r  \0  \0  \0  \0  \0  \0  \0  \0  \0  \0  \0  \0  \0  \0
0000020   O   R   C   L   D   I   S   K  \0  \0  \0  \0  \0  \0  \0  \0
48+0 records in
48+0 records out
0000030
48 bytes (48 B) copied, 0.0075762 s, 6.3 kB/s
```

1 号文件的第一个 AU（0 号盘 2 号 AU）中，只能保存 1 至 255 号文件的。从 256 号文件开始，AU 的分布信息保存在 1 号文件第二个 AU 中，也就是（0 号盘，58 号 AU）。

```
kfffde[1].xptr.au:                    58 ; 0x4a8: 0x0000003a --> 58 号 AU (58 对应 16 进制 3a)
```

其中第一个块（0 号块），对应 256 号文件。1 号块对应 257 号文件，等等，依此类推。

[grid@oracle11 ~]$ **kfed read /dev/asm-data1 aun=58 blkn=0 | more**

```
kfbh.endian:                          1 ; 0x000: 0x01
kfbh.hard:                          130 ; 0x001: 0x82
kfbh.type:                            4 ; 0x002: KFBTYP_FILEDIR
kfbh.datfmt:                          1 ; 0x003: 0x01
kfbh.block.blk:                     256 ; 0x004: blk=256
kfbh.block.obj:                       1 ; 0x008: file=1
```

- on each disk, AU=0: disk header (disk name, etc), first stride of the Allocation Table (AT) and Free Space Table (FST)

[grid@oracle11 ~]$ **kfed read /dev/asm-data1 aun=0 | more**

- on each disk, AU=1: space allocate for the Partner Status Table (PST) (not all disks have PST data)

[grid@oracle11 ~]$ **kfed read /dev/asm-data1 aun=1 | more**

- on each disk, AU=11 block 1: 12c additional copy of the disk header

You can store the various file types in Oracle ASM disk groups, including:

- Control files
- Data files, temporary data files, and data file copies
- SPFILEs
- Online redo logs, archive logs, and Flashback logs
- RMAN backups
- Disaster recovery configurations
- Change tracking bitmaps
- Data Pump dumpsets

Oracle ASM automatically generates Oracle ASM file names as part of file creation and tablespace creation. Oracle ASM file names begin with a plus sign (+) followed by a disk group name.

命名以+号开头，后面跟随磁盘组名称。

```
DB_Name   Instance_Name   Path
racdb      racdb1         +systemdg/racdb/controlfile/current.256.831554215
racdb      racdb1         +systemdg/racdb/datafile/sysaux.260.831554225
```

ASM 文件命令规则：

```
+diskgroup_name/database_name/database file type/tag_name.file_number.incarnation
```

1.3.1. File Blocks

所有被 ASM 所支持的文件类型仍以其 file block 作为读和写的基本单位。在 ASM 中的文件仍保持其原有的 Block Size 例如 Datafile 仍是 2k~32k(默认 8k)，ASM 并不能影响这些东西。

值得一提的是在 ASM FILE NUMBER 1 的 FILEDIR 中记录了每一种 FILE TYPE 对应的 BLOCK SIZE，例如：

这里的 kfffdb.blkSize 即是一个数据文件的 Block Size。

由于这个 blocksize 总是 2 的次方,所以一个 block 总是在 一个 AU allocation Unit 中，而不会跨 2 个 AU。

[grid@oracle11 ~]$ **kfed read /dev/asm-data1 aunum=0 blkn=0 | grep -i blk**

```
kfbh.block.blk:                       0 ; 0x004: blk=0
kfdhdb.blksize:                    4096 ; 0x0ba: 0x1000
kfdhdb.acdb.aba.blk:                  0 ; 0x1d8: 0x00000000
```

每种文件对应的块大小都有不同。

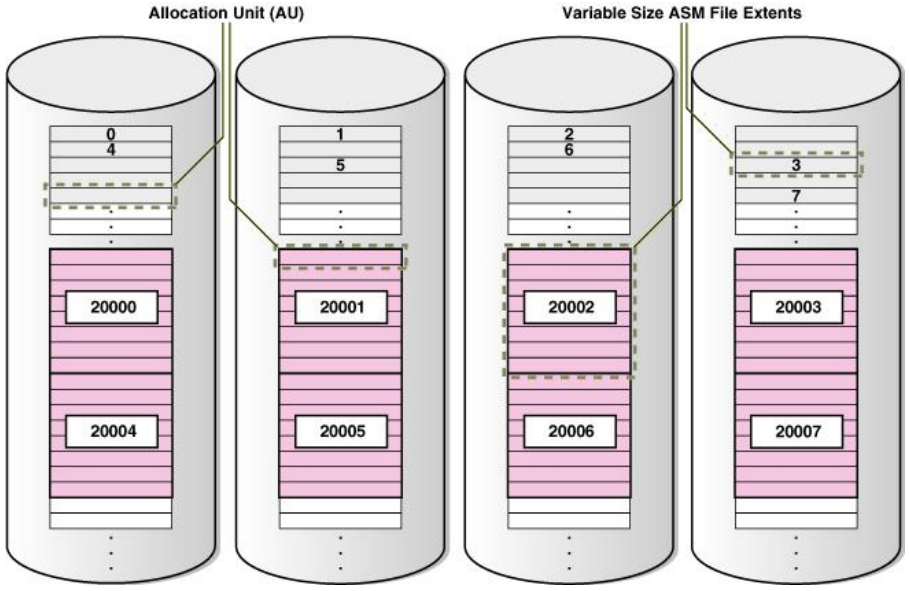
```
SQL> SELECT FILE_NUMBER,
       (SELECT A.NAME
          FROM V$ASM_ALIAS A
         WHERE A.ALIAS_DIRECTORY = 'N'
           AND A.SYSTEM_CREATED = 'Y'
           AND A.FILE_NUMBER = F.FILE_NUMBER) NAME,
       BLOCK_SIZE
  FROM V$ASM_FILE F;

FILE_NUMBER NAME                                 BLOCK_SIZE
```

```
---------- ------------------------------------ ----------

       256 SYSTEM.256.880415831                       8192
       257 SYSAUX.257.880415833                       8192
       258 UNDOTBS1.258.880415833                     8192
       259 USERS.259.880415833                        8192
       260 Current.260.880415929                     16384
       261 group_1.261.880415931                       512
       262 group_2.262.880415931                       512
       263 group_3.263.880415933                       512
       264 TEMP.264.880415939                         8192
       265 spfile.265.880416171                        512
       253 REGISTRY.253.880414225                      512
```

1.3.2. Data Extents



数据盘区 Data Extents 是裸的存储，用以存放文件内容。每一个 Data Extent 在 11g 之前对应某一个 ASM disk 上的一个 Allocation Unit，在 11g 之后 一个 Extent 可以对应多个 AU。

The contents of Oracle ASM files are stored in a disk group as a set, or collection, of extents that are stored on individual disks within disk groups. Extents consist of one or more allocation units (AU).

Variable size extents enable support for larger Oracle ASM data files, reduce SGA memory requirements for very large databases, and improve performance for file create and open operations.

The extent size of a file varies as follows:

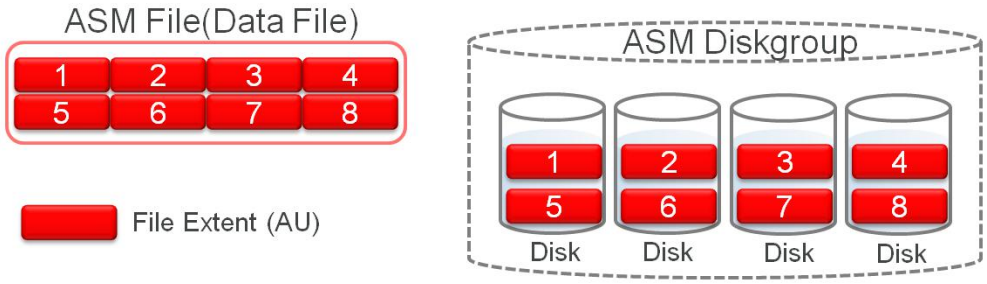
- Extent size always equals the disk group AU size for the first 20000 extent sets (0 - 19999).
- Extent size equals 4*AU size for the next 20000 extent sets (20000 - 39999).
- Extent size equals 16*AU size for the next 20000 and higher extent sets (40000+).

select * from x$kffxp where size_kffxp!=1 and rownum<3;

kfed read /dev/asm-disk9 aun=22 blkn=201|less

http://www.askmaclean.com/archives/%E3%80%90oracle-asm%E3%80%91variable-extent-size-%E5%8E%9F%E7%90%86.html

1.3.3. Virtual Data Extents

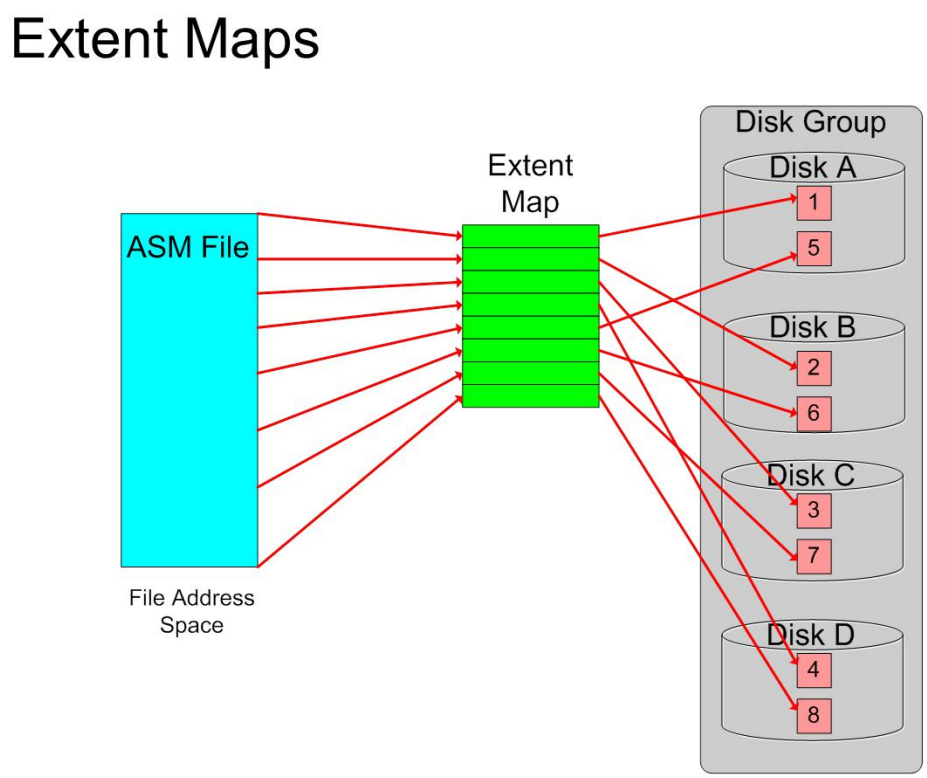


虚拟数据盘区 Virtual Data Extent 是几个 data extent 的集合，这些 data Extent 中包含了相同的数据。 镜像 Mirror 是在虚拟 Extent 级别实现的。每一个虚拟 extent 为文件块提供一个盘区地址空间。每一个写入到文件块均是写入到一个虚拟 extent 中每一个 Online 在线的 data extent 中。每一个对文件块的读取也是被重定位到一个虚拟 extent 中的主镜像 extent (primary Extent)，除非 primary extent 所在 Disk 被 OFFLINE 了。对于没有冗余度(即 external redundancy disk group 上的 FILE)的文件而言，一个虚拟 Extent 实际就是一个 data Extent。

对于 Normal redundancy+普通数据库文件而言， 一个虚拟 Extent 实际就是 2 个 Data Extent。

对于 High redundancy+普通数据库文件而言， 一个虚拟 Extent 实际就是 3 个 Data Extent。

1.3.4. Extent Map



Extent Map 盘区图是盘区指针的列表，这些指针将支出所有属于一个文件的数据盘区。这些盘区是真正存放数据的裸存储空间。每一个盘区指针给出其所在的磁盘号和 AU 信息。为了保证可信，每一个盘区指针也会包含一个 checksum byte 来确认本指针未损坏。当这个 checksum 值和实际存放的 Extent 信息不匹配时可能出现 ORA-600 错误，例如 ORA-00600: internal error code, arguments: [kffFdLoadXmap_86], [256], [0], [1], [68], [54], [29], [], [], [], [], []。

1.3.5. ASM Striping

Oracle ASM striping has two primary purposes:

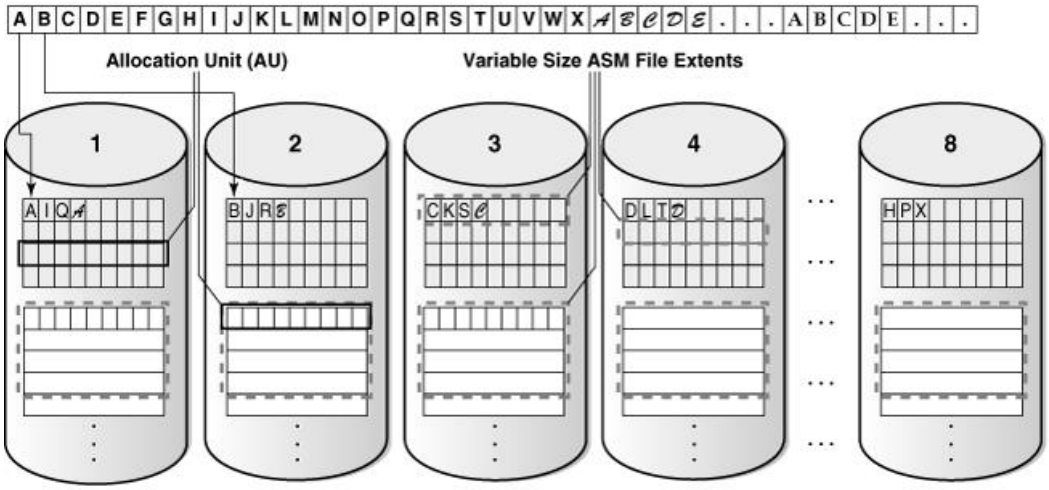
- To balance loads across all of the disks in a disk group
- To reduce I/O latency

To stripe data, Oracle ASM separates files into stripes and spreads data evenly across all of the disks in a disk group. The fine-grained stripe size always equals 128 KB

in any configuration; this provides lower I/O latency for small I/O operations. The coarse-grained stripe size is always equal to the AU size (not the data extent size).

- Oracle ASM Fine-Grained Striping (细粒度条带化)

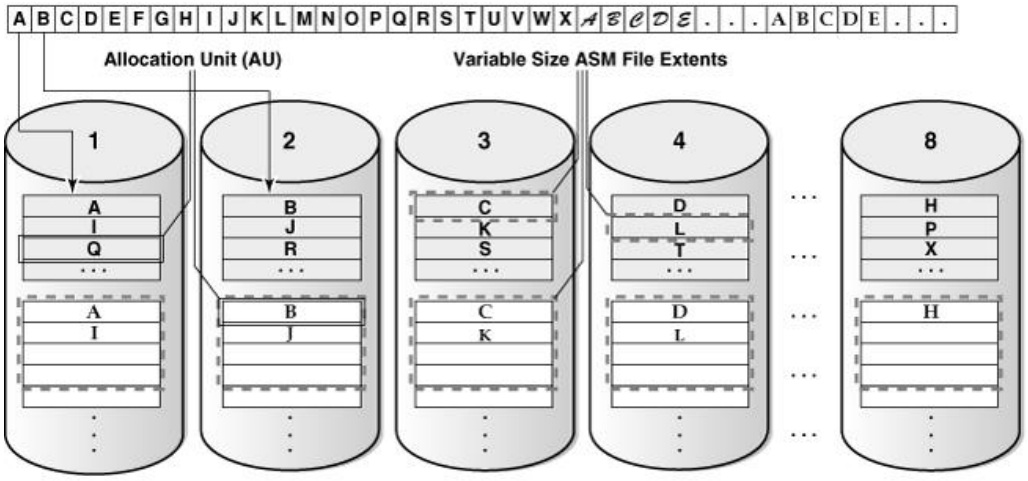
细粒度与粗粒度的区别在于，文件块不是线性地布局在每一个虚拟 Extent 上，而是文件将以 1/8 个虚拟 Extent 成长，由此文件块被在 diskgroup 内以 1/8 的条带化深度分布。 由此当该文件的 block size 为 8k，则 block 0~15 在虚拟 Virtual Extent 0 上面，而 block 16-31 在 Vritual Extent 1 上面，blocks 112-127 在 vritual extent 7， block 128-143 在 block 0-15 之后仍在 virtual extent 0 上。



The file is striped in 128 K chunks (labeled A..X) with each 128 K chunk stored in an extent, starting at the first extent in disk 1, then the first extent in disk 2, and then continuing in a round-robin pattern through all the disks until the entire file has been striped.

- Oracle ASM Coarse-Grained Striping (粗粒度条带化)

粗粒度条带化就是对虚拟 data Extent 的简单联接。类似于传统卷管理器使用 1MB 作为 stripe size。



The file is striped in 1 M chunks (labeled A..X) with each 1 M chunk stored uniquely in an extent, starting at the first extent in disk 1, then the first extent in disk 2, and then continuing in a round-robin pattern through all the disks until the entire file has been striped.

1.3.6. File Templates

System Template	External	Normal	High	Striped
CONTROLFILE	unprotected	2-way mirror	3-way mirror	fine
DATAFILE	unprotected	2-way mirror	3-way mirror	coarse
ONLINELOG	unprotected	2-way mirror	3-way mirror	fine
ARCHIVELOG	unprotected	2-way mirror	3-way mirror	coarse
TEMPFILE	unprotected	2-way mirror	3-way mirror	coarse
BACKUPSET	unprotected	2-way mirror	3-way mirror	coarse
XTRANSPORT	unprotected	2-way mirror	3-way mirror	coarse
PARAMETERFILE	unprotected	2-way mirror	3-way mirror	coarse
DATAGUARDCONFIG	unprotected	2-way mirror	3-way mirror	coarse
FLASHBACK	unprotected	2-way mirror	3-way mirror	fine
CHANGETRACKING	unprotected	2-way mirror	3-way mirror	coarse
AUTOBACKUP	unprotected	2-way mirror	3-way mirror	coarse
DUMPSET	unprotected	2-way mirror	3-way mirror	coarse

S

File Template 文件模板当文件被创建时用以指定 条带化 (coarse 或 FINE) 和冗余度(external, normal, high)。 ORACLE 默认已经为每一个 ORACLE 数据库文件提供了默认模板。可以修改默认模板 也可以客制化模板。修改模板只影响新创建的文件，而不是现有文件。 创建文件时可以指定使用某一个模板。

ASM_SQL> ALTER DISKGROUP DATA ADD TEMPLATE NONCRITCAL_FILES ATTRIBUTES (UNPROTECTED);

ASM_SQL> ALTER DISKGROUP DATA ALTER TEMPLATE NONCRITICAL_FILES ATTRIBUTES (COARSE);

ASM_SQL> ALTER DISKGROUP DATA DROP TEMPLATE NONCRITICAL_FILES;

1.3.7. ASM 文件名称

```
SQL>
SELECT LEVEL, CONCAT('+' || GNAME, SYS_CONNECT_BY_PATH(ANAME, '/')) FULL_ALIAS_PATH
  FROM (SELECT G.NAME GNAME, A.PARENT_INDEX PINDEX, A.NAME ANAME, A.REFERENCE_INDEX RINDEX
          FROM V$ASM_ALIAS A, V$ASM_DISKGROUP G
         WHERE A.GROUP_NUMBER = G.GROUP_NUMBER)
 START WITH (MOD(PINDEX, POWER(2, 24))) = 0
CONNECT BY PRIOR RINDEX = PINDEX;
```

```
     LEVEL FULL_ALIAS_PATH
---------- --------------------------------------------------------------------------------
         1 +GRIDDG/ASM
         2 +GRIDDG/ASM/ASMPARAMETERFILE
         3 +GRIDDG/ASM/ASMPARAMETERFILE/REGISTRY.253.880414225
```

1.4. ASM 目录

SQL> ALTER DISKGROUP DATA ADD DIRECTORY '+DATA/yoda/oradata';

ASMCMD> mkdir +DATA/yoda/oradata

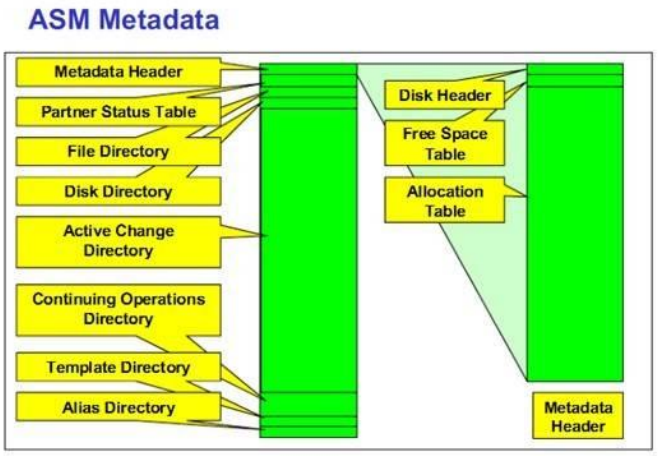
1.5. ASM 与多路径

Multipathing solutions provide failover by using redundant physical path components. These components include adapters, cables, and switches that reside between the server and the storage subsystem.

If Oracle ASM discovers multiple paths to the same disk device, Oracle ASM then raises an error. Because a single disk can appear multiple times in a multipath configuration, you must configure Oracle ASM to discover only the multipath disk.

2. ASM 元数据

对于 asm，可以理解为跟 database 一样，是一个单独的数据库，而 asm 元数据，则可以理解为 database 中的数据字典信息。



SQL>

```
SELECT NUMBER_KFFXP "ASM file number",
      DECODE(NUMBER_KFFXP,
            1,
            'File directory',
            2,
            'Disk directory',
            3,
            'Active change directory',
            4,
            'Continuing operations directory',
            5,
            'Template directory',
            6,
            'Alias directory',
            7,
            'AVD volume file directory',
            8,
            'Disk free space directory',
            9,
            'Attributes directory',
            10,
            'ASM user directory',
            11,
```

```
            'ASM user group directory',
            12,
            'Staleness directory') "ASM metadata file name",
      COUNT(AU_KFFXP) "Allocation units"
  FROM X$KFFXP
 WHERE GROUP_KFFXP = 1 -- disk group 1
   AND NUMBER_KFFXP < 17 -- ASM metadata files
   AND DISK_KFFXP <> 65534 -- ignore disk number 65534
 GROUP BY NUMBER_KFFXP;
```

```
ASM file number ASM metadata file name             Allocation units
--------------- ---------------------------------- ----------------
              1 File directory                                    2
              2 Disk directory                                    1
              3 Active change directory                          42
              4 Continuing operations directory                   8
              5 Template directory                                1
              6 Alias directory                                   1
              8 Disk free space directory                         1
              9 Attributes directory                              1
```

上述查询的 file 信息，其实就的 asm 所涉及的元数据信息，对应关系如下：

file# 1 ---file directory

file# 2 ---disk directory

file# 3 ---active Change Directory(ACD)

file# 4 ---continuing Operations Directory (COD)

file# 5 ---template directory

file# 6 ---alias directory

file# 7 ---volume directory

file# 8 ---disk Used Space Directory (USD)

file# 9 ---attributes directory

2.1. 物理元数据

Physical metadata are located at fixed locations on disk. This fixed location is necessary for ASM bootstrapping. As such, physical metadata cannot be stored in ASM files. The following structures comprise the physical metadata:

- Disk Header
- Allocation Table (AT)

- Free Space Table (FST)
- Partnership Status Table (PST)

2.1.1. Disk Header (磁盘头)

Each ASM disk has a disk header. The disk header, which identifies the disk to ASM, occupies the first block (block 0) of the first allocation unit (AU 0) of each disk.

ASM 磁盘头记录了如下信息：

- disk name
- disk number
- diskgroup name
- failure group name
- disk size
- allocation unit (AU) size
- creation time
- mount time
- ASM compatibility
- RDBMS compatibility
- file directory pointer
- ASMLIB reserved block

[grid@oracle11 ~]$ **kfed read /dev/asm-data1 aun=0 blkn=0 | more**

```
kfbh.endian:                          1 ; 0x000: 0x01
kfbh.hard:                          130 ; 0x001: 0x82
kfbh.type:                            1 ; 0x002: KFBTYP_DISKHEAD
kfbh.datfmt:                          1 ; 0x003: 0x01
kfbh.block.blk:                       0 ; 0x004: blk=0
kfbh.block.obj:              2147483648 ; 0x008: disk=0
kfbh.check:                   398300963 ; 0x00c: 0x17bd9723
kfbh.fcn.base:                     5750 ; 0x010: 0x00001676
kfbh.fcn.wrap:                        0 ; 0x014: 0x00000000
kfbh.spare1:                          0 ; 0x018: 0x00000000
kfbh.spare2:                          0 ; 0x01c: 0x00000000
kfdhdb.driver.provstr:          ORCLDISK ; 0x000: length=8
kfdhdb.driver.reserved[0]:            0 ; 0x008: 0x00000000
kfdhdb.driver.reserved[1]:            0 ; 0x00c: 0x00000000
```

```
kfdhdb.driver.reserved[2]:            0 ; 0x010: 0x00000000
kfdhdb.driver.reserved[3]:            0 ; 0x014: 0x00000000
kfdhdb.driver.reserved[4]:            0 ; 0x018: 0x00000000
kfdhdb.driver.reserved[5]:            0 ; 0x01c: 0x00000000
kfdhdb.compat:                186646528 ; 0x020: 0x0b200000
kfdhdb.dsknum:                        0 ; 0x024: 0x0000
kfdhdb.grptyp:                        1 ; 0x026: KFDGTP_EXTERNAL
kfdhdb.hdrsts:                        3 ; 0x027: KFDHDR_MEMBER
kfdhdb.dskname:             DATADG_0000 ; 0x028: length=11
kfdhdb.grpname:                  DATADG ; 0x048: length=6
kfdhdb.fgname:              DATADG_0000 ; 0x068: length=11
kfdhdb.capname:                         ; 0x088: length=0
--创建时间
kfdhdb.crestmp.hi:             33019607 ; 0x0a8: HOUR=0x17 DAYS=0x16 MNTH=0x5 YEAR=0x7df
kfdhdb.crestmp.lo:           3625030656 ; 0x0ac: USEC=0x0 MSEC=0x65 SECS=0x1 MINS=0x36
kfdhdb.mntstmp.hi:             33020841 ; 0x0b0: HOUR=0x9 DAYS=0x1d MNTH=0x6 YEAR=0x7df
kfdhdb.mntstmp.lo:           1021878272 ; 0x0b4: USEC=0x0 MSEC=0x228 SECS=0xe MINS=0xf
kfdhdb.secsize:                    512 ; 0x0b8: 0x0200
kfdhdb.blksize:                   4096 ; 0x0ba: 0x1000
kfdhdb.ausize:                 1048576 ; 0x0bc: 0x00100000
kfdhdb.mfact:                   113792 ; 0x0c0: 0x0001bc80
kfdhdb.dsksize:                   4096 ; 0x0c4: 0x00001000
kfdhdb.pmcnt:                        2 ; 0x0c8: 0x00000002
kfdhdb.fstlocn:                      1 ; 0x0cc: 0x00000001
kfdhdb.altlocn:                      2 ; 0x0d0: 0x00000002
kfdhdb.flblocn:                      2 ; 0x0d4: 0x00000002
```

Important ASM metadata block 0 header data

Data type	Value
kfbh.endian	System endianness. 0 - big endian, 1 - little endian.
kfbh.type	ASM block type. **KFBTYP_DISKHEAD** tells us this is an ASM disk header block.
kfbh.block.blk	ASM block number. Note the ASM disk header is **block number 0**.

Important ASM disk header specific data

Data type	Value
kfdhdb.driver.provstr	ORCLDISK+[ASM disk name] for ASMLIB disks. ORCLDISK for non-ASMLIB disks.
kfdhdb.dsknum	**ASM disk number**.
kfdhdb.grptyp	Disk group redundancy. KFDGTP_EXTERNAL - external, KFDGTP_NORMAL - normal, KFDGTP_HIGH - high.
kfdhdb.hdrsts	ASM disk header status. For possible values see V$ASM_DISK.HEADER_STATUS.
kfdhdb.dskname	**ASM disk name**.
kfdhdb.grpname	**ASM disk group name**.
kfdhdb.fgname	**ASM failgroup name**.
kfdhdb.crestmp.hi|lo	The date and time disk was added to the disk group.
kfdhdb.mntstmp.hi|lo	Last time the disk was mounted.
kfdhdb.secsize	Disk sector size (bytes).
kfdhdb.blksize	ASM **metadata block size** (bytes).
kfdhdb.ausize	**Alloocation unit size** (bytes). 1 MB is the default allocation unit size.
kfdhdb.dsksize	**Disk size** (**allocation units**). In this case the disk size is 10239 MB.
kfdhdb.fstlocn	Pointer to ASM Free Space Table. 1 = ASM block 1 in this allocation unit.
kfdhdb.altlocn	Pointer to ASM Allocation Table. 2 = ASM block 2 in this allocation unit.
kfdhdb.f1b1locn	Pointer to ASM File Directory. 2 = allocation unit 2.
kfdhdb.dbcompat	Minimum database version. 0x0a100000 = 10.1.
kfdhdb.grpstmp.hi|lo	The date and time the disk group was created.
kfdhdb.vfstart|vfend	Start and end allocation unit number for the **clusterware voting disk**. If this is zero, the disk does not have voting disk data. Version 11.2 and later only.
kfdhdb.spfile	Allocation unit number of the ASM spfile. Version 11.2 and later only.

	$ kfed read /dev/asm-grid aun=363
kfdhdb.spfflg	ASM spfile flag. If this is 1, the ASM spfile is on this disk in allocation unit kfdhdb.spfile. Version 11.2 and later only.

In ASM versions 11.1.0.7 and later, the ASM disk header block is backed up in the second last ASM metadata block in the allocation unit 1. To work out the second last block number we need to know the allocation unit size and ASM metadata block size.

I talked about this in my post on kfed, but let's do that again - get those values from the block header and calculate the second last block number in allocation unit 1:

$ ausize=`kfed read /dev/oracleasm/disks/ASMD1 | grep ausize | tr -s ' ' | cut -d' ' -f2`
$ blksize=`kfed read /dev/oracleasm/disks/ASMD1 | grep blksize | tr -s ' ' | cut -d' ' -f2`
$ let n=$ausize/$blksize-2
$ echo $n
254

Kfed 默认是按 1M 来计算，非 1M 的 AU，需要指定 AUSZ 参数：

[grid@rac11g1 ~]$ kfed read /dev/asm-data3 ausz=4m aun=1 blkn=1022 | more

2.1.2. Free Space Table (FST)

FST 的信息都的存在固定的位置，如下：

[grid@oracle11 ~]$ kfed read /dev/asm-data1 | grep fstlocn

```
kfdhdb.fstlocn:                          1 ; 0x0cc: 0x00000001
```

The FST indicates which ATBs might contain free allocation units. ASM consults a disk's FST when that disk is selected for allocation.

The Free Space Table occupies the second block (block 1) of each AT.

[grid@oracle11 ~]$ kfed read /dev/asm-data1 aun=0 blkn=1 | more

```
kfbh.endian:                             1 ; 0x000: 0x01
kfbh.hard:                             130 ; 0x001: 0x82
kfbh.type:                               2 ; 0x002: KFBTYP_FREESPC
kfbh.datfmt:                             2 ; 0x003: 0x02
kfbh.block.blk:                          1 ; 0x004: blk=1
kfbh.block.obj:               2147483648 ; 0x008: disk=0
......
kfdfsb.flag:                             1 ; 0x00a: B=1
```

```
kfdfsb.ublspare:                  0 ; 0x00b: 0x00
kfdfsb.spare[0]:                  0 ; 0x00c: 0x00000000
kfdfsb.spare[1]:                  0 ; 0x010: 0x00000000
kfdfsb.spare[2]:                  0 ; 0x014: 0x00000000
kfdfse[0].fse:                    0 ; 0x018: FREE=0x0 FRAG=0x0
```

For this FST block, the first allocation table block is in AU 0:

```
kfdfsb.aunum:                     0 ; 0x000: 0x00000000
```

Maximum number of the FST entries this block can hold is 254:

```
kfdfsb.max:                     254 ; 0x004: 0x00fe
```

How many Free Space Tables

Large ASM disks may have more than one stride. The field kfdhdb.mfact in the ASM disk header, shows the stride size - expressed in allocation units. Each stride will have its own physically addressed metadata, which means that it will have its own Free Space Table.

The second stride will have its physically addressed metadata in the first AU of the stride. Let's have a look.

```
$ kfed read /dev/sdc1 | grep mfact
kfdhdb.mfact:                   113792 ; 0x0c0: 0x0001bc80
```

This shows the stride size is 113792 AUs. Let's check the FST for the second stride. That should be in block 1 in AU113792.

```
$ kfed read /dev/sdc1 aun=113792 blkn=1 | grep type
kfbh.type:                          2 ; 0x002: KFBTYP_FREESPC
```

As expected, we have another FTS in AU113792. If we had another stride, there would be another FST at the beginning of that stride. As it happens, I have a large disk, with few strides, so we see the FST at the beginning at the third stride as well:

```
$ kfed read /dev/sdc1 aun=227584 blkn=1 | grep type
kfbh.type:                          2 ; 0x002: KFBTYP_FREESPC
```

从上面信息我们可以知道，FST 信息存在第一个 AU 中，由于 asm 中，block 编号都是从第 0 个开始，所以可以使用 kfed 直接查看元数据信息，不过这里第 0 个 block 的数据并不是 FST 的信息，而是 Partnership and Status Table (PST)信息。顾名思义，PST，就是存放 diskgroup 中 disk 的关系以及 disk 状态灯信息，下面通过 kfed 来查看详细详细：

```
[grid@oracle11 ~]$ kfed read /dev/asm-data1 aun=1 blkn=0 | grep kfbh.type
kfbh.type:                            17 ; 0x002: KFBTYP_PST_META

[grid@oracle11 ~]$ kfed read /dev/asm-data1 aun=1 blkn=1 | grep kfbh.type
kfbh.type:                            17 ; 0x002: KFBTYP_PST_META

[grid@oracle11 ~]$ kfed read /dev/asm-data1 aun=1 blkn=2 | grep kfbh.type
kfbh.type:                            18 ; 0x002: KFBTYP_PST_DTA

[grid@oracle11 ~]$ kfed read /dev/asm-data1 aun=1 blkn=3 | grep kfbh.type
kfbh.type:                            18 ; 0x002: KFBTYP_PST_DTA

[grid@oracle11 ~]$ kfed read /dev/asm-data1 aun=1 blkn=4 | grep kfbh.type
kfbh.type:                            13 ; 0x002: KFBTYP_PST_NONE
```

从上面信息可以看出，在前面 2 个 block 是存的 pst 元数据，而 blk 2,3 则是存放的 pst 数据。

2.1.3. Allocation Table (AT)

Every ASM disk contains at least one Allocation Table (AT) that describes the contents of the disk. The AT has one entry for every allocation unit (AU) on the disk. If an AU is allocated, the Allocation Table will have the extent number and the file number the AU belongs to.

- Finding the Allocation Table

The location of the first block of the Allocation Table is stored in the ASM disk header (field kfdhdb.altlocn). In the following example, the look up of that field shows that the AT starts at block 2.

```
[grid@rac11g1 ~]$ kfed read /dev/asm-data1 aun=0 blkn=0 | grep kfdhdb.altlocn
```

```
kfdhdb.altlocn:                  2 ; 0x0d0: 0x00000002
```

[grid@rac11g1 ~]$ kfed read /dev/asm-data1 blkn=2 | more

```
kfbh.endian:                     1 ; 0x000: 0x01
kfbh.hard:                     130 ; 0x001: 0x82
kfbh.type:                       3 ; 0x002: KFBTYP_ALLOCTBL
kfbh.datfmt:                     2 ; 0x003: 0x02
kfbh.block.blk:                  2 ; 0x004: blk=2
kfbh.block.obj:         2147483648 ; 0x008: disk=0
kfbh.check:             2183614523 ; 0x00c: 0x8227503b
kfbh.fcn.base:               53559 ; 0x010: 0x0000d137
kfbh.fcn.wrap:                   0 ; 0x014: 0x00000000
kfbh.spare1:                     0 ; 0x018: 0x00000000
kfbh.spare2:                     0 ; 0x01c: 0x00000000
kfdatb.aunum:                    0 ; 0x000: 0x00000000
kfdatb.shrink:                 448 ; 0x004: 0x01c0
```

The kfdatb.aunum=0, means that AU0 is the first AU described by this AT block. The kfdatb.shrink=448 means that this AT block can hold the information for 448 AUs. In the next AT block we should see kfdatb.aunum=448, meaning that it will have the info for AU448 + 448 more AUs. Let’s have a look:

[grid@rac11g1 ~]$ kfed read /dev/asm-data1 blkn=2 | grep kfdatb.aunum

```
kfdatb.aunum:                    0 ; 0x000: 0x00000000
```

[grid@rac11g1 ~]$ kfed read /dev/asm-data1 blkn=3 | grep kfdatb.aunum

```
kfdatb.aunum:                  448 ; 0x000: 0x000001c0
```

[grid@rac11g1 ~]$ kfed read /dev/asm-data1 blkn=4 | grep kfdatb.aunum

```
kfdatb.aunum:                  896 ; 0x000: 0x00000380
```

- Allocation table entries

For allocated AUs, the Allocation Table entry (kfdate[i]) holds the extent number, file number and the state of the allocation unit - normally allocated (flag V=1), vs a free or unallocated AU (flag V=0).

[grid@rac11g1 ~]$ kfed read /dev/asm-data1 blkn=3 | more

```
kfbh.endian:                          1 ; 0x000: 0x01
kfbh.hard:                          130 ; 0x001: 0x82
kfbh.type:                            3 ; 0x002: KFBTYP_ALLOCTBL
kfbh.datfmt:                          2 ; 0x003: 0x02
kfbh.block.blk:                       3 ; 0x004: blk=3
kfbh.block.obj:              2147483648 ; 0x008: disk=0
......
kfdatb.aunum:                       448 ; 0x000: 0x000001c0
kfdatb.shrink:                      448 ; 0x004: 0x01c0
kfdatb.ub2pad:                        0 ; 0x006: 0x0000
......
kfdatb.spare:                         0 ; 0x024: 0x00000000
kfdate[0].discriminator:              1 ; 0x028: 0x00000001
kfdate[0].allo.lo:                   44 ; 0x028: XNUM=0x2c
kfdate[0].allo.hi:              8388932 ; 0x02c: V=1 I=0 H=0 FNUM=0x144
......
kfdate[214].discriminator:            1 ; 0x6d8: 0x00000001
kfdate[214].allo.lo:                470 ; 0x6d8: XNUM=0x1d6
kfdate[214].allo.hi:            8388932 ; 0x6dc: V=1 I=0 H=0 FNUM=0x144
```

The excerpt shows the Allocation Table entries for file 324(hexadecimal FNUM=0x144), which start at kfdate[0] and end at kfdate[214]. That shows the ASM file 324has the total of 215 AUs. The AU numbers will be the index of kfdate[i] + offset (kfdatb.aunum=448). In other words, 0+448=448, 1+448=449... 214+448=662. Let's verify that by querying X$KFFXP:

SQL> SELECT FILE_ID, FILE_NAME FROM DBA_DATA_FILES;

```
   FILE_ID FILE_NAME
---------- --------------------------------------------------------------------------------
         4 +DATA_DG/acct/datafile/users.282.893580521
         3 +DATA_DG/acct/datafile/undotbs1.290.893580521
         2 +DATA_DG/acct/datafile/sysaux.324.893580521    --文件为 324
         1 +DATA_DG/acct/datafile/system.295.893580519
         5 +DATA_DG/acct/datafile/undotbs2.303.893580817
         6 +DATA_DG/acct/datafile/sky.366.893598667


6 rows selected.
```

[grid@rac11g1 ~]$ kfed read /dev/asm-data1 blkn=2 | more

```
kfdatb.aunum:                         0 ; 0x000: 0x00000000
kfdatb.shrink:                      448 ; 0x004: 0x01c0
kfdate[426].allo.hi:            8388932 ; 0xd7c: V=1 I=0 H=0 FNUM=0x144
......
kfdate[447].allo.hi:            8388932 ; 0xe24: V=1 I=0 H=0 FNUM=0x144
```

数据文件 AU 位置。426 - 447

[grid@rac11g1 ~]$ kfed read /dev/asm-data1 blkn=3 | grep 'FNUM=0x144'

```
kfdatb.aunum:                       448 ; 0x000: 0x000001c0
kfdatb.shrink:                      448 ; 0x004: 0x01c0
......
kfdate[0].allo.hi:              8388932 ; 0x02c: V=1 I=0 H=0 FNUM=0x144
kfdate[1].allo.hi:              8388932 ; 0x034: V=1 I=0 H=0 FNUM=0x144
......
kfdate[404].allo.hi:            8388932 ; 0xccc: V=1 I=0 H=0 FNUM=0x144
kfdate[405].allo.hi:            8388932 ; 0xcd4: V=1 I=0 H=0 FNUM=0x144
```

数据文件 AU 位置。(0+448) - (405+448) => 448 - 853

SQL> SELECT AU_KFFXP FROM X$KFFXP
WHERE GROUP_KFFXP=1 AND NUMBER_KFFXP=324 ORDER BY 1;

```
 AU_KFFXP
----------
       426
......
      4997
```

http://asmsupportguy.blogspot.com.au/2013/08/allocation-table.html

2.1.4. Partnership and Status Table (PST)

The PST tracks diskgroup membership and the disk partnerships. ASM consults the PST to determine whether a sufficient set of disks is online to mount the diskgroup.

Each disk in a diskgroup reserves the second allocation unit (AU 1) for the PST.

The number of PSTs is determined by the diskgroup redundancy and the number of failure groups.

External redundancy diskgroups have a single PST. Normal-redundancy diskgroups contain three PSTs if at least three failure groups exist. Otherwise, normal-redundancy diskgroups have one PST per failure group. High-redundancy diskgroups have five PSTs if sufficient failure groups exist. Otherwise, high-redundancy diskgroups have one PST per failure group.

[grid@oracle11 ~]$ **kfed read /dev/asm-data1 aun=1 blkn=0 | more**

```
kfbh.endian:                          1 ; 0x000: 0x01
kfbh.hard:                          130 ; 0x001: 0x82   ---这里是 HARD magic number
kfbh.type:                           17 ; 0x002: KFBTYP_PST_META   ---这表示元数据 block 类型
kfbh.datfmt:                          2 ; 0x003: 0x02   ---表示元数据 block 格式
kfbh.block.blk:                     256 ; 0x004: blk=256   ---表示 block 位置
kfbh.block.obj:              2147483648 ; 0x008: disk=0
kfbh.check:                  2709258679 ; 0x00c: 0xa17c01b7   ---主要是做校验用的，check 一致
性
kfbh.fcn.base:                        0 ; 0x010: 0x00000000
kfbh.fcn.wrap:                        0 ; 0x014: 0x00000000
kfbh.spare1:                          0 ; 0x018: 0x00000000
kfbh.spare2:                          0 ; 0x01c: 0x00000000
```

The first part of the PST contains information about the PST itself:

- version number
- timestamp
- PST size (number of disks)
- number of PST copies
- list of disks containing the PST
- value of compatible.asm (if COMPATIBLE.ASM >= 11.1)
- disk status (for example, whether it is online or offline)
- number of partners
- list of partners

The last block of the PST is reserved for the diskgroup heartbeat.ASM uses this block to prevent diskgroups from being simultaneously mounted in different clusters.一个 AU 大小为 1M，一个块为 4096 字节，因此一个 AU 中包含有 256 个块，块号从 0 开始。因此 AU1 的最后一个块的块号是 255。

[grid@oracle11 ~]$ kfed read /dev/asm-data1 aun=1 blkn=255 | more

```
kfbh.endian:                          1 ; 0x000: 0x01
kfbh.hard:                          130 ; 0x001: 0x82
kfbh.type:                           19 ; 0x002: KFBTYP_HBEAT
kfbh.datfmt:                          2 ; 0x003: 0x02
kfbh.block.blk:                     511 ; 0x004: blk=511
kfbh.block.obj:              2147483648 ; 0x008: disk=0
kfbh.check:                  1549052467 ; 0x00c: 0x5c54aa33
kfbh.fcn.base:                        0 ; 0x010: 0x00000000
kfbh.fcn.wrap:                        0 ; 0x014: 0x00000000
kfbh.spare1:                          0 ; 0x018: 0x00000000
kfbh.spare2:                          0 ; 0x01c: 0x00000000
kfdpHbeatB.instance:                  1 ; 0x000: 0x00000001
kfdpHbeatB.ts.hi:              33020998 ; 0x004: HOUR=0x6 DAYS=0x2 MNTH=0x7 YEAR=0x7df
kfdpHbeatB.ts.lo:             781844480 ; 0x008: USEC=0x0 MSEC=0x280 SECS=0x29 MINS=0xb
kfdpHbeatB.rnd[0]:           3899403624 ; 0x00c: 0xe86c2d68
kfdpHbeatB.rnd[1]:           4065393526 ; 0x010: 0xf250fb76
kfdpHbeatB.rnd[2]:             75023373 ; 0x014: 0x0478c40d
kfdpHbeatB.rnd[3]:           4017022873 ; 0x018: 0xef6ee799
```

2.2. 虚拟元数据

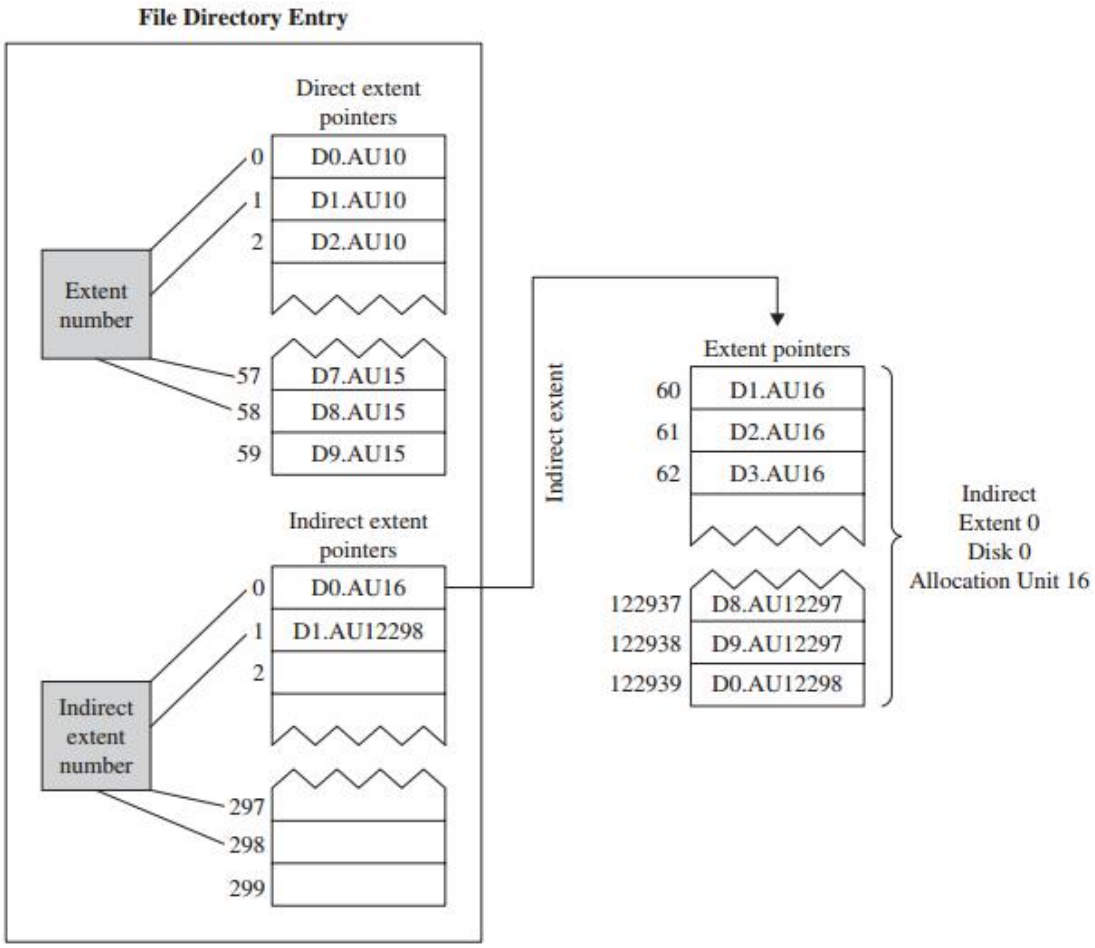
ASM virtual metadata are stored in ASM files. Directories are the metadata files that are accessed exclusively by the ASM instance. File numbers for directories begin at 1. Registry numbers count down from 255. File numbers that are not mentioned in this section are reserved for future use.

The following structures comprise the virtual metadata:

- File Directory
- Disk Directory
- Active Change Directory (ACD)
- Continuing Operations Directory (COD)
- Template Directory
- Alias Directory
- Attribute Directory
- Staleness Directory

- Staleness Registry

2.2.1. oFile directory (文件目录)



The File Directory contains all of the metadata relevant to ASM files.

ASM 文件目录 File Directory 针对本 Disk Group 中的每一个文件包含一条记录。该记录指向该文件的前 60 个数据盘区 extents，必要时还包括间接盘区 indirect extents。该文件目录在必要容纳更多文件数目时会自动增长。每一个文件目录记录保持更新以下文件信息：

- 文件大小
- 该文件的块大小
- 文件种类，例如：数据文件，ASM 元数据文件，在线日志，归档日志，控制文件等等
- 文件冗余度：外部、2 路或者 3 路镜像
- 条带化配置，coarse or fine
- 到前 60 个 extent 的直接盘区指针(direct extent pointer)

- 300 个间接盘区指针(indirect extent pointers)
- 创建时间戳
- 最后修改或更新时间戳
- 指向别名目录中的用户别名和文件名

文件号 file number 是文件目录中找到对应文件记录的重要索引键。其中第一条记录是该文件目录自身。为了找出过期的文件号，所以在每个文件创建时都生成了一个唯一的 32 bit 的识别号 incarnation number。由此，disk group 的 ID+ file number + 该 incarnation number 可以做到唯一识别某个指定文件。

The block size for ASM files is independent of the ASM metadata block size.

请注意，约定俗成地将 ASM 文件的第一个 block 称为 0 号块 - block zero。 0 号块通常包含十分重要的接口信息。

为了找出 file directory 所在 AU 的位置，我们需要使用 kfed 工具浏览 ASM disk header 磁盘头部 0 号 AU 中的 kfdhdb.f1b1locn 信息，例如我们使用 kfed 查看 asm disk /dev/asm-data1 上的信息：

```
[root@oracle11 ~]# kfed read /dev/asm-data1 aun=0 | grep 'kfdhdb.f1b1locn'
kfdhdb.f1b1locn:                          2 ; 0x0d4: 0x00000002
```

或者通过 ASM 实例的数据字典 X$KFFXP 进行查询:

```
SQL>
SELECT xnum_kffxp "Virtual extent", pxn_kffxp "Physical extent", au_kffxp
"Allocation unit", disk_kffxp "Disk"
  FROM x$kffxp
 WHERE group_kffxp = 1 -- Diskgroup 1 (DATA)
   AND number_kffxp = 1 -- File 1 (file directory)
 ORDER BY 1, 2;

Virtual extent Physical extent Allocation unit       Disk
-------------- --------------- --------------- ----------
             0               0               2          0   --> ASM 源数据文件使用
             1               1               3          0   --> ASM 文件使用
             2               2            2485          0
```

With the allocation unit size of 1MB and the ASM metadata block size of 4KB, one allocation unit can hold up to 256 directory entries. As numbers 1-255 are reserved for the ASM metadata files, extent 0 will only have enough room for the ASM metadata files. Extent 1 will hold information about next 256 files managed by the ASM and so on.

SQL>

```
SELECT NAME "File", block_size "Block size", block_size * (file_size_blks + 1)
"File size" FROM v$controlfile;
```

```
File                                                    Block size  File size
------------------------------------------------------- ---------- ----------
+DATA_DG/acct/controlfile/current.306.893580631               16384   20627456
```

ASM 文件编号为 306(current.306.893580631). First, we query the X$KFFXP view in the ASM instance, to get the extent and AU distribution:

SQL>

```
SELECT xnum_kffxp "Virtual extent", pxn_kffxp "Physical extent", au_kffxp
"Allocation unit", disk_kffxp "Disk"
  FROM x$kffxp
 WHERE group_kffxp = 1 -- Diskgroup 1
   AND number_kffxp = 306 -- File 306 (control file)
   AND xnum_kffxp <> 2147483648
 ORDER BY 1, 2;
```

```
Virtual extent Physical extent Allocation unit       Disk
-------------- --------------- --------------- ----------
             0               0             679          0
             1               1            3997          0
             2               2             680          0
             3               3            3998          0
             4               4             681          0
             5               5            3999          0
             6               6             682          0
             7               7            4000          0
             8               8             683          0
             9               9            4001          0
            10              10             684          0
            11              11            4002          0
            12              12             685          0
            13              13            4003          0
            14              14             686          0
            15              15            4004          0
            16              16             687          0
            17              17            4005          0
```

```
        18         18        688          0
        19         19       4006          0
        20         20        689          0
        21         21       4007          0
        22         22        690          0
        23         23       4008          0
```

24 rows selected.

SQL> SELECT disk_number, path FROM v$asm_disk WHERE group_number = 1 ORDER BY 1;

```
DISK_NUMBER PATH
----------- ------------------------------
          0 /dev/asm-data1
```

ASM 文件 306(current.306.893580631)的文件目录在 ASM 文件目录的虚拟扩展 1 中存放，存放的块位置为(306-256) = 50， 因为前 256 个块是 ASM 源数据文件目录信息。

```
Virtual extent Physical extent Allocation unit       Disk
-------------- --------------- --------------- ----------
             0               0               2          0   --> ASM 源数据文件使用
             1               1               3          0   --> ASM 文件使用
```

[grid@rac11g1 ~]$ kfed read /dev/asm-data1 aun=3 blkn=50 | more

```
kfbh.endian:                          1 ; 0x000: 0x01
kfbh.hard:                          130 ; 0x001: 0x82
kfbh.type:                            4 ; 0x002: KFBTYP_FILEDIR
kfbh.datfmt:                          1 ; 0x003: 0x01
kfbh.block.blk:                     306 ; 0x004: blk=306
kfbh.block.obj:                       1 ; 0x008: file=1
kfbh.check:                  3450573143 ; 0x00c: 0xcdab9157
kfbh.fcn.base:                   70004 ; 0x010: 0x00011174
kfbh.fcn.wrap:                        0 ; 0x014: 0x00000000
kfbh.spare1:                          0 ; 0x018: 0x00000000
kfbh.spare2:                          0 ; 0x01c: 0x00000000
kfffdb.node.incarn:           893580631 ; 0x000: A=1 NUMM=0x1aa17aab
kfffdb.node.frlist.number:   4294967295 ; 0x004: 0xffffffff
kfffdb.node.frlist.incarn:            0 ; 0x008: A=0 NUMM=0x0
```

```
kfffdb.hibytes:                 0 ; 0x00c: 0x00000000
kfffdb.lobytes:          20627456 ; 0x010: 0x013ac000
kfffdb.xtntcnt:                24 ; 0x014: 0x00000018
kfffdb.xtnteof:                24 ; 0x018: 0x00000018
kfffdb.blkSize:             16384 ; 0x01c: 0x00004000
kfffdb.flags:                  19 ; 0x020: O=1 S=1 S=0 D=0 C=1 I=0 R=0 A=0
kfffdb.fileType:                1 ; 0x021: 0x01
kfffdb.dXrs:                   17 ; 0x022: SCHE=0x1 NUMB=0x1
kfffdb.iXrs:                   17 ; 0x023: SCHE=0x1 NUMB=0x1
kfffdb.dXsiz[0]:       4294967295 ; 0x024: 0xffffffff
kfffdb.dXsiz[1]:                0 ; 0x028: 0x00000000
kfffdb.dXsiz[2]:                0 ; 0x02c: 0x00000000
kfffdb.iXsiz[0]:       4294967295 ; 0x030: 0xffffffff
kfffdb.iXsiz[1]:                0 ; 0x034: 0x00000000
kfffdb.iXsiz[2]:                0 ; 0x038: 0x00000000
kfffdb.xtntblk:                24 ; 0x03c: 0x0018
kfffdb.break:                  60 ; 0x03e: 0x003c
kfffdb.priZn:                   0 ; 0x040: KFDZN_COLD
kfffdb.secZn:                   0 ; 0x041: KFDZN_COLD
kfffdb.ub2spare:                0 ; 0x042: 0x0000
kfffdb.alias[0]:              477 ; 0x044: 0x000001dd
kfffdb.alias[1]:       4294967295 ; 0x048: 0xffffffff
kfffdb.strpwdth:                8 ; 0x04c: 0x08
kfffdb.strpsz:                 17 ; 0x04d: 0x11
kfffdb.usmsz:                   0 ; 0x04e: 0x0000
kfffdb.crets.hi:         33024648 ; 0x050: HOUR=0x8 DAYS=0x14 MNTH=0xa YEAR=0x7df
kfffdb.crets.lo:       3388694528 ; 0x054: USEC=0x0 MSEC=0x2d8 SECS=0x1f MINS=0x32
kfffdb.modts.hi:         33025328 ; 0x058: HOUR=0x10 DAYS=0x9 MNTH=0xb YEAR=0x7df
kfffdb.modts.lo:                0 ; 0x05c: USEC=0x0 MSEC=0x0 SECS=0x0 MINS=0x0
kfffdb.dasz[0]:                 0 ; 0x060: 0x00
kfffdb.dasz[1]:                 0 ; 0x061: 0x00
kfffdb.dasz[2]:                 0 ; 0x062: 0x00
kfffdb.dasz[3]:                 0 ; 0x063: 0x00
kfffdb.permissn:                0 ; 0x064: 0x00
kfffdb.ub1spar1:                0 ; 0x065: 0x00
kfffdb.ub2spar2:                0 ; 0x066: 0x0000
kfffdb.user.entnum:             0 ; 0x068: 0x0000
```

```
kfffdb.user.entinc:                   0 ; 0x06a: 0x0000
kfffdb.group.entnum:                  0 ; 0x06c: 0x0000
kfffdb.group.entinc:                  0 ; 0x06e: 0x0000
kfffdb.spare[0]:                      0 ; 0x070: 0x00000000
......
kfffdb.usm:                             ; 0x0a0: length=0
kfffde[0].xptr.au:                  679 ; 0x4a0: 0x000002a7
kfffde[0].xptr.disk:                  0 ; 0x4a4: 0x0000
kfffde[0].xptr.flags:                 0 ; 0x4a6: L=0 E=0 D=0 S=0
kfffde[0].xptr.chk:                 143 ; 0x4a7: 0x8f
......
kfffde[17].xptr.au:                4005 ; 0x528: 0x00000fa5
kfffde[17].xptr.disk:                 0 ; 0x52c: 0x0000
kfffde[17].xptr.flags:                0 ; 0x52e: L=0 E=0 D=0 S=0
kfffde[17].xptr.chk:                128 ; 0x52f: 0x80
```

The first part of the kfed output (the kfbh fields) confirm this is an ASM file directory block (kfbh.type=KFBTYP_FILEDIR), for file 262 (kfbh.block.blk=262).

The second part of the kfed output (the kfffdb fields) shows:

File incarnation number (kfffdb.node.incarn=822925011), which is part of the file name

File size in bytes (kfffdb.lobytes=17973248)

Physical extent count (kfffdb.xtntcnt=72)

File block size in bytes (kfffdb.blkSize=16384)

File type (kfffdb.fileType=1), i.e. the database control file

The third part of the output (the kfffde fields) shows the physical extent distribution that agrees with the query output from X$KFFXP:

Physical extent 0 is in AU 679(kfffde[0].xptr.au=679), on disk 0 (kfffde[0].xptr.disk=0)

Physical extent 17 is in AU 4005(kfffde[17].xptr.au=4005), on disk 0 (kfffde[0].xptr.disk=0)

- **文件目录结构**

Allocation unit=2 的 block 1 描述了该 ASM 1 号文件 file directory 自身。该块的前部分包含了标准的头部信息，并显示该块的类型为 KFBTYP_FILEDIR。 在该 kfffdb 结构之后，该 file directory 的每一个 block 包含描述文件物理属性和盘区指针的信息， 以及指向所有间接盘区的指针。

以下是 aun=2 block=1 的 file directory 信息

[root@oracle11 ~]# **kfed read /dev/asm-data1 aun=2 blkn=1 | less > block.txt**

[root@oracle11 ~]# more block.txt

```
kfbh.endian:                          1 ; 0x000: 0x01
kfbh.hard:                          130 ; 0x001: 0x82
kfbh.type:                            4 ; 0x002: KFBTYP_FILEDIR
kfbh.datfmt:                          1 ; 0x003: 0x01
kfbh.block.blk:                       1 ; 0x004: blk=1
kfbh.block.obj:                       1 ; 0x008: file=1
kfbh.check:                  3623248765 ; 0x00c: 0xd7f6637d
kfbh.fcn.base:                      441 ; 0x010: 0x000001b9
kfbh.fcn.wrap:                        0 ; 0x014: 0x00000000
```

其中字段的含义：

```
KFBTYP_FILEDIR // block type = file directory block
kfffdb.node.incarn: File incarnation information
kfffdb.hibytes File size (high bytes)
kfffdb.lobyte 2097152 ; 0x010: 0x00200000 File size (low bytes) 2097152 ==》2MB 大小
kfffdb.xtntcnt: 6 ; 0x014: 0x00000006 // 6 extents for this file
kfffdb.xtnteof: 6 ; 0x018: 0x00000006 // 6 extents before eof
kfffdb.blkSize: 4096 ; 0x01c: 0x00001000 // 标准 ASM block 大小
kfffdb.flags: 1 ; 0x020: O=1 S=0 S=0 D=0 C=0 I=0 R=0 A=0
// Flag definitions
O - File is original, not snapshot
S - File is striped
S - Strict allocation policy
D - File is damaged
C - File creation is committed
I - File has empty indirect block
R - File has known at-risk value
```

A - The at-risk value itsefl

接下来看一个 ASM metadata 文件的实际目录记录，我们就查看 aun=2 的 blkn=4 (259-255)

[root@oracle11 ~]# **kfed read /dev/asm-data1 aun=2 blkn=4 | more**

```
kfbh.endian:                          1 ; 0x000: 0x01
kfbh.hard:                          130 ; 0x001: 0x82
kfbh.type:                            4 ; 0x002: KFBTYP_FILEDIR
kfbh.datfmt:                          1 ; 0x003: 0x01
kfbh.block.blk:                       4 ; 0x004: blk=4
kfbh.block.obj:                       1 ; 0x008: file=1
```

```
List of Permanent Datafiles
===========================
File Size(MB) Tablespace           RB segs Datafile Name
---- -------- -------------------- ------- ------------------------
1    750      SYSTEM               ***     +DATADG/sky/datafile/system.256.880415831
2    590      SYSAUX               ***     +DATADG/sky/datafile/sysaux.257.880415833
3    90       UNDOTBS1             ***     +DATADG/sky/datafile/undotbs1.258.880415833
4    5        USERS                ***     +DATADG/sky/datafile/users.259.880415833
```

其中字段的含义：

kfffdb.lobytes: 8331264 ; 0x010: 0x007f2000 ==>说明文件大小为 8331264bytes

kfffdb.xtntcnt: 24 ; 0x014: 0x00000018

kfffdb.xtnteof: 24 ; 0x018: 0x00000018 ==> 说明该文件目前共 24 个 extents

kfffdb.blkSize: 4096 ; 0x01c: 0x00001000 ==> 4k 的 ASM Block size

kfffdb.fileType: 15 ; 0x021: 0x0f filetype=15 说明是 ASM Metadata File

kfffde[0].xptr.au: 36 ; 0x4a0: 0x00000024 file number=4 的第一 extent 指向 36 号 AU

kfffde[0].xptr.disk: 1 ; 0x4a4: 0x0001 Disk number 1

kfffde[1].xptr.au: 45 ; 0x4a8: 0x0000002d file number=4 的第二 extent 指向 45 号 AU

kfffde[2].xptr.au: 4294967295 ; 0x4b0: 0xffffffff 若 kfffde[N].xptr.au=4294967295 说明该 FILE 没有更多 extent 了

AU 的指针情况， 可以这样查看:

[root@oracle11 ~]#

kfed read /dev/asm-data1 aun=2 blkn=4 | egrep "xptr.au|xptr.disk" | less

```
kfffde[0].xptr.au:                   46 ; 0x4a0: 0x0000002e
```

```
kfffde[0].xptr.disk:                  0 ; 0x4a4: 0x0000
kfffde[1].xptr.au:                   47 ; 0x4a8: 0x0000002f
kfffde[1].xptr.disk:                  0 ; 0x4ac: 0x0000
kfffde[2].xptr.au:                   52 ; 0x4b0: 0x00000034
kfffde[2].xptr.disk:                  0 ; 0x4b4: 0x0000
kfffde[3].xptr.au:                   53 ; 0x4b8: 0x00000035
kfffde[3].xptr.disk:                  0 ; 0x4bc: 0x0000
kfffde[4].xptr.au:                   54 ; 0x4c0: 0x00000036
kfffde[4].xptr.disk:                  0 ; 0x4c4: 0x0000
kfffde[5].xptr.au:                   55 ; 0x4c8: 0x00000037
kfffde[5].xptr.disk:                  0 ; 0x4cc: 0x0000
kfffde[6].xptr.au:                   56 ; 0x4d0: 0x00000038
kfffde[6].xptr.disk:                  0 ; 0x4d4: 0x0000
kfffde[7].xptr.au:                   57 ; 0x4d8: 0x00000039
kfffde[7].xptr.disk:                  0 ; 0x4dc: 0x0000
kfffde[8].xptr.au:           4294967295 ; 0x4e0: 0xffffffff
kfffde[8].xptr.disk:              65535 ; 0x4e4: 0xffff
```

```
SQL> SELECT DISK_KFFXP, AU_KFFXP, XNUM_KFFXP
  FROM X$KFFXP
 WHERE GROUP_KFFXP = 1
   AND NUMBER_KFFXP = 4;
```

```
DISK_KFFXP   AU_KFFXP XNUM_KFFXP
---------- ---------- ----------
         0         46          0
         0         47          1
         0         52          2
         0         53          3
         0         54          4
         0         55          5
         0         56          6
         0         57          7
```

- **找出数据文件对应的目录记录 directory entry**

SQL> SELECT GROUP_NUMBER, FILE_NUMBER, NAME FROM V$ASM_ALIAS GROUP BY GROUP_NUMBER, FILE_NUMBER, NAME;

```
GROUP_NUMBER FILE_NUMBER NAME
------------ ----------- ------------------------------
           1         256 SYSTEM.256.880415831
           1         257 SYSAUX.257.880415833
           1         258 UNDOTBS1.258.880415833
           1         259 USERS.259.880415833
           1         260 Current.260.880415929
           1         261 group_1.261.880415931
           1         262 group_2.262.880415931
           1         263 group_3.263.880415933
           1         264 TEMP.264.880415939
           1         265 spfilesky.ora
           1         265 spfile.265.880416171

GROUP_NUMBER FILE_NUMBER NAME
------------ ----------- ------------------------------
           1  4294967295 SKY
           1  4294967295 DATAFILE
           1  4294967295 TEMPFILE
           1  4294967295 ONLINELOG
           1  4294967295 CONTROLFILE
           1  4294967295 PARAMETERFILE
           2         253 REGISTRY.253.880414225
           2  4294967295 ASM
           2  4294967295 ASMPARAMETERFILE
```

知识总结：

asm disk 的前 50 个 AU(50MB)是为 asm metadata 保留的

ASM 的前 255 个 file number 是为 metadata file 保留的，文件号从 1 开始， file numner=1 的 1 号文件为 ASM 的 file directory

普通的 ASM File 的 file number 从 256 开始

ASM disk 的第二个 AU 即是 file number=1 的 file directory (非必然)，在 1MB AU 和 4096 bytes block 的情况下可以存放 255 个 file directory information，其 block type 为 KFBTYP_FILEDIR

普通 ASM FILE 的 directory entry 的位置，可以这样计算 File number=1 的 第 (file number-256)/256 +2 个 extent，blkn=mod(file number-256),256) ，例如文件号 258 =》 第二个 extent 的 blkn=2。

KFBTYP_FILEDIR 中从 kfffde[0].xptr.au 到 kfffde[59].xptr.au 是直接盘区指针 directly extent pointers， kfffde[60].xptr.au 以上是 KFBTYP_INDIRECT(kffixe)间接盘区指针 Indirectly extents pointers。

2.2.2. Disk Directory

Fields in a Disk Directory entry include the following:

- Disk name
- Failure group name
- Disk size
- Disk free space
- Disk creation time

Disk Header Block 中的信息并没有直接反应出来 Disk Directory 的信息，但是，你可以想象，既然 allocate table 元数据都在第 2 个 AU 里面，而那么必然 Disk Directory 信息也在该 AU 里面，因为进行在读取 allocate table 信息时，必然要先读取 disk directory。

```
[grid@oracle11 ~]$ kfed read /dev/asm-data1 aun=0 blkn=0 | grep f1b1locn
kfdhdb.f1b1locn:                          2 ; 0x0d4: 0x00000002
```

```
[grid@oracle11 ~]$ kfed read /dev/asm-data1 aun=2 blkn=2 | grep au | more
kfffde[0].xptr.au:                        3 ; 0x4a0: 0x00000003
kfffde[1].xptr.au:               4294967295 ; 0x4a8: 0xffffffff
```

```
[grid@oracle11 ~]$ kfed read /dev/asm-data1 aun=3 blkn=0 | more
kfbh.endian:                              1 ; 0x000: 0x01
kfbh.hard:                              130 ; 0x001: 0x82
kfbh.type:                                6 ; 0x002: KFBTYP_DISKDIR
kfbh.datfmt:                              1 ; 0x003: 0x01
kfbh.block.blk:                           0 ; 0x004: blk=0
kfbh.block.obj:                           2 ; 0x008: file=2
kfbh.check:                       816396623 ; 0x00c: 0x30a9394f
```

```
kfbh.fcn.base:                       5725 ; 0x010: 0x0000165d
kfbh.fcn.wrap:                          0 ; 0x014: 0x00000000
kfbh.spare1:                            0 ; 0x018: 0x00000000
kfbh.spare2:                            0 ; 0x01c: 0x00000000
--目录结构信息
kffdnd.bnode.incarn:                    1 ; 0x000: A=1 NUMM=0x0
kffdnd.bnode.frlist.number:  4294967295 ; 0x004: 0xffffffff
kffdnd.bnode.frlist.incarn:             0 ; 0x008: A=0 NUMM=0x0
kffdnd.overfl.number:        4294967295 ; 0x00c: 0xffffffff
kffdnd.overfl.incarn:                   0 ; 0x010: A=0 NUMM=0x0
kffdnd.parent.number:                   0 ; 0x014: 0x00000000
kffdnd.parent.incarn:                   1 ; 0x018: A=1 NUMM=0x0
kffdnd.fstblk.number:                   0 ; 0x01c: 0x00000000
kffdnd.fstblk.incarn:                   1 ; 0x020: A=1 NUMM=0x0

kfddde[0].entry.incarn:                 1 ; 0x024: A=1 NUMM=0x0
kfddde[0].entry.hash:                   0 ; 0x028: 0x00000000
kfddde[0].entry.refer.number:4294967295 ; 0x02c: 0xffffffff
kfddde[0].entry.refer.incarn:           0 ; 0x030: A=0 NUMM=0x0
kfddde[0].dsknum:                       0 ; 0x034: 0x0000
kfddde[0].state:                        2 ; 0x036: KFDSTA_NORMAL
#define KFDSTA_INVALID  ((kfdsta)0)  /* Illegal value */
#define KFDSTA_UNKNOWN  ((kfdsta)1)  /* ASM disk state not known */
#define KFDSTA_NORMAL   ((kfdsta)2)  /* Happy disk */
#define KFDSTA_UNUSED   ((kfdsta)3)  /* Unused State - Open */
#define KFDSTA_DROPPING ((kfdsta)4)  /* Disk being dropped from group */
#define KFDSTA_HUNG     ((kfdsta)5)  /* Disk drop operation hung */
#define KFDSTA_FORCING  ((kfdsta)6)  /* Disk beinng drop forced */
#define KFDSTA_DROPPED  ((kfdsta)7)  /* Disk no longer part of group */
#define KFDSTA_ADDING   ((kfdsta)8)  /* Disk being globally validated */

kfddde[0].ddchgfl:                    132 ; 0x037: 0x84
kfddde[0].dskname:           DATADG_0000 ; 0x038: length=11
kfddde[0].fgname:            DATADG_0000 ; 0x058: length=11
kfddde[0].crestmp.hi:           33019607 ; 0x078: HOUR=0x17 DAYS=0x16 MNTH=0x5 YEAR=0x7df
kfddde[0].crestmp.lo:         3625030656 ; 0x07c: USEC=0x0 MSEC=0x65 SECS=0x1 MINS=0x36
kfddde[0].failstmp.hi:                  0 ; 0x080: HOUR=0x0 DAYS=0x0 MNTH=0x0 YEAR=0x0
kfddde[0].failstmp.lo:                  0 ; 0x084: USEC=0x0 MSEC=0x0 SECS=0x0 MINS=0x0
kfddde[0].timer:                        0 ; 0x088: 0x00000000
kfddde[0].size:                      4096 ; 0x08c: 0x00001000
```

SQL>

```
SELECT x.xnum_kffxp "Extent", x.au_kffxp "AU", x.disk_kffxp "Disk #", d.name "Disk name"
  FROM x$kffxp x, v$asm_disk_stat d
 WHERE x.group_kffxp = d.group_number
   AND x.disk_kffxp = d.disk_number
   AND x.group_kffxp = 1
   AND x.number_kffxp = 2
 ORDER BY 1, 2;
```

```
    Extent         AU     Disk # Disk name
---------- ---------- ---------- ------------------------------------------------------------

         0          4          0 DATA_DG_0000
```

[grid@rac11g1 ~]$ kfed read /dev/asm-data1 aun=4 | more

```
kfbh.endian:                          1 ; 0x000: 0x01
kfbh.hard:                          130 ; 0x001: 0x82
kfbh.type:                            6 ; 0x002: KFBTYP_DISKDIR
kfbh.datfmt:                          1 ; 0x003: 0x01
kfbh.block.blk:                       0 ; 0x004: blk=0
kfbh.block.obj:                       2 ; 0x008: file=2
```

2.2.3. Active Change Directory(ACD)

大家如何把 asm 实例也看成一个微型的数据库实例的话，那么 ACD 信息，就好比是 redo。换句话将，ACD 里面的信息，记录了 asm 的所有元数据 block 的操作记录 (The ACD is a log that allows ASM to make atomic changes to multiple data structures. It is similar to the redo logs used by the Oracle RDBMS.)。

The first block of each chunk contains the open/close status and checkpoint. The checkpoint is updated every 3 sec.

The ACD is file number three in every diskgroup.

When the ASM instance needs to make an atomic change to multiple metadata blocks, a log record is written into the ASM active change directory (ACD), which is the ASM metadata file number 3. These log records are written in a single I/O.

The ACD is divided into chunks or threads, and each running ASM instance has its own 42 MB chunk. When a disk group is created, a single chunk is allocated for the ACD. As more instances mount the disk group, the ACD grows (by 42 MB) to accommodate every running instance with its own ACD chunk.

The ACD components are:

- ACDC - ACD checkpoint

- ABA - ACD block address
- LGE - ACD redo log record
- BCD - ACD block change descriptor

SQL>

```
SELECT x.xnum_kffxp "Extent", x.au_kffxp "AU", x.disk_kffxp "Disk #", d.name "Disk name"
  FROM x$kffxp x, v$asm_disk_stat d
 WHERE x.group_kffxp = d.group_number
   AND x.disk_kffxp = d.disk_number
   AND x.group_kffxp = 1
   AND x.number_kffxp = 3
 ORDER BY 1, 2;
```

```
    Extent         AU     Disk # Disk name
---------- ---------- ---------- ------------------------------------------------------------
         0          5          0 DATA_DG_0000
         1       2486          0 DATA_DG_0000
         2          7          0 DATA_DG_0000
         3       2487          0 DATA_DG_0000
```

```
SELECT NUMBER_KFFXP "ASM file number",
       DECODE(NUMBER_KFFXP,
              1,
              'File directory',
              2,
              'Disk directory',
              3,
              'Active change directory',
              4,
              'Continuing operations directory',
              5,
              'Template directory',
              6,
              'Alias directory',
              7,
              'AVD volume file directory',
              8,
              'Disk free space directory',
              9,
              'Attributes directory',
              10,
              'ASM user directory',
              11,
              'ASM user group directory',
              12,
```

```
            'Staleness directory') "ASM metadata file name",
       COUNT(AU_KFFXP) "Allocation units"
  FROM X$KFFXP
 WHERE GROUP_KFFXP = 1 -- disk group 1
   AND NUMBER_KFFXP < 17 -- ASM metadata files
   AND DISK_KFFXP <> 65534 -- ignore disk number 65534
 GROUP BY NUMBER_KFFXP;
```

ASM file number	ASM metadata file name	Allocation units
1	File directory	2
2	Disk directory	1
3	Active change directory	42
4	Continuing operations directory	8
5	Template directory	1
6	Alias directory	1
8	Disk free space directory	1
9	Attributes directory	1

[grid@rac11g1 ~]$ kfed read /dev/asm-data1 aun=2 blkn=3 | more

```
kfffde[0].xptr.au:                5 ; 0x4a0: 0x00000005
kfffde[0].xptr.disk:              0 ; 0x4a4: 0x0000
kfffde[0].xptr.flags:             0 ; 0x4a6: L=0 E=0 D=0 S=0
kfffde[0].xptr.chk:              47 ; 0x4a7: 0x2f
```

[grid@rac11g1 ~]$ kfed read /dev/asm-data1 aun=5 blkn=0 | more

```
kfbh.endian:                      1 ; 0x000: 0x01
kfbh.hard:                      130 ; 0x001: 0x82
kfbh.type:                        7 ; 0x002: KFBTYP_ACDC
kfbh.datfmt:                      1 ; 0x003: 0x01
kfbh.block.blk:                   0 ; 0x004: blk=0
kfbh.block.obj:                   3 ; 0x008: file=3
```

通常来讲，一个 block 是元数据，而后面的 block 就是实际数据了。继续读取 Active change directory data：

[grid@oracle11 ~]$ **kfed read /dev/asm-data1 aun=5 blkn=1 | more**

```
kfbh.endian:                      1 ; 0x000: 0x01
kfbh.hard:                      130 ; 0x001: 0x82
kfbh.type:                        8 ; 0x002: KFBTYP_CHNGDIR
kfbh.datfmt:                      1 ; 0x003: 0x01
kfbh.block.blk:                   1 ; 0x004: blk=1
```

```
kfbh.block.obj:                       3 ; 0x008: file=3
```

最后简单总结一下：

1. Active change dictectory，也就是 asm 元数据 file 3，一共占据 42 个 AU 大小，简称 ACD. 每个 asm 实例对应一份 ACD 信息，换句话讲，你是双节点 asm rac，那么就有 84M 的 ACD 数据,以此类推.(事实上不管你 AU 是多大 ACD 的信息都是固定的大小)

2. asm 中 ACD 就类似数据库实例中的 redo，记录 asm 元数据操作记录,以便于 asm crash 后进行 instance recover.

3. ACD 信息所在 AU，第一个 block 是其元数据，后面的 block 是 data 信息.

4. ACD data 的数据，跟 redo 的结构有点类似，里面记录的也是 thread，sequence,len,opcode 等信息.

5. ACD 数据的变化是通过 asm lgwr 进程来完成的，该进程跟数据库实例的 lgwr 进程类似，也存在一个 3s check 的机制。

2.2.4. Continuing Operations Directory(COD)

Some long-running ASM operations, like the rebalance, drop disk, create/delete/resize file, cannot be described by a single record in the ASM active change directory. Those operations are tracked via the ASM continuing operations directory (COD) - the ASM file number 4. There is one COD per disk group.

If the process performing the long-running operation dies before completing it, a recovery process will look at the entry and either complete or rollback the operation. There are two types of continuing operations - background and rollback.

Continuing Operations Directory (COD)，就其作用而来可能就类似 undo 了。

在 oracle asm 实例中，有一些长时间运行的操作，比如当你 add/drop disk 时,add/delete/resize datafile 时，可能运行时间相对较长，这时 asm 的一些元数据信息就无法仅仅通过 active change directory 来记录，还需要 COD 来进行记录。

有一点大家需要记住的是，你的 asm 实例中，有多个 COD，那么你的 asm 实例中就有多少个 ASM disk group，其关系是 1:1 的。大家可以看到，我这里的 asm 实例中，其中 diskgroup 1 中包含 2 个 disk，所以通过如下 sql 查询的结果显示有 2 个 COD 条目:

SQL>

```
SELECT x.xnum_kffxp "Extent", x.au_kffxp "AU", x.disk_kffxp "Disk #", d.name "Disk name"
  FROM x$kffxp x, v$asm_disk_stat d
 WHERE x.group_kffxp = d.group_number
   AND x.disk_kffxp = d.disk_number
```

```
  AND x.group_kffxp = 1
  AND x.number_kffxp = 4
ORDER BY 1, 2;
```

```
   Extent         AU     Disk # Disk name
---------- ---------- ---------- ------------------------------------------------------------
        0         90          0 DATA_DG_0000
        1       2528          0 DATA_DG_0000
        2         92          0 DATA_DG_0000
        3       2529          0 DATA_DG_0000
        4         94          0 DATA_DG_0000
        5       2530          0 DATA_DG_0000
        6         96          0 DATA_DG_0000
        7       2531          0 DATA_DG_0000
```

如果 asm 的这些操作失败了，那么 asm 实例的 recover processes 会去读取 COD 中的信息的，就好比数据库实例中的 smon 进程在进行实例恢复时，当进入到 rollback 的过程时，会去读取 undo 一样。 针对 asm 实例，完成这个操作的进程也是 asm_gmon 来实现的。

[grid@oracle11 ~]$ kfed read /dev/asm-data1 aun=2 blkn=4 | grep au | more

```
kfbh.endian:                          1 ; 0x000: 0x01
kfbh.hard:                          130 ; 0x001: 0x82
kfbh.type:                            4 ; 0x002: KFBTYP_FILEDIR
kfbh.datfmt:                          1 ; 0x003: 0x01
kfbh.block.blk:                       4 ; 0x004: blk=4
kfbh.block.obj:                       1 ; 0x008: file=1
......
kfffde[0].xptr.au:                   90 ; 0x4a0: 0x0000005a
kfffde[0].xptr.disk:                  0 ; 0x4a4: 0x0000
kfffde[0].xptr.flags:                 0 ; 0x4a6: L=0 E=0 D=0 S=0
kfffde[0].xptr.chk:                 112 ; 0x4a7: 0x70
kfffde[1].xptr.au:                 2528 ; 0x4a8: 0x000009e0
kfffde[1].xptr.disk:                  0 ; 0x4ac: 0x0000
kfffde[1].xptr.flags:                 0 ; 0x4ae: L=0 E=0 D=0 S=0
kfffde[1].xptr.chk:                 195 ; 0x4af: 0xc3
kfffde[2].xptr.au:                   92 ; 0x4b0: 0x0000005c
kfffde[2].xptr.disk:                  0 ; 0x4b4: 0x0000
kfffde[2].xptr.flags:                 0 ; 0x4b6: L=0 E=0 D=0 S=0
kfffde[2].xptr.chk:                 118 ; 0x4b7: 0x76
```

[grid@oracle11 ~]$ kfed read /dev/asm-data1 aun=90 blkn=0 | more

```
kfbh.endian:                          1 ; 0x000: 0x01
kfbh.hard:                          130 ; 0x001: 0x82
kfbh.type:                            9 ; 0x002: KFBTYP_COD_BGO
kfbh.datfmt:                          1 ; 0x003: 0x01
kfbh.block.blk:                       0 ; 0x004: blk=0
kfbh.block.obj:                       4 ; 0x008: file=4
```

BGO 即为 background operations 的简写。

[grid@oracle11 ~]$ kfed read /dev/asm-data1 aun=90 blkn=1 | more

```
kfbh.endian:                          1 ; 0x000: 0x01
kfbh.hard:                          130 ; 0x001: 0x82
kfbh.type:                           15 ; 0x002: KFBTYP_COD_RBO
kfbh.datfmt:                          2 ; 0x003: 0x02
kfbh.block.blk:                       1 ; 0x004: blk=1
kfbh.block.obj:                       4 ; 0x008: file=4
......
kfrcrb[0].opcode:                     0 ; 0x000: 0x0000
kfrcrb[1].opcode:                     0 ; 0x002: 0x0000
```

kfbh.type 表示操作类型,15 即为 KFBTYP_COD_RBO,RBO 即为 rollback operation 的简写。

```
kfrcrb[0].opcode      表示具体的操作类型，该 opcode 有很多种属性值,如下：
 1 - Create a file
 2 - Delete a file                9  - Disk Resync
 3 - Resize a file                10 - Disk Repair Time
 4 - Drop alias entry             11 - Volume create
 5 - Rename alias entry           12 - Volume delete
 6 - Rebalance space COD          13 - Attribute directory creation
 7 - Drop disks force             14 - Set zone attributes
 8 - Attribute drop               15 - User drop
```

2.2.5. **Template Directory**

The Template Directory - ASM file number 5 - contains information about all file templates for the disk group.

There are two types of templates - system and user created. The default (system) templates are always available for each file type supported by ASM. User created templates can be added for a custom template specifications.

Each template entry contains the following information:

The template name (for the default templates this corresponds to the file type)

The file redundancy (defaults to the disk group redundancy)

The file striping (default is file-type specific)

The system flag (set for the system templates)

SQL>

```
SELECT NAME "Template Name", redundancy "Redundancy", stripe "Striping", system "System"
  FROM v$asm_template
 WHERE group_number = 1;
```

```
Template Name                  Redundancy   Striping     Sy
------------------------------ ------------ ------------ --
PARAMETERFILE                  UNPROT       COARSE       Y
ASMPARAMETERFILE               UNPROT       COARSE       Y
DUMPSET                        UNPROT       COARSE       Y
CONTROLFILE                    UNPROT       FINE         Y
FLASHFILE                      UNPROT       COARSE       Y
ARCHIVELOG                     UNPROT       COARSE       Y
ONLINELOG                      UNPROT       COARSE       Y
DATAFILE                       UNPROT       COARSE       Y
TEMPFILE                       UNPROT       COARSE       Y
BACKUPSET                      UNPROT       COARSE       Y
XTRANSPORT BACKUPSET           UNPROT       COARSE       Y
AUTOBACKUP                     UNPROT       COARSE       Y
XTRANSPORT                     UNPROT       COARSE       Y
CHANGETRACKING                 UNPROT       COARSE       Y
FLASHBACK                      UNPROT       COARSE       Y
DATAGUARDCONFIG                UNPROT       COARSE       Y
OCRFILE                        UNPROT       COARSE       Y
```

17 rows selected.

[grid@oracle11 ~]$ kfed read /dev/asm-data1 aun=2 blkn=5 | more

```
kfbh.endian:                          1 ; 0x000: 0x01
kfbh.hard:                          130 ; 0x001: 0x82
kfbh.type:                            4 ; 0x002: KFBTYP_FILEDIR
kfbh.datfmt:                          1 ; 0x003: 0x01
kfbh.block.blk:                       5 ; 0x004: blk=5
......
```

```
kfffde[0].xptr.au:                51 ; 0x4a0: 0x00000033
kfffde[0].xptr.disk:               0 ; 0x4a4: 0x0000
kfffde[0].xptr.flags:              0 ; 0x4a6: L=0 E=0 D=0 S=0
kfffde[0].xptr.chk:               25 ; 0x4a7: 0x19
```

[grid@oracle11 ~]$ kfed read /dev/asm-data1 aun=51 blkn=1 | more

```
kfbh.endian:                       1 ; 0x000: 0x01
kfbh.hard:                       130 ; 0x001: 0x82
kfbh.type:                        10 ; 0x002: KFBTYP_TMPLTDIR
kfbh.datfmt:                       1 ; 0x003: 0x01
kfbh.block.blk:                    1 ; 0x004: blk=1
kfbh.block.obj:                    5 ; 0x008: file=5
```

其实 template directory 的结构也很简单，主要是如下几部分：

1) kfbh,头部信息，跟前面其他的文章描述的一样，不累述;

2) kffdnd,kffdnd,从上面输出的信息，我们不难猜测，这部分信息其实就是用来定位和描述 block 在目录树中的具体位置的。

跟前面描述 disk directory 的 kffdnd 结构是一样的，所以这里也不多说。

3) kftmde,这部分结构主要是包括 template 模板的详细信息，如条带大小，宽度等信息.

2.2.6. ASM Alias

The alias directory - ASM file number 6 - provides a hierarchical naming system for all the files in a disk group.

A system file name is created for every file and it is based on the file type, database instance and type-specific information such as tablespace name. User alias may also be created if a full path name was given when the file was created.

Alias Directory entries include the following fields:

Alias or directory name

Alias incarnation number

File number

File incarnation number

Parent directory

System flag

The ASM alias information is externalised via V$ASM_ALIAS view.

SQL>

```
SELECT full_path, dir, sys
  FROM (SELECT CONCAT('+' || gname, sys_connect_by_path(aname, '/')) full_path, dir, sys
          FROM (SELECT g.name          gname,
                       a.parent_index   pindex,
                       a.name          aname,
                       a.reference_index rindex,
                       a.alias_directory dir,
                       a.system_created  sys
                  FROM v$asm_alias a, v$asm_diskgroup g
                 WHERE a.group_number = g.group_number)
         START WITH (MOD(pindex, power(2, 24))) = 0
        CONNECT BY PRIOR rindex = pindex
         ORDER BY dir DESC, full_path ASC);
```

```
FULL_PATH                                                                      DI SY
------------------------------------------------------------------------------ -- --
+DATA_DG/ACCTPHY/ONLINELOG/group_6.380.893601943                                N  Y
+DATA_DG/ACCTPHY/ONLINELOG/group_7.381.893601947                                N  Y
+DATA_DG/ACCTPHY/ONLINELOG/group_8.382.893601949                                N  Y
+DATA_DG/ACCTPHY/ONLINELOG/group_9.383.893601953                                N  Y
+GRID_ACCT/acct/spfileacct.ora                                                 N  N
+GRID_ACCT/rac11g-cluster/ASMPARAMETERFILE/REGISTRY.253.872770027              N  Y
+GRID_ACCT/rac11g-cluster/OCRFILE/REGISTRY.255.872769411                       N  Y
```

[grid@rac11g1 ~]$ kfed read /dev/asm-data1 aun=2 blkn=6 | more

```
kfbh.endian:                          1 ; 0x000: 0x01
kfbh.hard:                          130 ; 0x001: 0x82
kfbh.type:                            4 ; 0x002: KFBTYP_FILEDIR
kfbh.datfmt:                          1 ; 0x003: 0x01
kfbh.block.blk:                       6 ; 0x004: blk=6
kfbh.block.obj:                       1 ; 0x008: file=1
kfffde[0].xptr.au:                 2532 ; 0x4a0: 0x000009e4
kfffde[0].xptr.disk:                  0 ; 0x4a4: 0x0000
kfffde[0].xptr.flags:                 0 ; 0x4a6: L=0 E=0 D=0 S=0
kfffde[0].xptr.chk:                 199 ; 0x4a7: 0xc7
```

[grid@rac11g1 ~]$ kfed read /dev/asm-data1 aun=2532 blkn=0 | more

```
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
```

2.2.7. ADVM Volume Directory

ASM metadata file number 7 - volume directory - keeps track of files associated with ASM Dynamic Volume Manager (ADVM) volumes.

An ADVM volume device is constructed from an ASM dynamic volume. One or more ADVM volume devices may be configured within each disk group. ASM Cluster File System (ACFS) is layered on ASM through the ADVM interface. ASM dynamic volume manager is another client of ASM - the same way the database is. When a volume is opened, the corresponding ASM file is opened and ASM extents are sent to the ADVM driver.

There are two file types associated with ADVM volumes

- ASMVOL – The volume file which is the container for the volume storage
- ASMVDRL – The file that contains the volume's Dirty Region Logging (DRL) information. This file is required for re-silvering mirrors

```
SQL> create diskgroup ACFS
disk 'ORCL:ASMDISK5', 'ORCL:ASMDISK6'
attribute 'COMPATIBLE.ASM' = '11.2', 'COMPATIBLE.ADVM' = '11.2';

$ asmcmd volcreate -G ACFS -s 2G ACFS_VOL1
$ asmcmd volcreate -G ACFS -s 2G ACFS_VOL2

$ asmcmd volinfo -a

SELECT x.xnum_kffxp "Extent", x.au_kffxp "AU", x.disk_kffxp "Disk #", d.name "Disk
name"
   FROM x$kffxp x, v$asm_disk_stat d
 WHERE x.group_kffxp = d.group_number
    AND x.disk_kffxp = d.disk_number
    AND x.group_kffxp = 2
    AND x.number_kffxp = 7
 ORDER BY 1, 2;
```

```
     Extent          AU        Disk # Disk name
---------- ---------- ---------- ------------------------------
         0         53          1 ASMDISK6
         0         53          0 ASMDISK5
```

```
$ kfed read /dev/oracleasm/disks/ASMDISK5 aun=53 | more
kfbh.endian:                          1 ; 0x000: 0x01
kfbh.hard:                          130 ; 0x001: 0x82
kfbh.type:                           22 ; 0x002: KFBTYP_VOLUMEDIR
```

```
SELECT x.xnum_kffxp "Extent", x.au_kffxp "AU", x.disk_kffxp "Disk #", d.name "Disk
name"
  FROM x$kffxp x, v$asm_disk_stat d
 WHERE x.group_kffxp = d.group_number
   AND x.disk_kffxp = d.disk_number
   AND x.group_kffxp = 2
   AND x.number_kffxp = 7
 ORDER BY 1, 2;
```

```
SELECT file_number "File #", bytes / 1024 / 1024 "Size (MB)", TYPE FROM v$asm_file
WHERE group_number = 2;
```

```
# /sbin/mkfs -t acfs /dev/asm/acfs_vol1-159
# mkdir /acfs1
# mount -t acfs /dev/asm/acfs_vol1-159 /acfs1
# mount
$ asmcmd volinfo -G ACFS ACFS_VOL1
$ kfed read /dev/oracleasm/disks/ASMDISK6 aun=53 blkn=1 | grep mountpath
```

2.2.8. Disk Used Space Directory

The disk Used Space Directory (USD) – ASM file number 8 - maintains the number of allocation units (AU) used per zone, per disk in a disk group. The USD is split into a set of Used Space Entries (USE). Each USE will maintain a counter for the number of used AUs per disk, per zone. A disk zone can be either HOT or COLD.

This structure is version 11.2 specific and is relevant to the Intelligent Data Placement feature. The USD will be present in a newly created disk group in version 11.2 or when the ASM compatibility is advanced to 11.2.

```
SELECT d.group_number "Group#", x.disk_kffxp "Disk#", x.xnum_kffxp "Extent",
x.au_kffxp "AU", d.name "Disk name"
  FROM x$kffxp x, v$asm_disk_stat d
 WHERE x.group_kffxp = d.group_number
   AND x.disk_kffxp = d.disk_number
   AND x.number_kffxp = 8
 ORDER BY 1, 2;
```

```
   Group#      Disk#     Extent         AU Disk name
--------- ---------- ---------- ---------- ------------------
        1          0          0        100 DATA_DG_0000
        2          0          0         50 GRID_ACCT_0000
```

[grid@rac11g1 ~]$ kfed read /dev/asm-data1 aun=100 blkn=0 | more

```
kfbh.endian:                          1 ; 0x000: 0x01
kfbh.hard:                          130 ; 0x001: 0x82
kfbh.type:                           26 ; 0x002: KFBTYP_USEDSPC
kfbh.datfmt:                          1 ; 0x003: 0x01
kfbh.block.blk:                       0 ; 0x004: blk=0
kfbh.block.obj:                       8 ; 0x008: file=8
```

2.2.9. ASM Attributes Directory

SQL>

```
SELECT g.name "Group", a.name "Attribute", a.value "Value"
  FROM v$asm_diskgroup g, v$asm_attribute a
 WHERE g.group_number = a.group_number
   AND a.name NOT LIKE 'template%';
```

```
Group      Attribute                      Value
---------- ------------------------------ ------------------------------
DATA_DG    disk_repair_time               3.6h
DATA_DG    au_size                        1048576
DATA_DG    access_control.umask           066
DATA_DG    access_control.enabled         FALSE
DATA_DG    cell.smart_scan_capable        FALSE
DATA_DG    compatible.rdbms               10.1.0.0.0
DATA_DG    compatible.asm                 11.2.0.0.0
```

```
DATA_DG    sector_size                    512
GRID_ACCT  disk_repair_time               3.6h
GRID_ACCT  access_control.enabled         FALSE
GRID_ACCT  cell.smart_scan_capable        FALSE
GRID_ACCT  compatible.rdbms               10.1.0.0.0
GRID_ACCT  compatible.asm                 11.2.0.0.0
GRID_ACCT  sector_size                    512
GRID_ACCT  au_size                        1048576
GRID_ACCT  access_control.umask           066
```

[grid@rac11g1 ~]$ asmcmd lsattr -lm disk_repair_time

```
Group_Name  Name              Value  RO  Sys
DATA_DG     disk_repair_time  3.6h   N   Y
GRID_ACCT   disk_repair_time  3.6h   N   Y
```

SQL>

```
SELECT x.disk_kffxp "Disk#", x.xnum_kffxp "Extent", x.au_kffxp "AU", d.name "Disk name"
  FROM x$kffxp x, v$asm_disk_stat d
 WHERE x.group_kffxp = d.group_number
   AND x.disk_kffxp = d.disk_number
   AND d.group_number = 1
   AND x.number_kffxp = 9
 ORDER BY 1, 2;
```

```
    Disk#     Extent         AU Disk name
---------- ---------- ---------- --------------------

        0          0       2533 DATA_DG_0000
```

[grid@rac11g1 ~]$ kfed read /dev/asm-data1 aun=2 blkn=9 | more

```
kfbh.endian:                          1 ; 0x000: 0x01
kfbh.hard:                          130 ; 0x001: 0x82
kfbh.type:                            4 ; 0x002: KFBTYP_FILEDIR
kfbh.datfmt:                          1 ; 0x003: 0x01
kfbh.block.blk:                       9 ; 0x004: blk=9
kfbh.block.obj:                       1 ; 0x008: file=1
......
kfffde[0].xptr.au:                 2533 ; 0x4a0: 0x000009e5
kfffde[0].xptr.disk:                  0 ; 0x4a4: 0x0000
kfffde[0].xptr.flags:                 0 ; 0x4a6: L=0 E=0 D=0 S=0
kfffde[0].xptr.chk:                 198 ; 0x4a7: 0xc6
```

[grid@rac11g1 ~]$ kfed read /dev/asm-data1 aun=2533 blkn=0 | more

```
kfbh.endian:                          1 ; 0x000: 0x01
kfbh.hard:                          130 ; 0x001: 0x82
kfbh.type:                           23 ; 0x002: KFBTYP_ATTRDIR
kfbh.datfmt:                          1 ; 0x003: 0x01
kfbh.block.blk:                       0 ; 0x004: blk=0
kfbh.block.obj:                       9 ; 0x008: file=9
```

[grid@rac11g1 ~]$ kfed read /dev/asm-data1 aun=2533 blkn=0 | egrep "name|value"

```
kfede[0].name:          disk_repair_time ; 0x034: length=16
kfede[0].value:                    3.6h ; 0x074: length=4
kfede[1].name:       _rebalance_compact ; 0x1a8: length=18
kfede[1].value:                    TRUE ; 0x1e8: length=4
kfede[2].name:            _extent_sizes ; 0x31c: length=13
kfede[2].value:                 1 4 16 ; 0x35c: length=6
```

http://www.killdb.com/2013/01/15/oracle-asm-%E5%89%96%E6%9E%90%E7%B3%BB%E5%88%975-alias-directory.html#comment-1487

3. 管理 ASM 实例

3.1. ASM Instance

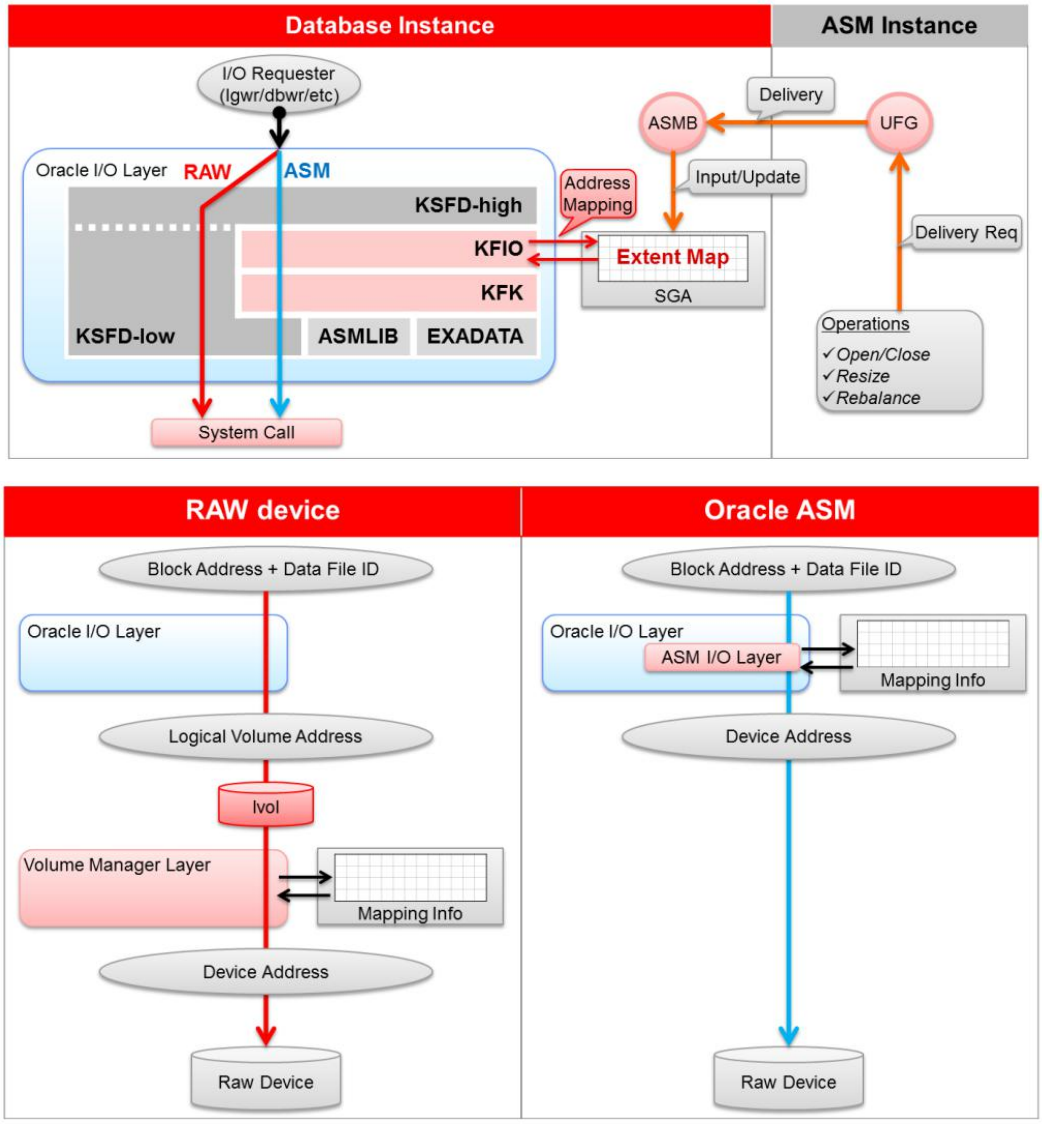
ASM instance 的主要任务之一就是管理 ASM metadata 元数据； ASM Instance 类似于 ORACLE RDBMS INSTANCE 有其 SGA 和大多数主要后台进程。在 10.2 中使用与 RDBMS 一样的 2 进制软件，到 11.2 中分家。但 ASM instance 加载的不是数据库，而是 Disk Group; 并负责告诉 RDBMS database instance 必要的 ASM 文件信息。 ASM 实例和 DB 实例均需要访问 ASM DISK。 ASM 实例管理 metadata 元数据，这些元数据信息足以描述 ASM 中的 FILE 的信息。 数据库实例仍旧直接访问文件，虽然它需要通过 ASM 实例来获得例如 文件 Extent Map 盘区图等信息，但 I/O 仍由其自行完成，而不是说使用了 ASM 之后 DB 的文件 I/O 需要通过 ASM 来实现； 其仅仅是与 ASM instance 交互来获得文件位置、状态等信息。

有一些操作需要 ASM 实例介入处理，例如 DB 实例需要创建一个数据文件，则数据库服务进程直接连接到 ASM 实例来实现一些操作； 每一个数据库维护一个连接池到其 ASM 实例，来避免文件操作导致的反复连接。

ASM metadata 通过一个独立的 ASM 实例来管理以便减少其被损坏的可能。ASM instance 很相似于 db instance，虽然它一般只使用 ORACLE KERNEL 内核的一小部分代码，则其遇到 bug 或导致 buffer cache 讹误或者写出讹误到磁盘的概率由此比 DB 实例要小。 数据库实例自己从来不更新 ASM metadata。ASM metadata 中每一个指针一般都有 check byte 以便验证。

和 DB RAC 一样，ASM instance 自己可以被集群化，一样是使用 ORACLE Distributed Lock Manager(DLM)分布式锁管理器架构。在一个集群中每一个节点上可以有一个 ASM instance。如果一个节点上有多个数据库、多个实例，则他们共享使用一个 ASM instance 。

如果一个节点上的 ASM instance 失败了，则所有使用该 ASM instance 均会失败。但其他节点上的 ASM 和数据库实例将做 recover 并继续操作。



Oracle ASM is installed in the Oracle Grid Infrastructure home separate from the Oracle Database home. Only one Oracle ASM instance is supported on a server. When managing an Oracle ASM instance, the administration activity must be performed in the Oracle Grid Infrastructure home.

For compatibility between Oracle Clusterware and Oracle ASM, the Oracle Clusterware release must be greater than or equal to the Oracle ASM release.

The V$ASM_CLIENT view contains the SOFTWARE_VERSION and COMPATIBLE_VERSION columns with information about the software version number and instance compatibility level.

SQL> SELECT GROUP_NUMBER, DB_NAME, SOFTWARE_VERSION, COMPATIBLE_VERSION FROM v$asm_client;

```
GROUP_NUMBER DB_NAME  SOFTWARE_VERSION     COMPATIBLE_VERSION
------------ -------- -------------------- --------------------
           1 sky      11.2.0.4.0           11.2.0.0.0
```

[grid@rac11g1 ~]$ asmcmd lsct

```
DB_Name  Status     Software_Version  Compatible_version  Instance_Name  Disk_Group
+ASM     CONNECTED        11.2.0.4.0          11.2.0.4.0  +ASM1          GRID_ACCT
+ASM     CONNECTED        11.2.0.4.0          11.2.0.4.0  +ASM1          DATA_DG
acct     CONNECTED        11.2.0.4.0          11.2.0.4.0  acct1          DATA_DG
```

3.1.1. ab_<ASM SID>.dat

ASM 实例启动后生成，用于 RDBMS 确定连接 ASM 实例的信息。文件被删除后，RDBMS 无法正常连接数据库，该文件不包含连接的认证信息。

重启 ASM 实例，可以自动恢复。

3.1.2. hc_<SID>.dat

实例健康检查监控文件。

3.2. ASM 初始化参数管理

3.2.1. ASM 初始化参数

When an Oracle ASM instance searches for an initialization parameter file, the search order is:

- The location of the initialization parameter file specified in the Grid Plug and Play (GPnP) profile
- If the location has not been set in the GPnP profile, then the search order changes to:
  - SPFILE in the Oracle ASM instance home

    For example, the SPFILE for Oracle ASM has the following default path in the Oracle Grid Infrastructure home in a Linux environment:

    $ORACLE_HOME/dbs/spfile+ASM.ora
  - PFILE in the Oracle ASM instance home

[grid@rac11g1 ~]$ gpnptool get

3.2.2. 初始化参数备份、移动

You can back up, copy, or move an Oracle ASM SPFILE with the ASMCMD spbackup, spcopy or spmove commands.

You can also use the SQL CREATE SPFILE to create an Oracle ASM SPFILE when connected to the Oracle ASM instance.

If the COMPATIBLE.ASM disk group attribute is set to 11.2 or greater for a disk group, you can create, copy, or move an Oracle ASM SPFILE into the disk group.

```
SQL> CREATE SPFILE = '+DATA/asmspfile.ora'
        FROM PFILE = '$ORACLE_HOME/dbs/asmpfile.ora';
```

- spget

Retrieves the location of the Oracle ASM SPFILE from the Grid Plug and Play (GPnP) profile.

ASMCMD> spget

+OCRDG/oeldb-cluster/asmparameterfile/registry.253.831550619

ASMCMD> spcopy +GRID_ACCT/acct/spfileacct.ora /home/grid/spfile.sp

3.2.3. ASM 内存管理

Automatic memory management automatically manages the memory-related parameters for both Oracle ASM and database instances with the MEMORY_TARGET parameter.

The default value used for MEMORY_TARGET is acceptable for most environments. This is the only parameter that you must set for complete Oracle ASM memory management.

SQL> select * from v$sgainfo;

```
NAME                                  BYTES RES
------------------------------ ---------- ---

Fixed SGA Size                      2227664 No
ASM Cache Size                     25165824 No
Shared Pool Size                  167772160 Yes
```

```
Large Pool Size                  12582912 Yes
Granule Size                      4194304 No
Maximum SGA Size                283930624 No
Startup overhead in Shared Pool  76946896 No
Free SGA Memory Available        75497472
```

3.2.4. ASM 建议参数

- ASM_DISKGROUPS

The ASM_DISKGROUPS initialization parameter specifies a list of the names of disk groups that an Oracle ASM instance mounts at startup.

The ASM_DISKGROUPS parameter is dynamic.

Oracle ASM automatically adds a disk group to this parameter when the disk group is successfully created or mounted.

Oracle ASM also automatically removes a disk group from this parameter when the disk group is dropped or dismounted.

在增加和删除磁盘组时，自动更新该参数。

SQL> ALTER SYSTEM SET ASM_DISKGROUPS = DATA, FRA;

Note:

Issuing the ALTER DISKGROUP...ALL MOUNT or ALTER DISKGROUP...ALL DISMOUNT commands does not affect the value of ASM_DISKGROUPS.

- ASM_DISKSTRING

The ASM_DISKSTRING initialization parameter specifies a comma-delimited (逗号分割) list of strings that limits the set of disks that an Oracle ASM instance discovers.

The same disk cannot be discovered twice. 无法被多次发现，因此需要注意多路径设备的管理。

The discovery string format depends on the Oracle ASM library and the operating system that are in use. Pattern matching is supported.

The ? character, when used as the first character of a path, expands to the Oracle home directory. ？ 代表 ORACLE_HOME 环境变量。

Oracle ASM cannot use a disk unless all of the Oracle ASM instances in the cluster can discover the disk through one of their own discovery strings.

➢ ASM_POWER_LIMIT

The ASM_POWER_LIMIT initialization parameter specifies the default power for disk rebalancing in a disk group. The range of values is 0 to 1024. The default value is 1. A value of 0 disables rebalancing. Higher numeric values enable the rebalancing operation to complete more quickly, but might result in higher I/O overhead and more rebalancing processes.

- For disk groups that have the disk group ASM compatibility set to 11.2.0.2 or greater (for example, COMPATIBLE.ASM = 11.2.0.2), the operational range of values is 0 to 1024 for the rebalance power.
- For disk groups that have the disk group ASM compatibility set to less than 11.2.0.2, the operational range of values is 0 to 11 inclusive. If the value for ASM_POWER_LIMIT is larger than 11, a value of 11 is used for these disk groups.

➢ ASM_PREFERRED_READ_FAILURE_GROUPS

The ASM_PREFERRED_READ_FAILURE_GROUPS initialization parameter value is a comma-delimited list of strings that specifies the **failure groups** that should be preferentially read by the given instance.

For example:

```
diskgroup_name1.failure_group_name1, ...
```

➢ DB_CACHE_SIZE

The setting for the DB_CACHE_SIZE parameter determines the size of the buffer cache. This buffer cache is used to store metadata blocks.

- DIAGNOSTIC_DEST

The DIAGNOSTIC_DEST initialization parameter specifies the directory where diagnostics for an instance are located. The default value for an Oracle ASM instance is the **$ORACLE_BASE** directory for the Oracle Grid Infrastructure installation.

- INSTANCE_TYPE

The INSTANCE_TYPE initialization parameter is optional for an Oracle ASM instance in an Oracle Grid Infrastructure home.

SQL> show parameter instance_type;

```
NAME                                 TYPE        VALUE
------------------------------------ ----------- ------------------------------
instance_type                        string      asm
```

- PROCESSES

The PROCESSES initialization parameter affects Oracle ASM, but the default value is usually suitable. However, if multiple database instances are connected to an Oracle ASM instance, you can use the following formula:

```
PROCESSES = 50 + 50*n
```

where n is the number database instances connecting to the Oracle ASM instance.

- SHARED_POOL

```
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';
```

1. For diskgroups using external redundancy = (every 100GB of file space needs 1MB of extra shared pool) + 2MB

2. For diskgroups using normal redundancy: (every 50GB of file space needs 1MB of extra shared pool) + 4MB

3. For diskgroups using high redundancy: (every 33GB of file space needs 1MB of extra shared pool) + 6MB

3.2.5. 手工调整 ASM 参数

The following are configuration guidelines for SGA sizing on the database instance:

- PROCESSES initialization parameter—Add 16 to the current value
- LARGE_POOL_SIZE initialization parameter—Add an additional 600K to the current value
- SHARED_POOL_SIZE initialization parameter—Aggregate the values from the following queries to obtain the current database storage size that is either on Oracle ASM or stored in Oracle ASM.

```
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';
```

  - 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

3.3. 管理 ASM 实例

3.3.1. Oracle Restart

When you install the Oracle Grid Infrastructure for a standalone server, it includes both Oracle ASM and Oracle Restart.

3.3.2. ASM 实例启动

在 ASM 文件可以通过 ASM 实例来访问之前，ASM 实例必须先启动。 在 11.2 下不管是 RAC 还是 StandAlone 环境下 ASM 实例都会随系统 BOOT 自动启动。 启动一个 ASM 实例和启动一个数据库实例类似。 SGA 和一组后台进程在启动过程中被创建出来。初始化参数 instance_type 决定了是 ASM 实例还是数据库实例。除非 STARTUP 时使用了 NOMOUNT 选项，否则默认 STARTUP 会执行 ALTER DISKGROUP ALL MOUNT。

ASM 实例启动过程中将加入到 CSS 中的+ASM 成员组中。这将允许本实例与其他 +ASM 实例共享锁。数据库实例不会加入到这个成员组中，因为数据库实例的实例名不能以"+"开头。

- To connect to a local Oracle ASM instance with SQL*Plus, set the ORACLE_SID environment variable to the Oracle ASM system identifier (SID).

The default Oracle ASM SID for a single-instance database is +ASM, and the default SID for Oracle ASM for an Oracle RAC node is +ASMnode_number where node_number is the number of the node. The ORACLE_HOME environment variable must be set to the Grid Infrastructure home where Oracle ASM was installed.

Note:

Oracle recommends that you do not change the default Oracle ASM SID name.

- The initialization parameter file must contain the following entry:

INSTANCE_TYPE = ASM

- When you run the STARTUP command, rather than trying to mount and open a database, this command attempts to mount Oracle ASM disk groups.

SPFILE='+DATA/asm/asmparameterfile/asmspfile.ora'

SQL> STARTUP PFILE=/u01/oracle/dbs/spfileasm_init.ora

ASMCMD> spset +DATA/asm/asmparameterfile/asmspfile.ora

3.3.3. Mounting Disk Groups

At startup, the Oracle ASM instance attempts to mount the following disk groups:

- Disk groups specified in the ASM_DISKGROUPS initialization parameter
- Disk group used by Cluster Synchronization Services (CSS) for voting files
- Disk groups used by Oracle Clusterware for Oracle Cluster Registry (OCR)

- Disk group used by the Oracle ASM instance to store the ASM server parameter file (SPFILE)

3.3.4. ASM 实例权限

An Oracle ASM instance does not have a data dictionary, so the only way to connect to an Oracle ASM instance is by using one of three system privileges, SYSASM, SYSDBA, or SYSOPER.

Operating system authentication using membership in the group or groups designated as OSDBA, OSOPER, and OSASM is valid on all Oracle platforms.

3.4. 数据库连接 ASM 实例

Rather than requiring a static configuration file to locate the ASM instance, the RDBMS contacts the Cluster Synchronization Services (CSS) daemon where the ASM instance has registered.

当数据库实例尝试打开或者创建名字以“+”开头的文件时， 它会通过 CSS(Cluster Synchronization Services)来查看 disk group 和 mount 该 DG 的 ASM 实例的信息。 如果数据库实例之前访问过其他 Disk Group 里的文件，则将使用同一个 ASM 实例。 如果这是第一次访问 ASM 上的文件，数据库实例就需要连接到 ASM 实例。

下面为数据库实例准备访问 ASM 上文件的步骤：

后台进程 ASMB 启动并 connect 连接到 ASM 实例。 数据库实例所打开的任意文件的 extent map 盘区图被发送给 ASMB 后台进程。 其有义务去维护 extent map。若发生任何 extent 移位，则 ASM 实例将更新发送给数据库实例的 ASMB 进程。I/O 统计信息定期由 ASMB 进程反馈给 ASM 实例。

RBAL 后台进程启动，其对 disk group 下的所有磁盘做全局打开操作，其类似于 DBWR 进程全局打开数据文件。此全局打开允许数据库实例访问 diskgroup 中的任意文件。 若还有其他 disk group 需要被访问，则 RBAL 也将打开对应 diskgroup 下的所有磁盘。 对 add 加入或者 drop 的磁盘，RBAL 也会打开和关闭它们。 关于磁盘的讯息先是发送给 ASMB，之后 ASMB 转发给 RBAL。

会创建一个连接池，一组 slave 进程将建立到 ASM 实例的连接。数据库进程若需要发送信息给 ASM 实例，则需要使用这些 slave 进程。 举个例子来说，打开一个文件，将通过

slave 给 ASM 发送一个 OPEN 的请求。 但对于长时间运行的操作例如创建文件，则不使用 slave。

3.5. 核心进程

[oracle@rac11g1 ~]$ ps -ef | grep asm_

```
grid      4002     1  0 09:31 ?        00:00:00 asm_pmon_+ASM1
grid      4004     1  0 09:31 ?        00:00:00 asm_psp0_+ASM1
grid      4006     1  0 09:31 ?        00:00:20 asm_vktm_+ASM1
grid      4010     1  0 09:31 ?        00:00:00 asm_gen0_+ASM1
grid      4012     1  0 09:31 ?        00:00:00 asm_diag_+ASM1
grid      4014     1  0 09:31 ?        00:00:00 asm_ping_+ASM1
grid      4016     1  0 09:31 ?        00:00:01 asm_dia0_+ASM1
grid      4018     1  0 09:31 ?        00:00:01 asm_lmon_+ASM1
grid      4020     1  0 09:31 ?        00:00:01 asm_lmd0_+ASM1
grid      4022     1  0 09:31 ?        00:00:03 asm_lms0_+ASM1
grid      4026     1  0 09:31 ?        00:00:00 asm_lmhb_+ASM1
grid      4028     1  0 09:31 ?        00:00:00 asm_mman_+ASM1
grid      4030     1  0 09:31 ?        00:00:00 asm_dbw0_+ASM1
grid      4032     1  0 09:31 ?        00:00:00 asm_lgwr_+ASM1
grid      4034     1  0 09:31 ?        00:00:00 asm_ckpt_+ASM1
grid      4036     1  0 09:31 ?        00:00:00 asm_smon_+ASM1
grid      4038     1  0 09:31 ?        00:00:00 asm_rbal_+ASM1
grid      4040     1  0 09:31 ?        00:00:00 asm_gmon_+ASM1
grid      4042     1  0 09:31 ?        00:00:00 asm_mmon_+ASM1
grid      4044     1  0 09:31 ?        00:00:00 asm_mmnl_+ASM1
grid      4047     1  0 09:31 ?        00:00:00 asm_lck0_+ASM1
grid      4148     1  0 09:31 ?        00:00:00 asm_asmb_+ASM1
```

3.5.1. ASMB

This process contacts CSS using the diskgroup name and acquires the associated ASM connect string.

3.5.2. ARBx

These are the slave processes that do the rebalance activity (where x is a number).

3.5.3. CKPT

The CKPT process manages cross-instance calls (in RAC).

3.5.4. DBWR

This process manages the SGA buffer cache in the ASM instance. DBWR writes out dirty buffers (changed metadata buffers) from the ASM buffer cache to disk.

3.5.5. GMON

This process is responsible for managing the disk-level activities (drop/offline) and advancing diskgroup compatibility.

3.5.6. KATE

The Konductor or ASM Temporary Errands (KATE) process is used to process disks online. This process runs in the ASM instance and is started only when an offlined disk is onlined.

3.5.7. LGWR

The LGWR process maintains the ASM Active Change Directory (ACD) buffers from the ASM instance and flushes ACD change records to disk.

3.5.8. MARK

The Mark Allocation Unit (AU) for Resync Koordinator (MARK) process coordinates the updates to the Staleness Registry when the disks go offline. This process runs in the RDBMS instance and is started only when disks go offline in ASM redundancy diskgroups.

3.5.9. O0nn

This group of slave processes establishes connections to the ASM instance, where nn is a number from 01 to 10. Through this connection pool, RDBMS processes can send messages to the ASM instance.

3.5.10. PING

The PING process measures network latency and has the same functionality in RDBMS instances.

3.5.11. PMON

This manages processes and process death in the ASM instance.

3.5.12. PSP0

This process spawner process is responsible for creating and managing other Oracle processes.

3.5.13. RBAL

This opens all device files as part of discovery and coordinates the rebalance activity.

3.5.14. SMON

This process is the system monitor and also acts as a liaison to the Cluster Synchronization Services (CSS) process (in Oracle Clusterware) for node monitoring.

3.5.15. VKTM

This process is used to maintain the fast timer and has the same functionality in the RDBMS instances.

```
[grid@oeldb1 ~]$ ps -ef | grep -E -i 'rbal|arb|gmon' | grep -v grep
grid      3477     1  0 17:02 ?        00:00:00 asm_rbal_+ASM1
grid      3481     1  0 17:02 ?        00:00:00 asm_gmon_+ASM1
```

oracle 4373 1 0 17:04 ? 00:00:00 ora_rbal_racdb1

ASM 实例包含三种新的后台进程类型。第一种类型负责协调磁盘组的重新平衡活动，称为 RBAL。第二种类型实际上执行数据区移动。可以同时存在大量这样的进程，将称为 ARB0、ARB1 等，依此类推。第三种类型负责特定的磁盘组监视操作，这些操作用于维护磁盘组内部的 ASM 元数据。将磁盘组监视进程称为 GMON。

4. 管理 ASM 磁盘

4.1. Disk Discovery

Disk Discovery 磁盘发现是指从 OS 层面找到那些 ASM 值得访问的磁盘。也用来找到那些需要被 mount 的 diskgroup 名下的磁盘 ASM DISK，以及管理员希望将其加入到 diskgroup 中的 Disk，或管理员会考虑将其加入到 diskgroup 的 Disk。Discovery 使用一个 discovery string(asm_diskstring)作为输入参数，并返回一系列可能的 DISK。注意一个是要指定 asm_diskstring，另一个是要求这些 disk 的权限可以被 oracle/grid 用户使用。精确的 asm_diskstring discovery 语法取决于操作系统平台和 ASMLIB 库。OS 可接受的路径名生成的匹配，一般对于 discovery strings 也是可用的。一般推荐这个路径名下最好只有 ASM Disk，来避免管理上的问题。

ASM 实例会打开和读取由 asm_diskstring 指定的路径名匹配到的每一个文件并读取前 4k 的 block， 这样做的目的是判断 disk header 的状态；如果它发现这是一个 ASM disk header 则会认为这是一个可以 mount 的 diskgroup 的一部分。如果发现这 4k 的 block 其无法识别，则认为该 disk 可以加入到 ASM diskgroup 中(candidate)。

ASM 实例需要通过一个初始化参数来指定这个 discovery strings，实际就是 asm_diskstring； 注意 asm_diskstring 中可以加入多个路径字符串，例如 ‘/dev/raw*’,’ /dev/asm-disk*’ ；同样的磁盘不会被发现 2 次(除非你欺骗 ASM)。 在 RAC cluster 中如果一个磁盘不是在整个 cluster 范围内都可见，那么这个磁盘无法被加入到 RAC 的 ASM DISKGROUP 中。 在实际使用中每一个节点上的磁盘名字可以不一样，但其实际介质要被操作系统识别。

所有被 ASM 实例成功发现的磁盘，都可以在 V$ASM_DISK 视图中查到。 属于磁盘组的磁盘，磁盘头部都有磁盘组信息标识，它的状态为 MEMBER。未被分配给磁盘组的磁盘，状态为 CANDIDATE 或 PROVISIONED (表明该磁盘是由 ASMLIB(Linux 平台)或 ASMTOOL/ASMTOOLG(Windows 平台)提供)。如果磁盘之前属于某个磁盘组，并且被正常的删除，那么它的状态为 FORMER。

SQL> SELECT name, header_status, path FROM V$ASM_DISK;

```
NAME            HEADER_STATU PATH
--------------- ------------ ----------------------------------------------

                CANDIDATE    /dev/asm-diskg
```

```
                CANDIDATE    /dev/asm-diskf
SKY_0001        MEMBER       /dev/asm-diske
SKY_0000        MEMBER       /dev/asm-diskd
SYSTEMDG_0000   MEMBER       /dev/asm-diskc
OCRDG_0000      MEMBER       /dev/asm-diskb
```

对 discovery 一般存在于 2 种场景下； 第一种是使用 asm_diskstring 中指定的所有字符串来找出所有 ASM 实例必要访问的磁盘。 第二种是指定磁盘路径用以 create diskgroup 或者 add disk to diskgroup。

第一种 discovery 也叫做 shallow discovery，只会返回 asm_diskstring 指定下的磁盘。第二种也叫做 deep discovery，是读取每一个磁盘的第一个块。 disk header 在这里用以分类磁盘是否可用，是被 ASM 外的其他东西实用(例如 LVM)，还是已经被其他 diskgroup 实用。discovery 操作并不会 mount diskgroup 或者写任何磁盘头。

The rules for discovering Oracle ASM disks are as follows:

- Oracle ASM can discover up to 10,000 disks.
- Oracle ASM only discovers disk partitions. Oracle ASM does not discover partitions that include the partition table.
- From the perspective of the installation, candidate disks are those that have the CANDIDATE, PROVISIONED, or FORMER header status.
- When adding a disk, the FORCE option must be used if Oracle ASM recognizes that the disk was managed by Oracle.
- MEMBER disks can usually be added to a disk group by specifying the FORCE flag, if the disks are not part of a currently mounted disk group.

可以触发磁盘发现的操作：

- Mount a disk group with ALTER DISKGROUP ... MOUNT
- Online a disk with ALTER DISKGROUP ... ONLINE DISK
- Add a disk to a disk group with CREATE or ALTER DISKGROUP...ADD DISK
- Resize a disk in a disk group with ALTER DISKGROUP...RESIZE DISK
- Query with SELECT ... FROM V$ASM_DISKGROUP or V$ASM_DISK views

4.2. Disk Header

一个 ASM DISK 的最前面 4096 字节为 disk header，对于 ASM 而言是 block 0 (blkn=0)；许多操作系统会保留 LUN 的第一个 block 来存放分区表或其他 OS 信息。 一般不让 ASM

基础到这个 block，因为 ASM 会毫不犹豫地覆盖这个 block。在一些指定的平台上 ORACLE 从代码层跳过这些操作系统块，但实际操作时一般的惯例是只给 ASM 用那些上面没有分区表的 LUN DISK。

对于这一点详细的展开是，例如你在 AIX 操作系统上使用 PV 作为 ASM DISK，则 PV 上不能有 PVID，同时如果一个 PV 已经分给 ASM 用了，但是由于系统管理员的疏忽而给 PV 分配了一个 PVID，则该 PV 头部的 ASM disk header 会被覆盖掉，这将直接导致 disk header 丢失；如果是 External Redundancy 那么这个 diskgroup 就直接 mount 不起来了。所以对那些会影响 ASM disk header 的操作要慎之又慎，同时最好定期备份 disk header。

ASM disk header 描述了该 ASM disk 和 diskgroup 的属性，通过对现有 disk header 的加载，ASM 实例可以知道这个 diskgroup 的整体信息。

[root@oracle11 ~]# kfed read /dev/asm-data1 | grep ^kfdhdb | grep -v '0x00000000$'

```
kfdhdb.driver.provstr:         ORCLDISK ; 0x000: length=8
kfdhdb.compat:               186646528 ; 0x020: 0x0b200000
kfdhdb.dsknum:                       0 ; 0x024: 0x0000
kfdhdb.grptyp:                       1 ; 0x026: KFDGTP_EXTERNAL
kfdhdb.hdrsts:                       3 ; 0x027: KFDHDR_MEMBER
kfdhdb.dskname:            DATADG_0000 ; 0x028: length=11
kfdhdb.grpname:                 DATADG ; 0x048: length=6
kfdhdb.fgname:             DATADG_0000 ; 0x068: length=11
```

下面的信息是在同一个 diskgroup 中的所有 disk 的 header 上均会复制一份的：

- Disk group name and creation timestamp
- Physical sector size of all disks in the disk group
- Allocation unit size
- Metadata block size
- Software version compatibility
- Default redundancy
- Mount timestamp

下面的信息是每一个 asm disk 独有的：

- ASM disk name (not OS path name)
- Disk number within disk group

- Failure group name
- Disk size in allocation units

4.3. Freespace Table

AU=0 的 blkn=1 包含的是 free space table；其中包含了该 AU 中 allocation table 中每一个 block 上大致的可用剩余 FREE SPACE 可用空间信息。通过参考 free space table 可以避免在已经分配完的 allocation table 中查找空间。

[root@oracle11 ~]# **kfed read /dev/asm-data1 aun=0 blkn=1 | more**

```
kfbh.endian:                          1 ; 0x000: 0x01
kfbh.hard:                          130 ; 0x001: 0x82
kfbh.type:                            2 ; 0x002: KFBTYP_FREESPC
kfbh.datfmt:                          2 ; 0x003: 0x02
kfbh.block.blk:                       1 ; 0x004: blk=1
kfbh.block.obj:              2147483648 ; 0x008: disk=0
kfbh.check:                  3843655122 ; 0x00c: 0xe51985d2
kfbh.fcn.base:                    1836 ; 0x010: 0x0000072c
```

4.4. Allocation Table

Aun=0 的后 254 个 metadata block 用以存放 AU 分配信息。每一个 metadata 描述 448 个 AU 的状态， 如果一个 AU 已经分配给一个文件，则 allocation table 记录其 ASM 文件号和 data extent 号。对于还是 FREE 的 AU 则被 link 到 free list 上。

[root@oracle11 ~]# **kfed read /dev/asm-data1 aun=0 blkn=2 | more**

```
kfbh.endian:                          1 ; 0x000: 0x01
kfbh.hard:                          130 ; 0x001: 0x82
kfbh.type:                            3 ; 0x002: KFBTYP_ALLOCTBL
kfbh.datfmt:                          2 ; 0x003: 0x02
kfbh.block.blk:                       2 ; 0x004: blk=2
kfbh.block.obj:              2147483648 ; 0x008: disk=0
kfbh.check:                  2187821472 ; 0x00c: 0x826781a0
```

4.5. Partner and Status Table

一般来说 aun=1 是保留给 Partner and Status Table(PST)的拷贝使用的。 一般 5 个 ASM DISK 将包含一份 PST 拷贝。多数的 PST 内容必须相同且验证有效。否则无法判断哪些 ASM DISK 实际拥有相关数据。

在 PST 中每一条记录对应 Diskgroup 中的一个 ASM DISK。每一条记录会对一个 ASM disk 枚举其 partners 的 ASM DISK。同时会有一个 flag 来表示该 DISK 是否是 ONLINE 可读写的。这些信息对 recovery 是否能做很重要。

PST 表的 Blkn=0 是 PST 的 header，存放了如下的信息：

- Timestamp to indicate PST is valid
- Version number to compare with other PST copies
- List of disks containing PST copies
- Bit map for shadow paging updates

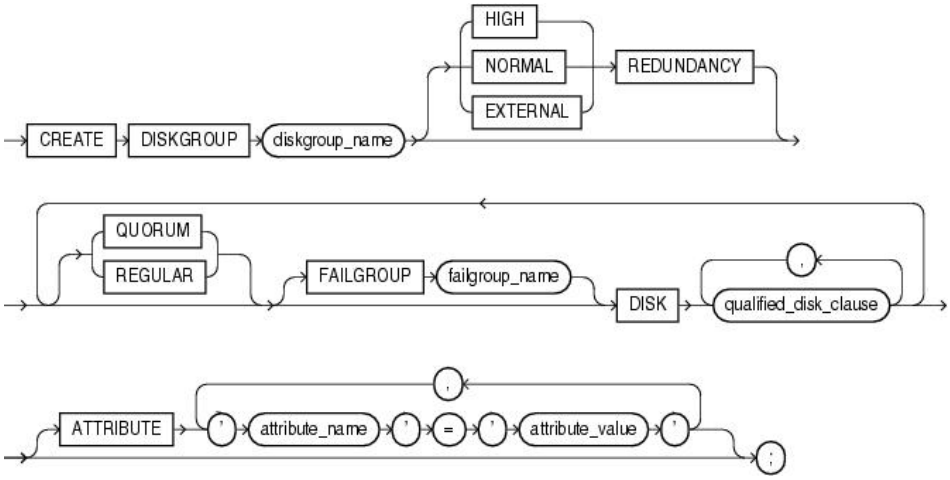
[root@oracle11 ~]# **kfed read /dev/asm-data1 aun=1 blkn=0 | more**

```
kfbh.endian:                          1 ; 0x000: 0x01
kfbh.hard:                          130 ; 0x001: 0x82
kfbh.type:                           17 ; 0x002: KFBTYP_PST_META
kfbh.datfmt:                          2 ; 0x003: 0x02
kfbh.block.blk:                     256 ; 0x004: blk=256
kfbh.block.obj:              2147483648 ; 0x008: disk=0
kfbh.check:                  1372837076 ; 0x00c: 0x51d3d4d4
```

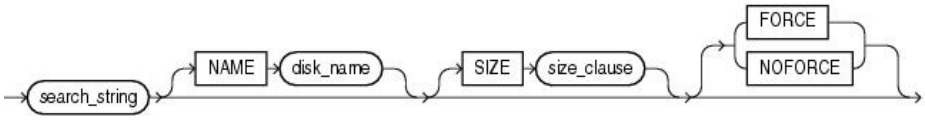
PST 的最后一个块是 heartbeat block，当 diskgroup mount 时其每 3 秒心跳更新一次。

5. 管理 ASM 磁盘组

5.1. 创建磁盘组



qualified_disk_clause::=



创建 diskgroup 需要指定多个磁盘路径，且这些磁盘需要通过如下的检测：

- 它不能是已经 mount 的 diskgroup 的一部分
- 它不能有一个有效的 ASM disk header，除非加了 FORCE 选项
- 它不能有一个有效的 ORACLE DATAFILE HEADER，除非加了 FORCE 选项
- 除非必须，不要用 FORCE 选项
- 它不能存在对 ASM 可见的 2 个不同的路径名字。
- 其必须可以通过 asm_diskstring 来发现

所有的磁盘均会以写入一个 disk header 的形式来验证。该 disk header 中 mount timestamp 为 0 ，由此可知 diskgroup 还没有被 mount 过。之后 free space block 和 allocation table blocks 元数据块将被写入。

You must use FORCE only when adding a disk that was dropped with FORCE. If a disk is dropped with NOFORCE, then you can add it with NOFORCE.

建议 FAILGROUP 中的磁盘来至于不同的控制器。

```
CREATE DISKGROUP data NORMAL REDUNDANCY
  FAILGROUP controller1 DISK
    '/devices/diska1' NAME diska1,
    '/devices/diska2' NAME diska2,
    '/devices/diska3' NAME diska3,
    '/devices/diska4' NAME diska4
  FAILGROUP controller2 DISK
    '/devices/diskb1' NAME diskb1,
    '/devices/diskb2' NAME diskb2,
    '/devices/diskb3' NAME diskb3,
    '/devices/diskb4' NAME diskb4
  ATTRIBUTE 'au_size'='4M',
    'compatible.asm' = '11.2',
    'compatible.rdbms' = '11.2',
    'compatible.advm' = '11.2';
```

```
SQL> CREATE DISKGROUP DATA_NRML NORMAL REDUNDANCY
FAILGROUP FL1GRP1 DISK '/dev/rdsk/c[34]*'
FAILGROUP FLGRP2 DISK '/dev/rdsk/c[56]*';
```

5.1.1. AU_SIZE

Specifies the size of the allocation unit for the disk group.

```
CREATE DISKGROUP fra NORMAL REDUNDANCY
  DISK '/devices/diskc*';
```

Oracle recommends that the allocation unit (AU) size for a disk group be set to 4 megabytes (MB). -- 建议 4M

You can store Oracle Cluster Registry (OCR) and voting files in Oracle ASM disk groups. The voting files and OCR are two important components of Oracle Clusterware.

Voting files manage information about node membership. OCR is a system that manages cluster and Oracle Real Application Clusters (Oracle RAC) database configuration information.

QUORUM disks (or disks in QUORUM failure groups) cannot have client data files, whereas REGULAR disks (or disks in non-quorum failure groups) have no such restriction.

```
CREATE DISKGROUP ocr_data NORMAL REDUNDANCY
    FAILGROUP fg1 DISK '/devices/diskg1'
    FAILGROUP fg2 DISK '/devices/diskg2'
    QUORUM FAILGROUP fg3 DISK '/devices/diskg3'
    ATTRIBUTE 'compatible.asm' = '11.2.0.0.0';
```

5.1.2. SECTOR_SIZE

You can use the optional SECTOR_SIZE disk group attribute with the CREATE DISKGROUP SQL statement to specify disks with the sector size set to the value of SECTOR_SIZE for the disk group. Oracle ASM provides support for 4 KB sector disk drives without negatively affecting performance. The SECTOR_SIZE disk group attribute can be set only during disk group creation.

The values for SECTOR_SIZE can be set to 512, 4096, or 4K if the disks support those values.

```
SELECT NAME, VALUE
  FROM V$ASM_ATTRIBUTE
 WHERE NAME = 'sector_size'
   AND group_number = 1;
```

```
CREATE DISKGROUP data NORMAL REDUNDANCY
FAILGROUP controller1 DISK
'/devices/diska1',
'/devices/diska2',
'/devices/diska3',
```

```
'/devices/diska4'
FAILGROUP controller2 DISK
'/devices/diskb1',
'/devices/diskb2',
'/devices/diskb3',
'/devices/diskb4'
ATTRIBUTE 'compatible.asm' = '11.2', 'compatible.rdbms' = '11.2',
          'sector_size'='4096';
```

Note:

Oracle Automatic Storage Management Cluster File System (Oracle ACFS) does not support 4 KB sector drives.

5.1.3. Intelligent Data Placement

Intelligent Data Placement enables you to specify disk regions on Oracle ASM disks for best performance.

The COMPATIBLE.ASM and COMPATIBLE.RDBMS disk group attributes must be set to 11.2 or higher to use Intelligent Data Placement.

```
ALTER DISKGROUP data ADD TEMPLATE datafile_hot
  ATTRIBUTE (
    HOT
    MIRRORHOT);
```

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

When you modify the disk region settings for a file, this action applies to new extensions of the file, but existing file contents are not affected until a rebalance operation.

该操作对新的扩展直接生效，旧的内容需要进行 rebalance 操作才能生效。

5.2. OCR 磁盘组

Voting files manage information about node membership. OCR is a system that manages cluster and Oracle Real Application Clusters (Oracle RAC) database configuration information. A quorum failure group is a special type of failure group and disks in these failure groups do not contain user data. A quorum failure group is not considered when determining redundancy requirements in respect to storing user data.

The QUORUM and REGULAR keywords provide an additional qualifier for failure group or disk specifications when creating or altering a disk group. QUORUM disks (or disks in QUORUM failure groups) cannot have client data files, whereas REGULAR disks (or disks in non-quorum failure groups) have no such restriction.

For Oracle Clusterware files a minimum of three disk devices or three failure groups is required with a normal redundancy disk group. A QUORUM failure group is not considered when determining redundancy requirements in respect to storing user data.

The COMPATIBLE.ASM disk group compatibility attribute must be set to 11.2 or greater to store OCR or voting files in a disk group.

```
CREATE DISKGROUP ocr_data NORMAL REDUNDANCY
    FAILGROUP fg1 DISK '/devices/diskg1'
    FAILGROUP fg2 DISK '/devices/diskg2'
    QUORUM FAILGROUP fg3 DISK '/devices/diskg3'
    ATTRIBUTE 'compatible.asm' = '11.2.0.0.0';
```

5.3. 查看磁盘组磁盘

SQL> SELECT NAME, PATH, MODE_STATUS, STATE, DISK_NUMBER FROM V$ASM_DISK;

```
NAME                           PATH                                     MODE_ST STATE    DISK_NUMBER
------------------------------ ---------------------------------------- ------- -------- -----------
DATADG_0001                    /dev/asm-data3                           ONLINE  NORMAL             1
DATADG_0000                    /dev/asm-data1                           ONLINE  NORMAL             0
GRIDDG_0001                    /dev/asm-grid2                           ONLINE  NORMAL             1
GRIDDG_0002                    /dev/asm-grid3                           ONLINE  NORMAL             2
GRIDDG_0000                    /dev/asm-grid1                           ONLINE  NORMAL             0
                               /oracle/asm/fakeasm1                     ONLINE  NORMAL             6
                               /dev/asm-data2                           ONLINE  NORMAL             1
```

可以为 ASM 磁盘指定名称，否则会进行自动命令。

5.4. 增加磁盘

加入一个磁盘到现有的 Diskgroup 来扩空间和增加吞吐量是很常见的需求。最简单的加入磁盘命令如：alter diskgroup Data add disk ‘/dev/asm-disk5’；如前文所述在 RAC cluster 中如果一个磁盘不是在整个 cluster 范围内都可见，那么这个磁盘无法被加入到 RAC 的 ASM DISKGROUP 中。

如果 add disk 指定的磁盘的 disk header 发现了其他 diskgroup 的信息或者操作系统的一些信息，则需要 alter diskgroup Data add disk ‘/dev/asm-disk5’ force；加入 FORCE 选项。实际使用中尽可能避免使用 FORCE 选项。

需要注意的事 add disk 命令返回后只代表 disk header 已经完成必要的 metadata 写入，但不代表该磁盘已经完成了 rebalance 操作。后续的 rebalance 会被引发并移动数据到新加入的磁盘中。一般推荐如果你要加入多个 ASM DISK，那么在同一时间加入，而不是分多次加入。 但是一般不推荐同时做 add disk 和 drop disk。

```
ALTER DISKGROUP data1 ADD DISK
      '/devices/diska5' NAME diska5,
      '/devices/diska6' NAME diska6,
      '/devices/diska7' NAME diska7,
      '/devices/diska8' NAME diska8;
```

增加磁盘 add disk 的命令将针对指定的 discovery strings 去识别磁盘，若此时发现的磁盘已经是 disk group 的一部分，则将被默许为忽略掉。 磁盘将允许被加入到 diskgroup，前提是：

- 该磁盘不能是已经 mount 的 diskgroup 的一部分
- 必须没有有效的 ASM disk header，除非使用了 FORCE 选项
- 必须没有有效的 ORACLE 数据文件头，除非使用了 FORCE 选项
- FORCE 选项 如非必须建议用户不要用， 避免滥用
- 其必须不能以 2 个不同的路径名同时可见
- 其必须能被 asm_diskstring 所匹配到

当所有的磁盘均被以上验证过后，下面的步骤将被执行：

- Disk Directory 中加入对应该磁盘的记录
- Free Space block 和 allocation table 将写入到该磁盘
- Disk header 将被以当前时间戳更新

- 磁盘将被加入到 PST，但还没有 partners，但是其已经 ONLINE 可做读写。这将让磁盘真正成为 diskgroup 的一份，即便发生实例 crash。
- 一次 rebalance 将被启动，这将给新的磁盘找 partners，并将数据移动到其上。 一般推荐一次加多个磁盘，而非一次次地加。

当 rebalance 开始时这个 add disk 操作就被返回。磁盘并不完全参与到 disk group 中，直到 rebalance 结束。

5.5. 删除磁盘

可以从现有的 Diskgroup 里 drop 出 disk，这些 disk 可以用作它途；当然由于 asm disk 失败，导致 ASM 实例自动 drop 该失败的 asm disk 也是常见的。若一个 ASM DISK 常发生一些非致命的错误，则一般推荐将该 Disk drop 出来，以避免如果某天发生真的磁盘失败导致可能的数据丢失。但是需要注意 drop disk 时 不要指定其路径名，而是指定 ASM DISK NAME。

drop disk 命令可能较短时间内返回，但是 diskgroup 必须完成 rebalance 后这个磁盘才能被挪作他用。rebalance 将读取即将被 drop 掉 disk 的数据，并拷贝这些数据到其他磁盘上。FORCE 选项可以用于避免读取该正被 drop 的磁盘。该 FORCE 选项当磁盘发生失败或磁盘确实需要立即被挪用。原来那些需要被拷贝的 extent，使用 FORCE 选项后会从冗余的备份中读取，所以 external redundancy 不支持使用 FORCE 选项。当然如果使用 FORCE 选项最后会导致在 NORMAL/HIGH 冗余的 Diskgroup 下造成数据丢失的话，则 FORCE 选项也将不可用。

Do not reuse, remove, or disconnect the dropped disk until the HEADER_STATUS column for this disk in the V$ASM_DISK view changes to FORMER. You can query the V$ASM_OPERATION view to determine the amount of time remaining for the drop/rebalance operation to complete.

5.5.1. 常规 drop

常规 drop disk 下磁盘将被标记为不可再分配，且开始一个 rebalance。 drop 命令当 rebalance 开始时即返回。在 rebalance 过程中该 drop 中的磁盘将不断被移动其上的内容到其他磁盘上。 当 rebalance 完成时该磁盘将被从 disk group 中移除并可以复用。 可以通过查询 V$ASM_DISK 来确认磁盘是否还是 disk group 的一部分。

当 rebalance 还在进行中时，disk 将处于正被 drop 的状态，即 dropping。 还可以通过命令 alter diskgroup undrop 来反转这个还未完成的 drop 命令的效果。 如此则磁盘上不可再分配的标记将被移除，并会重启一个 rebalance。 这个重启的 rebalance 将重新评估我们正在 drop 的磁盘的 partnerships,并可能将数据 data extent 移回到正被 dropping 的磁盘上。 这个重启的 rebalance 仅仅需要撤销之前 rebalance 所做的工作即可， 因此其所耗时间取决于之前的 drop 工作的 rebalance 的工作量。 最后的分配情况可能与开始有着些许区别，但其仍将是平衡的。

5.5.2. 强制 drop

对于 Normal 或者 High Redundancy disk group 而言一个磁盘可以使用 FORCE 选项被 DROP。FORCE 选项对于 external redundancy 的 disk group 而言是不可用的， 原因是无法正常从被 drop 掉的 disk 上将数据重构出来。 对于 normal 或者 high redundancy 的 disk group 而言如果有一个或者多个磁盘 partners 已经 OFFLINE 了，则可能也不允许 FORCE DROP。 总是当 FORCE DROP 可能造成丢失文件上数据的时候都不允许使用。

FORCE DROP 会立即将磁盘状态置为 OFFLINE。该磁盘上的所有 extent 都将脱离其对应的 extent set 集合，这意味着冗余度的降低。 该磁盘将从 disk directory 中被移除，PST 中也是这样。

该磁盘的 disk header 将被写入信息来表明其不再是 disk group 的一部分。 rebalance 也会被启动。 当 drop force 命令返回时，意味着磁盘已经完全从 disk group 中移除了，可以被重用，也可以从操作系统上断开了。

发生操作的 disk group 上所有文件的冗余度要直到 rebalance 才能重新完善。 与常规的 drop 不同，显然 force drop 是无法被 undrop 的。磁盘将完全被从 disk group 移除，所以 undrop 也无法撤销此操作； 所能做的是将该磁盘重新加入到 diskgroup， add disk。

5.6. 取消删除磁盘

The UNDROP DISKS clause of the ALTER DISKGROUP statement enables you to cancel all pending drops of disks within disk groups. If a drop disk operation has completed, then this statement cannot be used to restore it.

```
ALTER DISKGROUP data1 UNDROP DISKS;
```

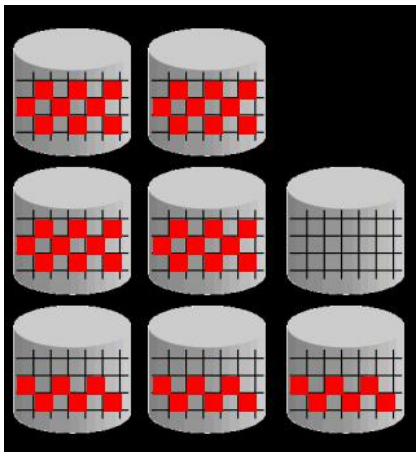
5.7. 调整磁盘容量

The RESIZE clause of ALTER DISKGROUP enables you to perform the following operations:

- Resize all disks in the disk group
- Resize specific disks
- Resize all of the disks in a specified failure group

```
ALTER DISKGROUP data1
      RESIZE DISKS IN FAILGROUP failgrp1 SIZE 100G;
```

5.8. Rebalance



rebalance diskgroup 将在 diskgroup 范围内将数据在其 DISK 上移动，以保证文件们均匀分布在 diskgroup 中的所有磁盘上，同时也会考虑到每一个 ASM DISK 的大小。 当文件均匀地分布在所有磁盘上，则各个磁盘的使用量也会很接近。如此以保证负载均衡。rebalance 的算法既不基于 I/O 统计信息也不基于其他统计结果； 完全取决于 Diskgroup 中 disk 的大小。

一旦 diskgroup 中发生了一些存储配置变化 例如 disk add/drop/resize 均会自动触发一次 rebalance。power 参数将决定有多少 slave 进程并发参数数据移动。所有的 slave 进程均从发生 rebalance 的实力启动并工作。rebalance 可以手动调控，即便已经在进行一次 rebalance 中了，也可以指定其他节点上的实例做 rebalance，只要管路员想要这样做。如果实例意外 crash，那么未结束的 rebalance 将自动重新启动。

注意 rebalance 中的每一次 extent 移动均会与数据库实例做协调，因为数据库实例可能同时需要读取或者写这个 extent，所以数据库在 rebalance 同时能正常工作。 其对数据库的影响一般较小，原因是同一时间只有一个 extent 被锁定以便移动，且仅仅是阻塞写入。

如何加快 asm rebalance 的速度，大概有如下几种方法：

1) 调大 asm_power_limit 参数

2) 将参数_disable_rebalance_compact 设置为 true,可动态调整

3) 设置 diskgroup 的 attributes 属性：_REBALANCE_COMPACT=false

4) 将参数_asm_imbalance_tolerance 调的更低(11gR2 默认为 3%)

4) 调整参数_disable_rebalance_space_check,关闭 compact 过程中的 space use 检查.

5) 调大_asm_rebalance_plan_size 参数,该参数控制 maximum rebalance work unit,通过调大该参数

6) 应该可以降低 extent relocation 的次数,但是这个也受限于系统的 io 能力.

```
ALTER DISKGROUP data2 REBALANCE POWER 5 WAIT;
```

5.9. Managing Capacity in Disk Groups

The V$ASM_DISKGROUP view contains the following columns that contain information to help you manage capacity:

- REQUIRED_MIRROR_FREE_MB indicates the amount of space that must be available in a disk group to restore full redundancy after the worst failure that can be tolerated by the disk group without adding additional storage.

- USABLE_FILE_MB indicates the amount of free space, adjusted for mirroring, that is available for new files to restore redundancy after a disk failure. Oracle ASM Mirroring and Disk Group Redundancy after a disk failure.

- TOTAL_MB is the total usable capacity of a disk group in megabytes. The calculations for data in this column take the disk header overhead into consideration. The disk header overhead depends on the number of Oracle ASM disks and Oracle ASM files. This value is typically about 1% of the total raw storage capacity.

- FREE_MB is the unused capacity of the disk group in megabytes, without considering any data imbalance

SQL> SELECT name, total_mb, free_mb, required_mirror_free_mb, usable_file_mb FROM V$ASM_DISKGROUP;

```
NAME         TOTAL_MB    FREE_MB REQUIRED_MIRROR_FREE_MB USABLE_FILE_MB
---------- ---------- ---------- ---------------------- --------------

DATA_DG          5120        650                      0             650
GRID_ACCT        1024        628                      0             628
```

TEST	5120	4996	0	4996

(FREE_MB - REQUIRED_MIRROR_FREE_MB) / 冗余类型 = USABLE_FILE_MB

EXTERN --1

NORMAL --2

HIGH --3

Negative Values of USABLE_FILE_MB

Due to the relationship between FREE_MB, REQUIRED_MIRROR_FREE_MB, and USABLE_FILE_MB, USABLE_FILE_MB can become negative. Although this is not necessarily a critical situation, it does mean that:

- Depending on the value of FREE_MB, you may not be able to create new files.
- The next failure might result in files with reduced redundancy.

If USABLE_FILE_MB becomes negative, it is strongly recommended that you add more space to the disk group as soon as possible.

The following guidelines help ensure that you have sufficient space to restore full redundancy for all disk group data after the failure of one or more disks.

- Normal redundancy disk group - It is best to have enough free space in your disk group to tolerate the loss of all disks in one failure group. The amount of free space should be equivalent to the size of the largest failure group.
- High redundancy disk group - It is best to have enough free space to cope with the loss of all disks in two failure groups. The amount of free space should be equivalent to the sum of the sizes of the two largest failure groups.

5.10. 磁盘组冗余

The redundancy levels are:

- External redundancy

Oracle ASM does not provide mirroring redundancy and relies on the storage system to provide RAID functionality. Any write error causes a forced dismount of the disk group. All disks must be located to successfully mount the disk group.

- Normal redundancy

Oracle ASM provides two-way mirroring by default, which means that all files are mirrored so that there are two copies of every extent. A loss of one Oracle ASM disk is tolerated.

You can optionally choose three-way or unprotected mirroring.A file specified with HIGH redundancy (three-way mirroring) in a NORMAL redundancy disk group provides additional protection from a bad disk sector, not protection from a disk failure.

- High redundancy

Oracle ASM provides triple mirroring by default. A loss of two Oracle ASM disks in different failure groups is tolerated.

5.10.1. Failure Groups

Failure groups are used to store mirror copies of data. When Oracle ASM allocates an extent for a normal redundancy file, Oracle ASM allocates a primary copy and a secondary copy. Oracle ASM chooses the disk on which to store the secondary copy so that it is in a different failure group than the primary copy.

```
ALTER DISKGROUP data SET ATTRIBUTE 'disk_repair_time' = '4.5h';
ALTER DISKGROUP data SET ATTRIBUTE 'disk_repair_time' = '270m';
```

After you repair the disk, run the SQL statement ALTER DISKGROUP ONLINE DISK. This statement brings a repaired disk group back online to enable writes so that no new writes are missed.

```
ALTER DISKGROUP data OFFLINE DISK DATA_001;
ALTER DISKGROUP data OFFLINE DISK IN FAILGROUP FG2;
ALTER DISKGROUP data ONLINE DISK IN FAILGROUP FG2;
ALTER DISKGROUP data ONLINE DISK DATA_001;
ALTER DISKGROUP data ONLINE ALL;
```

5.11. Oracle ASM Storage Limits

Oracle ASM has the following limits on the number of disk groups, disks, and files:

- 63 disk groups in a storage system
- 10,000 Oracle ASM disks in a storage system
- 1 million files for each disk group

Without any Oracle Exadata Storage, Oracle ASM has these storage limits:

- 2 terabytes (TB) maximum storage for each Oracle ASM disk
- 20 petabytes (PB) maximum for the storage system

With all Oracle Exadata Storage, Oracle ASM has these storage limits:

- 4 PB maximum storage for each Oracle ASM disk
- 40 exabytes (EB) maximum for the storage system

Maximum Oracle ASM file sizes for disk groups with AU_SIZE equal to 1 MB

Redundancy	COMPATIBLE.RDBMS = 10.1	COMPATIBLE.RDBMS >= 11.1
External	16 TB	140 PB
Normal	5.8 TB	23 PB
High	3.9 TB	15 PB

5.12. 挂载磁盘组

Mount Disk Group 使 Disk Group 其对本地 ASM 实例和连接到该实例的数据库实例可用。在该 diskgroup 中的文件在 OPEN/create/delete 之前必须先被本地 ASM 实例 mount；一般启动 ASM 时同时 mount 多个 diskgroup 较高效。典型情况下是 ASM_DISKGROUPS 匹配到的所有的 diskgroup 均通过 ALTER DISKGROUP ALL MOUNT 在 ASM 实例启动时被 mount 。

5.12.1. 磁盘组挂载命令

ALTER DISKGROUP ALL MOUNT;

ALTER DISKGROUP data1 MOUNT;

ALTER DISKGROUP data1 MOUNT FORCE;

5.12.2. 磁盘组挂载流程

1. Discovery

会通过 ASM_DISKSTRING.做一个 deep discovery； 每一个 disk header 均包含了其所属于的 diskgroup；该步骤应当要找到所有要被 mount 的 diskgroup 下属的所有磁盘。在 disk header 上获得如下信息：

- Disk Name
- Disk number

- 最后一次 mount 的 timestamp
- Disk Group Name

当 discovery 时若发现 2 个磁盘的 disk header 一样则可能报错，这样做的目的是为了避免损坏 disk group。

注意从 diskgroup 创建之后每一个 ASM DISK 的 OS 设备名可能发生变化，或者在集群中的每个节点上都不一样，这不要紧只需要 discovery 能找到它们并通过验证即可。

2. 第一次 mount 的实例

会通过 Instance Lock 实例锁来判断 ASM 实例是否是第一个 mount 该 diskgroup 的，还是已经被其他 ASM 实例 mount 了。如果是第一个做 mount 的，那么锁会被以排他持有直到 mount disk group 初始化完成，以防止其他实例也在该过程中做 mount。 如果不是第一个 mount 的，那么要等第一个 mount 的 ASM 完成其 mount 操作。

3. PST discovery

当 diskgroup 的一组磁盘被找到，必须要找到包含 PST 的那些磁盘。 每一个磁盘上的 AUN=1 的第一个块将被读取。这样来识别那些盘在 AUN=1 中存有 PST 拷贝。必须找到多数相同的 PST 拷贝来保证读出一个有效的 PST。

例如如果有 5 个 PST， 则需要找到 3 份内容一样的 PST 并读出。

一旦 PST 被读取后，ASM 实例将知道 mount disk group 必须要哪些个些 disk number。

4. Heartbeat

如果是第一个 mount 的实例，那么会做一个 heartbeat check 心跳检查。这是为了防止 2 个不同主机上的实例都认为其实第一个 mount 该 diskgroup 的，这种现象可能发生在 lock manager 配置不当的场景中。当 disk group 被实例 mount 时，PST 表上的最后一个块即心跳块每 3 秒被写入新的值，这种写入是由已经 mount 该 DG 的实例中的一个执行 。 若一个实例自认为第一个 mount，但是缺发现了 heartbeat 则其 mount 失败。若其发现没有 heartbeat，则其将开始 heartbeat。

5. Header validation

若是第一个 mount dg 的实例则一个新的 mount 时间戳 timestamp 被写入到各个磁盘。若不是第一个 mount 的实例，则会验证该 mount timestamp。这保证 2 个实例可能找到对一个磁盘的多份完全相同的拷贝时，仍能分辨出其实是不同的磁盘。

6. Redo recovery

若本实例时第一个 mount DG 的， 则其有义务做 crash recovery。若 ACD 中的任意 redo thread 在他们的检查点记录中标记为打开状态，则它们需要被 recover。 这个工序和数据库的 crash recovery 很像。在检查点和最后写入记录之间的 redo 将被扫描，来找出那些块需要恢复。这些块将被读取并应用 redo。

7. Redo thread selection

ACD 中需要找出一块未使用的区域来存放本实例生成的 redo。若是第一个 mount DG 的实例则需要保证所有 thread 都处于关闭状态，由此最小的 thread 将必然可用。 若不是第一个 MOUNT DG 的实例则可能整个 ACD 均已被使用。 若遇到此场景，则 mount 中的实例将要求已 mount 实例去扩展 ACD。一旦 ACD 扩容了新区域以便存放生成的 redo，则另一个 redo thread 将以写出到 checkpoint block 的形式来标记为 OPEN。

8. First done

若是第一个 mount DG 的实例，将开始允许其他实例也能 mount 该 DG。 若第一个实例到这一步之前就 crash 了，则其他实例将认为自己是第一个 mount DG 的实例。则若在多于 2 个实例的集群中后续的 mount 可以并行运行了。

9. Registration

实例将自己已 mount 的 DG 信息注册到 CSS 中。尝试访问这些 DG 的数据库实例将发现这些 CSS 注册信息并连接到 ASM 实例以便访问 DG。

10. COD recovery

若是第一个 mount DG 的实例，其会检查 COD 中的记录，若发现任何操作需要回滚，则将其回滚。若有一个 rebalance 作业仍在过程中，则该实例的 RBAL 将重新启动 rebalance。

5.13. 卸载磁盘组

```
ALTER DISKGROUP ALL DISMOUNT;
ALTER DISKGROUP data1 CHECK ALL;
```

5.14. 删除磁盘组

```
DROP DISKGROUP data1;
```

DROP DISKGROUP data1 FORCE;

Diskgroup 可以被 drop 掉的前提是其上所有的文件都处于关闭状态且仅有本地实例在 mount 它。 可以通过在集群件的所有 ASM 上通信来确认这 2 点。drop diskgroup 会在该 DG 下所有的磁盘头写入 header_status 为 FORMER 状态。

5.15. Renaming Disks Groups

The renamedg tool enables you to change the name of a cloned disk group. The disk group must be dismounted on all nodes in the cluster before running renamedg on the disk group.

需要先对磁盘组先做 DISMOUNT 操作，才能使用 renamedg 命令。

[grid@oeldb1 ~]$ renamedg dgname=SKY newdgname=PARADISE asm_diskstring='/dev/asm-diskd','/dev/asm-diske' verbose=true

```
NOTE: No asm libraries found in the system


Parsing parameters..

Parameters in effect:


        Old DG name       : SKY

        New DG name          : PARADISE

        Phases               :

                Phase 1

                Phase 2

        Discovery str        : /dev/asm-diskd,/dev/asm-diske

        Clean              : TRUE

        Raw only           : TRUE

renamedg operation: dgname=SKY newdgname=PARADISE asm_diskstring=/dev/asm-diskd,/dev/asm-diske verbose=true

Executing phase 1

Discovering the group

Performing discovery with string:/dev/asm-diskd,/dev/asm-diske

Identified disk UFS:/dev/asm-diskd with disk number:0 and timestamp (32993451 -1189735424)
```

```
Identified disk UFS:/dev/asm-diske with disk number:1 and timestamp (32993451 -1174044672)
Checking for hearbeat...
Re-discovering the group
Performing discovery with string:/dev/asm-diskd,/dev/asm-diske
Identified disk UFS:/dev/asm-diskd with disk number:0 and timestamp (32993451 -1189735424)
Identified disk UFS:/dev/asm-diske with disk number:1 and timestamp (32993451 -1174044672)
Checking if the diskgroup is mounted or used by CSS
Checking disk number:0
Checking disk number:1
Generating configuration file..
Completed phase 1
Executing phase 2
Looking for /dev/asm-diskd
Modifying the header
Looking for /dev/asm-diske
Modifying the header
Completed phase 2
Terminating kgfd context 0x7f5f428c50a0
```

11G DG 的信息会显示到 CRS 当中。

[grid@oeldb1 ~]$ srvctl status diskgroup -g PARADISE -a -v

Disk Group PARADISE is running on oeldb1

Disk Group PARADISE is enabled

5.16. CHECK

The CHECK keyword performs the following operations:

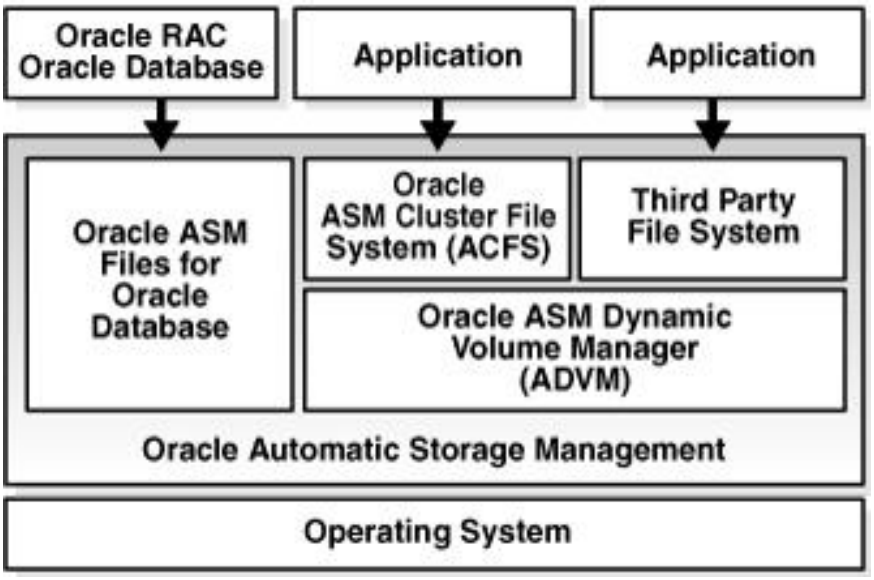
- Checks the consistency of the disk.
- Cross checks all the file extent maps and allocation tables for consistently.
- Checks that the alias metadata directory and file directory are linked correctly.
- Checks that the alias directory tree is linked correctly.

- Checks that Oracle ASM metadata directories do not have unreachable allocated blocks.

The ALTER DISKGROUP … CHECK command verifies that the Disk Directory and disk headers are consistent.

6. Oracle ACFS

Oracle Automatic Storage Management Cluster File System (Oracle ACFS) is a multi-platform, scalable file system, and storage management technology that extends Oracle Automatic Storage Management (Oracle ASM) functionality to support customer files maintained outside of Oracle Database.



Oracle ACFS file systems are generally mounted on all Oracle Cluster Synchronization Services (CSS) cluster members.

The Linux fuser or lsof commands or Windows handle command list information about processes and open files.

6.1. 卷管理常用命令

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.

6.2. Acfsutil 管理工具

[grid@rac11g1 ~]$ acfsutil

7. 常用视图

7.1. 数据字典

V$ASM_DISK_STAT and V$ASM_DISKGROUP_STAT. These two views are identical to V$ASM_DISK and V$ASM_DISKGROUP, but $ASM_DISK_STAT and V$ASM_DISKGROUP_STAT views are polled from memory and therefore do not require deep disk discovery.

View Name	X$ Table	Description
V$ASM_DISKGROUP	X$KFGRP	performs disk discovery and lists diskgroups
V$ASM_DISKGROUP_STAT	X$KFGRP_STAT	diskgroup stats without disk discovery
V$ASM_DISK	X$KFDSK, X$KFKID	performs disk discovery, lists disks and their usage metrics
V$ASM_DISK_STAT	X$KFDSK_STAT, X$KFKID	lists disks and their usage metrics
V$ASM_FILE	X$KFFIL	lists ASM files, including metadata/asmdisk files
V$ASM_ALIAS	X$KFALS	lists ASM aliases, files and directories
V$ASM_TEMPLATE	X$KFTMTA	lists the available templates and their properties
V$ASM_CLIENT	X$KFNCL	lists DB instances connected to ASM
V$ASM_OPERATION	X$KFGMG	lists rebalancing operations
N.A.	X$KFKLIB	available libraries, includes asmlib path

N.A.	X$KFDPARTNER	lists disk-to-partner relationships
N.A.	X$KFFXP	extent map table for all ASM files
N.A.	X$KFDAT	extent list for all ASM disks
N.A.	X$KFBH	describes the ASM cache (buffer cache of ASM in blocks of 4K (_asm_blksize)
N.A.	X$KFCCE	a linked list of ASM blocks. to be further investigated

11G 新增

View Name	X$ Table name	Description
V$ASM_ACFSSNAPSHOTS	X$KFVACFSS	snapshots of ACFS filesystems
V$ASM_ACFSVOLUMES	X$KFVACFSV	info on monted ACFS volumes
V$ASM_ACFS_ENCRYPTION_INFO	X$KFVACFSENCR	info on ACFS encryption config
V$ASM_ACFS_SECURITY_INFO	X$KFVACFSREALM	info on ACFS security (realm) config
V$ASM_ATTRIBUTE	X$KFENV	ASM DG attributes. Data stored in file #9 of each DG Notes: the X$ table shows also 'hidden' attributes,Example to turn off variable extents alter diskgroup set attribute '_extent_counts'='214748367 0 0';
V$ASM_DISK_IOSTAT	X$KFNSDSKIOST	I/O usage statistics
V$ASM_FILESYSTEM	X$KFVACFS	ACFS filesystems
V$ASM_USER	X$KFZUDR	os users info
V$ASM_USERGROUP	X$KFZGDR	creators of ASM file access control group
V$ASM_USERGROUP_MEMBER	X$KFZUAGR	members of ASM file access control groups
V$ASM_VOLUME	X$KFVOL, X$KFFIL	info on ADVM volumes created on ASM SGs

V$ASM_VOLUME_STAT	X$KFVOL,X$KFVOLSTAT	stats on ADVM volumes created on ASM SGs
N.A.	XXKFCBH	
N.A.	X$KFCLLE	
N.A.	X$KFDDD	
N.A.	X$KFDFS	
N.A.	X$KFFOF	reports the list of open files. it is the source for lsof in asmcmd
V$ASM_OPERATION in 11g	X$KFGBRB	
N.A.	X$KFGBRW	
N.A.	X$KFKLSOD	reports the list of open devices. it is the source for lsod in asmcmd
N.A.	X$KFMDGRP	
N.A.	X$KFRC	
N.A.	X$KFVOFS	no more there in 11.2.0.3
N.A.	X$KFVOFSV	no more there in 11.2.0.3

7.2. X$KFFXP (metadata, file extent pointers)

X$KFFXP 是 ASM(Automatic Storage Management)自动存储管理特性的重要内部视图，该视图反应了 File Extent Map 映射关系，ASM 会将文件 split 成多个多个 piece 分片，这些分片被称为 Extents。 在 Disk 上存放这些 Extent 的位置，就是我们常说的”Allocation Unit”。

KFF 意为 Kernel File，X$KFFXP 即 Kernel File Extent Maps， 该内部视图的一条记录代表一个 Extent。

X$KFFXP Column Name	Description
ADDR	x$ table address/identifier

INDX	row unique identifier
INST_ID	instance number (RAC)
GROUP_KFFXP	ASM disk group number. Join with v$asm_disk and v$asm_diskgroup
NUMBER_KFFXP	ASM file number. Join with v$asm_file and v$asm_alias
COMPOUND_KFFXP	File identifier. Join with compound_index in v$asm_file
INCARN_KFFXP	File incarnation id. Join with incarnation in v$asm_file
PXN_KFFXP	Progressive file extent number
XNUM_KFFXP	ASM file extent number (mirrored extent pairs have the same extent value) a value of 2147483648 is for the triple-mirrored file metadata
DISK_KFFXP	Disk number where the extent is allocated. Join with v$asm_disk can have the value 65534 when AU not present on physical storage (applies to normal or high redundancy DG)
AU_KFFXP	Relative position of the allocation unit from the beginning of the disk. The allocation unit size (1 MB) in v$asm_diskgroup can have the value 4294967294 when AU not present on physical storage because of failure for example (applies to normal or high redundancy DG)
LXN_KFFXP	0->primary extent, ->mirror extent, 2->2nd mirror copy (high redundancy and metadata)
FLAGS_KFFXP	N.K.
CHK_KFFXP	N.K.
SIZE_KFFXP	11g, to support variable size AU, integer value which marks the size of the extent in AU size units. extent sizes are determined by the diskgroup parameter _extent_sizes, the default value in 11gR2 and 12c this is: '1 4 16' and the extent sizes by _extent_counts, default= 20000 20000 214748367, that is the first 20000 extents have size 1 AU, then the next 20000 extents have size 4 AUs, all the subsequent extents have size 16 AUs.

```
SQL> SELECT GROUP_NUMBER, NAME, FILE_NUMBER FROM V$ASM_ALIAS WHERE ALIAS_DIRECTORY = 'N';
           1 spfile.265.880416171                 265
```

```
SQL> SELECT GROUP_KFFXP, DISK_KFFXP, AU_KFFXP
  FROM X$KFFXP
 WHERE NUMBER_KFFXP = 265;

SQL> SELECT DISK_NUMBER, PATH
  FROM V$ASM_DISK
 WHERE GROUP_NUMBER = 1
   AND DISK_NUMBER IN (0);
```

[grid@oracle11 ~]$ kfed read /dev/asm-data1 aun=1570

```
        Repeat 23 times
7FD9959FB5F0 00000000 00000000 00000000 00004301  [............C..]
7FD9959FB600 00002243 00000002 00000000 04010000  [C"............]
7FD9959FB610 00007667 2E796B73 62645F5F 6361635F  [gv..sky.__db_cac]
7FD9959FB620 735F6568 3D657A69 36333932 38323130  [he_size=29360128]
7FD9959FB630 6B730A30 5F5F2E79 6176616A 6F6F705F  [0.sky.__java_poo]
7FD9959FB640 69735F6C 343D657A 33343931 730A3430  [l_size=4194304.s]
7FD9959FB650 5F2E796B 72616C5F 705F6567 5F6C6F6F  [ky.__large_pool_]
7FD9959FB660 657A6973 3833383D 38303638 796B730A  [size=8388608.sky]
7FD9959FB670 6F5F5F2E 6C636172 61625F65 273D6573  [.__oracle_base=']
7FD9959FB680 61726F2F 2F656C63 2F707061 6361726F  [/oracle/app/orac]
7FD9959FB690 2327656C 4341524F 425F454C 20455341  [le'#ORACLE_BASE ]
7FD9959FB6A0 20746573 6D6F7266 766E6520 6E6F7269  [set from environ]
7FD9959FB6B0 746E656D 796B730A 705F5F2E 615F6167  [ment.sky.__pga_a]
7FD9959FB6C0 65726767 65746167 7261745F 3D746567  [ggregate_target=]
7FD9959FB6D0 36323732 36373932 6B730A30 5F5F2E79  [272629760.sky.__]
7FD9959FB6E0 5F616773 67726174 343D7465 37333136  [sga_target=46137]
7FD9959FB6F0 30343433 796B730A 735F5F2E 65726168  [3440.sky.__share]
7FD9959FB700 6F695F64 6F6F705F 69735F6C 303D657A  [d_io_pool_size=0]
7FD9959FB710 796B730A 735F5F2E 65726168 6F705F64  [.sky.__shared_po]
7FD9959FB720 735F6C6F 3D657A69 38363431 34363030  [ol_size=14680064]
7FD9959FB730 6B730A30 5F5F2E79 65727473 5F736D61  [0.sky.__streams_]
7FD9959FB740 6C6F6F70 7A69735F 0A303D65 75612E2A  [pool_size=0.*.au]
7FD9959FB750 5F746964 656C6966 7365645F 2F273D74  [dit_file_dest='/]
7FD9959FB760 6361726F 612F656C 6F2F7070 6C636172  [oracle/app/oracl]
7FD9959FB770 64612F65 2F6E696D 2F796B73 6D756461  [e/admin/sky/adum]
7FD9959FB780 2A0A2770 6475612E 745F7469 6C696172  [p'.*.audit_trail]
7FD9959FB790 6264273D 2E2A0A27 706D6F63 62697461  [='db'.*.compatib]
7FD9959FB7A0 273D656C 322E3131 342E302E 0A27302E  [le='11.2.0.4.0'.]
7FD9959FB7B0 6F632E2A 6F72746E 69665F6C 3D73656C  [*.control_files=]
7FD9959FB7C0 41442B27 47444154 796B732F 6E6F632F  ['+DATADG/sky/con]
7FD9959FB7D0 6C6F7274 656C6966 7275632F 746E6572  [trolfile/current]
7FD9959FB7E0 3036322E 3038382E 39353134 0A273932  [.260.880415929'.]
7FD9959FB7F0 62642E2A 6F6C625F 735F6B63 00004301  [*.db_block_s.C..]
7FD9959FB800 00002243 00000003 00000000 04010000  [C"............]
7FD9959FB810 0000351E 3D657A69 32393138 642E2A0A  [.5..ize=8192.*.d]
7FD9959FB820 72635F62 65746165 6C69665F 65645F65  [b_create_file_de]
7FD9959FB830 273D7473 5441442B 27474441 642E2A0A  [st='+DATADG'.*.d]
7FD9959FB840 6F645F62 6E69616D 0A27273D 62642E2A  [b_domain=''.*.db]
7FD9959FB850 6D616E5F 73273D65 0A27796B 69642E2A  [_name='sky'.*.di]
7FD9959FB860 6F6E6761 63697473 7365645F 2F273D74  [agnostic_dest='/]
7FD9959FB870 6361726F 612F656C 6F2F7070 6C636172  [oracle/app/oracl]
```

```
7FD9959FB880 2A0A2765 7369642E 63746170 73726568  [e'.*.dispatchers]
7FD9959FB890 5028273D 4F544F52 3D4C4F43 29504354  [='(PROTOCOL=TCP)]
7FD9959FB8A0 45532820 43495652 6B733D45 42445879  [ (SERVICE=skyXDB]
7FD9959FB8B0 2A0A2729 6D656D2E 5F79726F 67726174  [)'.*.memory_targ]
7FD9959FB8C0 373D7465 35383033 32373437 6F2E2A0A  [et=730857472.*.o]
7FD9959FB8D0 5F6E6570 73727563 3D73726F 0A303033  [pen_cursors=300.]
7FD9959FB8E0 72702E2A 7365636F 3D736573 0A303531  [*.processes=150.]
7FD9959FB8F0 65722E2A 65746F6D 676F6C5F 705F6E69  [*.remote_login_p]
7FD9959FB900 77737361 6664726F 3D656C69 43584527  [asswordfile='EXC]
7FD9959FB910 4953554C 0A274556 6E752E2A 745F6F64  [LUSIVE'.*.undo_t]
7FD9959FB920 656C6261 63617073 55273D65 544F444E  [ablespace='UNDOT]
7FD9959FB930 27315342 0000000A 00000000 00000000  [BS1'............]
7FD9959FB940 00000000 00000000 00000000 00000000  [................]
```

7.3. X$KFDAT (metadata, disk-to-AU mapping table)

X$KFDAT Column Name	Description
ADDR	x$ table address/identifier
INDX	row unique identifier
INST_ID	instance number (RAC)
GROUP_KFDAT	diskgroup number, join with v$asm_diskgroup
NUMBER_KFDAT	disk number, join with v$asm_disk
COMPOUND_KFDAT	disk compund_index, join with v$asm_disk
AUNUM_KFDAT	Disk allocation unit (relative position from the beginning of the disk), join with x$kffxp.au_kffxp
V_KFDAT	V=this Allocation Unit is used; F=AU is free
FNUM_KFDAT	file number, join with v$asm_file
I_KFDAT	N.K.
H_KFDAT	11g, N.K.
XNUM_KFDAT	Progressive file extent number join with x$kffxp.pxn_kffxp
RAW_KFDAT	raw format encoding of the disk,and file extent information
SIZE_KFDAT	11g, N.K.
FMT_KFDAT	11g, N.K.

查看归档日志分别在 DG 的那个磁盘上面：

```
SELECT NAME, FILE_NUMBER FROM V$ASM_ALIAS WHERE GROUP_NUMBER = 1 AND NAME LIKE 'thread%';

SELECT DISTINCT DISK_KFFXP
  FROM X$KFFXP
 WHERE NUMBER_KFFXP IN
       (SELECT FILE_NUMBER FROM V$ASM_ALIAS WHERE GROUP_NUMBER = 1 AND NAME LIKE 'thread%')
       AND GROUP_KFFXP = 1;

SELECT DISK_KFFXP, COUNT(1)
  FROM X$KFFXP
 WHERE NUMBER_KFFXP IN (SELECT FILE_NUMBER
                        FROM V$ASM_ALIAS
                       WHERE GROUP_NUMBER = 1
                         AND NAME LIKE 'thread%')
   AND GROUP_KFFXP = 1
 GROUP BY DISK_KFFXP;
```

7.4. X$KFFIL and metadata files

Three types of metadata:

1. **diskgroup metadata**: files with NUMBER_KFFIL <256 ASM metadata and ASMlog files. These files have high redundancy (3 copies) and block size =4KB.
   a) ASM log files are used for ASM instance and crash recovery when a crash happens with metadata operations (see below COD and ACD)
   b) at diskgroup creation 6 files with metadata are visible from x$kffil
2. **disk metadata**: disk headers (typically the first 2 AU of each disk) are not listed in x$kffil (they appear as file number 0 in x$kfdat). Contain disk membership information. This part of the disk has to be 'zeroed out' before the disk can be added to ASM diskgroup as a new disk.
3. **file metadata**: 3 mirrored extents with file metadata, visible from x$kffxp and x$kfdat
   * note: metadata i triple mirrored if at least 3 failgroups are available

7.5. X$KFDPARTNER

```
SELECT NUMBER_KFDPARTNER, FAILGROUP
  FROM X$KFDPARTNER A, V$ASM_DISK b
 WHERE A.DISK = 2
   AND A.GRP = 1
   AND A.NUMBER_KFDPARTNER = B.DISK_NUMBER;
```

8. ASM 工具

8.1. ASMCMD (ASM command line utility)

ASMCMD 用于管理 ASM 系统。

[grid@oracle11 diag]$ asmcmd help

- CD 命令

cd + is equivalent to cd / in Unix.

- LS 命令

ASMCMD [+] > **ls -l**

```
State    Type    Rebal  Name
MOUNTED  EXTERN  N      DATADG/
MOUNTED  NORMAL  N      GRIDDG/
```

- du (disk usage)

ASMCMD [+] > **du DATADG/***

```
Used_MB      Mirror_used_MB
   3918                3918
```

- lsct (list the ASM clients)

ASMCMD [+] > **lsct**

```
DB_Name  Status     Software_Version  Compatible_version  Instance_Name  Disk_Group
+ASM     CONNECTED        11.2.0.4.0          11.2.0.4.0  +ASM           DATADG
sky      CONNECTED        11.2.0.4.0          11.2.0.4.0  sky            DATADG
```

- find

ASMCMD [+] > **find +DATADG undo***

```
+DATADG/SKY/DATAFILE/UNDOTBS1.258.880415833
```

- lsdsk

ASMCMD [+] > **lsdsk -p**

```
Group_Num  Disk_Num      Incarn  Mount_Stat  Header_Stat  Mode_Stat  State   Path
```

```
        1          0  3916001522  CACHED      MEMBER      ONLINE      NORMAL  /dev/asm-data1
        1          1  3916001521  CACHED      MEMBER      ONLINE      NORMAL  /dev/asm-data3
        2          0  3916001523  CACHED      MEMBER      ONLINE      NORMAL  /dev/asm-grid1
        2          1  3916001525  CACHED      MEMBER      ONLINE      NORMAL  /dev/asm-grid2
        2          2  3916001524  CACHED      MEMBER      ONLINE      NORMAL  /dev/asm-grid3
```

ASMCMD [+] > **lsdsk --candidate -p**

```
Group_Num  Disk_Num      Incarn  Mount_Stat  Header_Stat  Mode_Stat  State   Path
        0         0  3916001514  CLOSED      FORMER       ONLINE     NORMAL  /dev/asm-data2
        0         6  3916001520  CLOSED      CANDIDATE    ONLINE     NORMAL  /oracle/asm/fakeasm1
```

- -p 参数，可以显示执行命令的当前路径：

alias asmcmd='asmcmd -p'

[grid@oracle11 ~]$ **asmcmd -p**

```
ASMCMD [+] > ls
ASMCMD [+] > cd DATADG
ASMCMD [+DATADG] >
```

- 非交互模式

[grid@oracle11 ~]$ **asmcmd lsdsk -p -G DATADG**

```
Group_Num  Disk_Num      Incarn  Mount_Stat  Header_Stat  Mode_Stat  State   Path
        1         0  3916001522  CACHED      MEMBER       ONLINE     NORMAL  /dev/asm-data1
        1         1  3916001521  CACHED      MEMBER       ONLINE     NORMAL  /dev/asm-data3
```

- COPY

ASMCMD [+] > find +DATADG undo*

```
+DATADG/SKY/DATAFILE/UNDOTBS1.258.880415833
```

ASMCMD [+] > cp +DATADG/SKY/DATAFILE/UNDOTBS1.258.880415833 /tmp/undo.tbs

```
copying +DATADG/SKY/DATAFILE/UNDOTBS1.258.880415833 -> /tmp/undo.tbs
```

- ASM Metadata Backup and Restore (AMBR)

■ Backup Mode

With the backup mode, information is gathered about ASM disks, diskgroup and failure group configurations, templates, attributes, and alias directory structures.

This information is converted into SQL commands and stored in a user-defined metadata backup (MDB) file, which can be used by the md_restore command during a diskgroup restore.

asmcmd md_backup [-b <location_of_backup>] [-g dgname [-g dgname …]]

ASMCMD [+] > **md_backup /tmp/dgbackup20090716 -G DATADG**

```
Disk group metadata to be backed up: DATADG
Current alias directory path: SKY/PARAMETERFILE
Current alias directory path: SKY/DATAFILE
Current alias directory path: SKY/ONLINELOG
Current alias directory path: SKY
Current alias directory path: SKY/CONTROLFILE
Current alias directory path: SKY/TEMPFILE
```

■ Restore Mode

The essential task of the AMBR restore mode is to restore the metadata into the diskgroup; this is done using the md_restore subcommand of ASMCMD.

asmcmd md_restore [-t full|nodg|newdg] -f <backup_file> -g dgname [-g dgname] [－o <override>] －i

■ Backup and Recovery Example

1. Create some user-defined ASM directories, aliases, and templates.

```
SQL> alter diskgroup data add template temp_unprot attributes (fine unprotected)
SQL> alter diskgroup data add template important_data attributes (fine mirror);
SQL> alter diskgroup data add alias '+DATA/RAC/USERS_01.DBF' for '+DATA/RAC/
DATAFILE/USERS.259.609660473';
SQL> alter diskgroup data add directory '+DATA/RAC/oradata';
SQL> alter diskgroup data add alias '+DATA/RAC/oradata/sysaux_01.dbf' for
'+DATA/RAC/DATAFILE/SYSAUX.257.609660473';
SQL> alter diskgroup data add directory '+DATA/RAC/oradata/temp_files';
SQL> alter diskgroup data add alias '+DATA/RAC/oradata/temp_files/temp_01.dbf' for
'+DATA/RAC/tempfile/TEMP.263.609660687'

SQL> select name,REDUNDANCY,STRIPE from v$asm_template where system='N';
NAME REDUND STRIPE
```

```
---------------------------- ------ ------
TEMP_UNPROT UNPROT FINE
IMPORTANT_DATA MIRROR FINE

SQL> SELECT NAME,FILE_NUMBER,ALIAS_DIRECTORY FROM V$ASM_ALIAS
WHERE SYSTEM_CREATED='N' AND ALIAS_DIRECTORY='Y'
NAME FILE_NUMBER A
---------------------------------------------- ----------- -
oradata 4294967295 Y
temp_files 4294967295 Y
```

2. Execute the md_backup command to create the MDB file.

[oracle@racnode1]$ asmcmd md_backup -b datadg_backup -g data

3. Use RMAN to back up the database stored within that diskgroup.

[oracle@racnode1]$ rman > BACKUP DATABASE;

4. To simulate a complete diskgroup failure, drop the diskgroup.

SQL> ALTER DISKGROUP DATA DISMOUNT;

SQL> DROP DISKGROUP DATA INCLUDING CONTENTS;

5. Recover the diskgroup using md_restore.

[oracle@racnode1]$ asmcmd md_restore -t full -g DATA -b datadg_backup

6. Restore and recover the database using RMAN.

[oracle@racnode1]$ rman> restore database;

[oracle@racnode1]$ rman> recover database;

8.2. KFOD (Kernel Files OSM Disk)

该工具用于 ASM 磁盘的发现。

Note:

1) KFOD is used at installation time as well (by OUI,DBCA or ASMCA) in order to perform disk discovery.

2) In case of a failed installation (e.g no $GRID_HOME/bin existing yet) KFOD can be found under your stage folder: <stage_folder>/grid/stage/ext/bin/

In such cases you might need to set LD_LIBRARY_PATH to <stage_folder>/grid/stage/ext/lib.

8.2.1. 查看帮助信息

[grid@oracle11 ~]$ kfod -help

8.2.2. 发现 ASM 磁盘设备

[grid@oracle11 ~]$ **kfod asm_diskstring='/dev/asm*' disks=all**

```
--------------------------------------------------------------------------------
 Disk          Size Header    Path                                     Disk Group   User     Group
================================================================================
   1:       4096 Mb MEMBER    /dev/asm-data1                           DATADG       grid     asmadmin
   2:       4096 Mb CANDIDATE /dev/asm-data2                           #            grid     asmadmin
   3:       4096 Mb CANDIDATE /dev/asm-data3                           #            grid     asmadmin
   4:       1024 Mb MEMBER    /dev/asm-grid1                           GRIDDG       grid     asmadmin
   5:       1024 Mb MEMBER    /dev/asm-grid2                           GRIDDG       grid     asmadmin
   6:       1024 Mb MEMBER    /dev/asm-grid3                           GRIDDG       grid     asmadmin
--------------------------------------------------------------------------------
ORACLE_SID ORACLE_HOME
================================================================================
     +ASM /oracle/app/11.2.0/grid
```

8.3. KFED (Kernel Files metadata EDitor)

11G 之前，需要手工编译编译 KFED：

[grid@oracle11 ~]$ cd $ORACLE_HOME/rdbms/lib

[grid@oracle11 lib]$ make -f ins_rdbms.mk ikfed

KFED 工具用于当磁盘组无法挂载时，分析 ASM 磁盘头信息。

8.3.1. 读取磁盘头信息

当设备不属于磁盘组时，会出现 KFED-00322 错误：

[grid@oracle11 ~]$ **kfed read /dev/asm-data2**

```
kfbh.endian:                          0 ; 0x000: 0x00
kfbh.hard:                            0 ; 0x001: 0x00
kfbh.type:                            0 ; 0x002: KFBTYP_INVALID
kfbh.datfmt:                          0 ; 0x003: 0x00
kfbh.block.blk:                       0 ; 0x004: blk=0
kfbh.block.obj:                       0 ; 0x008: file=0
kfbh.check:                           0 ; 0x00c: 0x00000000
kfbh.fcn.base:                        0 ; 0x010: 0x00000000
kfbh.fcn.wrap:                        0 ; 0x014: 0x00000000
kfbh.spare1:                          0 ; 0x018: 0x00000000
kfbh.spare2:                          0 ; 0x01c: 0x00000000
7FEB1DB5C400 00000000 00000000 00000000 00000000  [................]
  Repeat 255 times
KFED-00322: Invalid content encountered during block traversal: [kfbtTraverseBlock][Invalid OSM block type][][0]
```

当你使用 kfed 时，不指定 AU，默认是读取 au 0，block 0(记住，asm 中 block 编号都是从 0 开始的)。

In ASM versions 11.1.0.7 and later, the ASM disk header block is backed up in the second last ASM metadata block in the allocation unit 1。

[grid@oracle11 ~]$ **kfed read /dev/asm-data1**

```
kfbh.endian:                          1 ; 0x000: 0x01  ---1 表示的是 Little Endian，0 的话即使表示 BIG endian
kfbh.hard:                          130 ; 0x001: 0x82
kfbh.type:                            1 ; 0x002: KFBTYP_DISKHEAD
kfbh.datfmt:                          1 ; 0x003: 0x01
kfbh.block.blk:                       0 ; 0x004: blk=0
kfbh.block.obj:              2147483648 ; 0x008: disk=0
kfbh.check:                  2479020390 ; 0x00c: 0x93c2d966
kfbh.fcn.base:                    4845 ; 0x010: 0x000012ed
kfbh.fcn.wrap:                        0 ; 0x014: 0x00000000
kfbh.spare1:                          0 ; 0x018: 0x00000000
kfbh.spare2:                          0 ; 0x01c: 0x00000000
kfdhdb.driver.provstr:         ORCLDISK ; 0x000: length=8
kfdhdb.driver.reserved[0]:            0 ; 0x008: 0x00000000
kfdhdb.driver.reserved[1]:            0 ; 0x00c: 0x00000000
```

```
kfdhdb.driver.reserved[2]:            0 ; 0x010: 0x00000000
kfdhdb.driver.reserved[3]:            0 ; 0x014: 0x00000000
kfdhdb.driver.reserved[4]:            0 ; 0x018: 0x00000000
kfdhdb.driver.reserved[5]:            0 ; 0x01c: 0x00000000
kfdhdb.compat:                186646528 ; 0x020: 0x0b200000
kfdhdb.dsknum:                        0 ; 0x024: 0x0000
kfdhdb.grptyp:                        1 ; 0x026: KFDGTP_EXTERNAL  ---磁盘组冗余方式
kfdhdb.hdrsts:                        3 ; 0x027: KFDHDR_MEMBER  ---磁盘 header 状态  3 是可用状态
```

下面对改值的属性做一下补充：

```
KFDHDR_INVALID     ((kfdhdr)0) --  Illegal value
KFDHDR_UNKNOWN     ((kfdhdr)1) --  Disk header block unreadable
KFDHDR_CANDIDATE   ((kfdhdr)2) --  No OSM or OS disk header found
KFDHDR_MEMBER      ((kfdhdr)3) --  Normal member of the group  ---03 正常状态
KFDHDR_FORMER      ((kfdhdr)4) --  Disk dropped cleanly from group
KFDHDR_CONFLICT    ((kfdhdr)5) --  Header conflicts
KFDHDR_INCOMPAT    ((kfdhdr)6) --  Written by incompatible software
KFDHDR_PROVISIONED ((kfdhdr)7) --  Disk was prepared beforehand
```

```
kfdhdb.dskname:             DATADG_0000 ; 0x028: length=11
kfdhdb.grpname:                   DATADG ; 0x048: length=6  ---磁盘组名称
kfdhdb.fgname:              DATADG_0000 ; 0x068: length=11
kfdhdb.capname:                         ; 0x088: length=0
kfdhdb.crestmp.hi:            33019607 ; 0x0a8: HOUR=0x17 DAYS=0x16 MNTH=0x5 YEAR=0x7df
kfdhdb.crestmp.lo:          3625030656 ; 0x0ac: USEC=0x0 MSEC=0x65 SECS=0x1 MINS=0x36
kfdhdb.mntstmp.hi:            33019821 ; 0x0b0: HOUR=0xd DAYS=0x1d MNTH=0x5 YEAR=0x7df
kfdhdb.mntstmp.lo:          3098928128 ; 0x0b4: USEC=0x0 MSEC=0x179 SECS=0xb MINS=0x2e
kfdhdb.secsize:                   512 ; 0x0b8: 0x0200
kfdhdb.blksize:                  4096 ; 0x0ba: 0x1000
kfdhdb.ausize:                1048576 ; 0x0bc: 0x00100000 ---au 单位大小，单位是 byte，大小为 1m.
kfdhdb.mfact:                  113792 ; 0x0c0: 0x0001bc80
kfdhdb.dsksize:                  4096 ; 0x0c4: 0x00001000 ---该 disk 的大小，单位是 au，由于默认 au 是 1m，所以大小为 4096m
```

补充：

kfdhdb.ausize * kfdhdb.dsksize = disk size

即是 1m x 4096 = 4096m (注意这个大小是整个磁盘组的大小)

```
kfdhdb.pmcnt:                         2 ; 0x0c8: 0x00000002
kfdhdb.fstlocn:                       1 ; 0x0cc: 0x00000001  ----Free Space Table (FST)的 AU 位置信息
kfdhdb.altlocn:                       2 ; 0x0d0: 0x00000002
kfdhdb.f1b1locn:                      2 ; 0x0d4: 0x00000002  ---File Directory 所在的 au 位置
kfdhdb.redomirrors[0]:                0 ; 0x0d8: 0x0000
kfdhdb.redomirrors[1]:            65535 ; 0x0da: 0xffff
```

```
kfdhdb.redomirrors[2]:             65535 ; 0x0dc: 0xffff
kfdhdb.redomirrors[3]:             65535 ; 0x0de: 0xffff
kfdhdb.dbcompat:               168820736 ; 0x0e0: 0x0a100000 --数据库版本
kfdhdb.grpstmp.hi:              33019607 ; 0x0e4: HOUR=0x17 DAYS=0x16 MNTH=0x5 YEAR=0x7df
kfdhdb.grpstmp.lo:            3624970240 ; 0x0e8: USEC=0x0 MSEC=0x2a SECS=0x1 MINS=0x36
kfdhdb.vfstart:                        0 ; 0x0ec: 0x00000000
kfdhdb.vfend:                          0 ; 0x0f0: 0x00000000
kfdhdb.spfile:                         0 ; 0x0f4: 0x00000000
kfdhdb.spfflg:                         0 ; 0x0f8: 0x00000000
kfdhdb.ub4spare[0]:                    0 ; 0x0fc: 0x00000000
......
kfdhdb.acdb.aba.seq:                   0 ; 0x1d4: 0x00000000
kfdhdb.acdb.aba.blk:                   0 ; 0x1d8: 0x00000000
kfdhdb.acdb.ents:                      0 ; 0x1dc: 0x0000
kfdhdb.acdb.ub2spare:                  0 ; 0x1de: 0x0000
```

8.3.2. 写磁盘头信息

读取 AU 信息到文件

[grid@oracle11 ~]$ kfed op=read dev=/dev/asm-data1 aunum=3 blknum=3 text=a.txt

将修改后的 AU 信息写入磁盘

[grid@oracle11 ~]$ kfed op=write dev=/dev/asm-data1 aunum=3 blknum=3 text=a.txt

8.3.3. 修改 DROP 的磁盘属性

$ kfed read /dev/mapper/devstor4_1p1 aunum=0 blknum=0 text=devstor4_1p1.txt

vi devstor4_1p1.txt and change

from:

```
kfdhdb.hdrsts:                        4 ; 0x027: KFDHDR_FORMER
```

to:

```
kfdhdb.hdrsts:                        3 ; 0x027: KFDHDR_MEMBER
```

$ kfed write /dev/mapper/devstor4_1p1 aunum=0 blknum=0 text=devstor4_1p1

8.3.4. 查看块类型

[grid@rac11g1 ~]$ kfed find /dev/asm-grid aun=0 | more
Block 0 has type 1

[grid@rac11g1 ~]$ kfed find /dev/asm-grid aun=1
Block 510 has type 1

TYPE 为 1 的类型为磁盘头信息。

[grid@rac11g1 ~]$ kfed read /dev/asm-grid aun=0 blkn=0 | grep 'kfbh.type'

```
kfbh.type:                                1 ; 0x002: KFBTYP_DISKHEAD
```

8.3.5. 修复磁盘头

利用 ASM 自动备份的磁盘头信息，修复 ASM 磁盘头。

[grid@rac11g1 ~]$ kfed repair /dev/asm-data3

8.4. AMDU (ASM Metadata Dump Utility)

AMDU 用于抽取 ASM 磁盘的元数据信息。该工具不需要磁盘组被挂载。

8.4.1. 抽取磁盘组信息

[grid@oracle11 ~]$ **amdu -diskstring '/dev/asm*' -dump 'DATADG'**

```
amdu_2015_05_27_09_28_21/
AMDU-00204: Disk N0001 is in currently mounted diskgroup DATADG
AMDU-00201: Disk N0001: '/dev/asm-data1'
```

8.4.2. 抽取数据文件

[grid@oracle11 ~]$ **amdu -diskstring '/dev/asm*' -extract DATADG.259**

```
amdu_2015_05_30_15_00_30/
AMDU-00204: Disk N0001 is in currently mounted diskgroup DATADG
```

```
AMDU-00201: Disk N0001: '/dev/asm-data1'
```

[grid@oracle11 ~]$ cd amdu_2015_05_30_15_00_30/

[grid@oracle11 amdu_2015_05_30_15_00_30]$ ls -lrt

```
total 5140
-rw-r--r-- 1 grid oinstall 5251072 May 30 15:00 DATADG_259.f
-rw-r--r-- 1 grid oinstall    8763 May 30 15:00 report.txt
```

[grid@oracle11 amdu_2015_05_30_15_00_30]$ **dbv file=DATADG_259.f**

```
DBVERIFY: Release 11.2.0.4.0 - Production on Sat May 30 15:01:22 2015
Copyright (c) 1982, 2011, Oracle and/or its affiliates.  All rights reserved.
DBVERIFY - Verification starting : FILE = /home/grid/amdu_2015_05_30_15_00_30/DATADG_259.f
DBVERIFY - Verification complete

Total Pages Examined         : 640
Total Pages Processed (Data) : 15
Total Pages Failing   (Data) : 0
Total Pages Processed (Index): 2
Total Pages Failing   (Index): 0
Total Pages Processed (Other): 590
Total Pages Processed (Seg)  : 0
Total Pages Failing   (Seg)  : 0
Total Pages Empty            : 33
Total Pages Marked Corrupt   : 0
Total Pages Influx           : 0
Total Pages Encrypted        : 0
Highest block SCN            : 921351 (0.921351)
```

8.4.3. 抽取未挂载磁盘组中的文件

SQL> show parameter control;

+DATADG/sky/controlfile/current.260.880415929

[grid@oracle11 ~]$ **amdu -dis '/dev/asm-data*' -extract DATADG.260**

260 号文件，为控制文件。

[grid@oracle11]$ **strings DATADG_260.f | grep DATADG**

```
+DATADG/sky/onlinelog/group_3.263.880415933
```

```
+DATADG/sky/onlinelog/group_2.262.880415931
+DATADG/sky/onlinelog/group_1.261.880415931
+DATADG/sky/datafile/users.259.880415833
+DATADG/sky/datafile/undotbs1.258.880415833
+DATADG/sky/datafile/sysaux.257.880415833
+DATADG/sky/datafile/system.256.880415831
+DATADG/sky/tempfile/temp.264.880415939
+DATADG/sky/onlinelog/group_3.263.880415933
+DATADG/sky/onlinelog/group_2.262.880415931
+DATADG/sky/onlinelog/group_1.261.880415931
+DATADG/sky/datafile/users.259.880415833
+DATADG/sky/datafile/undotbs1.258.880415833
+DATADG/sky/datafile/sysaux.257.880415833
+DATADG/sky/datafile/system.256.880415831
+DATADG/sky/tempfile/temp.264.880415939
```

8.5. BBED

cd $ORACLE_HOME/rdbms/lib

make -f ins_rdbms.mk $ORACLE_HOME/rdbms/lib/bbed

Dump an extent or an ASM file with the methods described above, for example:

amdu -dis '/dev/mapper/itsto*p1' -extract TEST4_DATADG1.555

Edit: vi bbed.par

blocksize=8192

datafile=/ORA/dbs01/oracle/home/work/amdu_2010_02_01_17_22_39/TEST2_DATADG1_555.f

mode=browse

Finally:

$ORACLE_HOME/rdbms/lib/bbed parfile=bbed.par

8.6. ASM Oracle kernel components and prefixes

SQL> **oradebug doc component;**

SQL> **oradebug doc component asm;**

```
ASM                     Automatic Storage Management (kf)
  KFK                   KFK (kfk)
    KFKIO               KFK IO (kfkio)
    KFKSB               KFK subs (kfksubs)
  KFN                   ASM Networking subsystem (kfn)
    KFNU                ASM Umbillicus (kfnm, kfns, kfnb)
    KFNS                ASM Server networking (kfns)
    KFNC                ASM Client networking (kfnc)
  KFIS                  ASM Intelligent Storage interfaces (kfis)
  KFM                   ASM Node Monitor Interface Implementation (kfm)
    KFMD                ASM Node Monitor Layer for Diskgroup Registration (kfmd)
    KFMS                ASM Node Monitor Layers Support Function Interface (kfms)
  KFFB                  ASM Metadata Block (kffb)
  KFFD                  ASM Metadata Directory (kffd)
  KFZ                   ASM Zecurity subsystem (kfz)
  KFC                   ASM Cache (kfc)
  KFR                   ASM Recovery (kfr)
  KFE                   ASM attributes (kfe)
  KFDP                  ASM PST (kfdp)
  KFG                   ASM diskgroups (kfg)
  KFDS                  ASM staleness registry and resync (kfds)
  KFDX                  ASM Exadata interface (kfdx)
```

8.7. 跟踪 ASMCMD 命令

[grid@oracle11 ~]$ **export DBI_TRACE=1**

[grid@oracle11 ~]$ asmcmd

DBI 1.602-ithread default trace level set to 0x0/1 (pid 31570) at DBI.pm line 273 via asmcmdshare.pm line 205

从上面的返回信息可以看到，整体是通过 perl 脚本来实现的。

8.8. DBMS_DISKGROUP 包

dbms_diskgroup is an Oracle 'internal package' (C implementation, as opposed to PL/SQL), it provides and API to access ASM data.

[oracle@oracle11 ~]$ find $ORACLE_HOME -name asmcmd* | xargs grep -i dbms_diskgroup

9. ASM 迁移

9.1. 使用 RMAN 进行迁移

RMAN> SQL "ALTER TABLESPACE EXAMPLE OFFLINE";

RMAN> BACKUP AS COPY TABLESPACE EXAMPLE FORMAT '+ASMGRP1';

RMAN> SWITCH TABLESPACE EXAMPLE TO COPY;

RMAN> SQL "ALTER TABLESPACE EXAMPLE ONLINE";

9.2. DBMS_FILE_TRANSFER

SQL> SELECT FILE_NAME FROM DBA_DATA_FILES;

SQL> ALTER DATABASE DATAFILE '+ASMGRP1/SSKYDB/DATAFILE/BMF_DATA.273.572018897' OFFLINE;

SQL> CREATE DIRECTORY ASMSRC AS '+ASMGRP1/SSKYDB/DATAFILE';
SQL> CREATE DIRECTORY OSDEST AS '/ocfs9/oradata';

SQL>

```
BEGIN
   DBMS_FILE_TRANSFER.COPY_FILE('ASMSRC', 'BMF_DATA.273.572018897', 'OSDEST',
'BMF.dbf');
END;
/
```

SQL> ALTER DATABASE DATAFILE '+ASMGRP1/SSKYDB/DATAFILE/BMF_DATA.273.572018897' ONLINE;

9.3. XML DB FTP

使用$ORACLE_HOME/rdbms/admin/catqm.sql 创建 XML DB。

9.4. XML DB FTP

9.5. ASM 迁移至文件系统

--创建 pfile 文件

SQL> create pfile ='/tmp/pfile' from spfile;

File created.

SQL> exit

Disconnected from Oracle Database 10g Enterprise Edition Release 10.2.0.3.0 - Production

With the Partitioning, OLAP and Data Mining options

--修改 pfile 中关于 asm 中的内容

control_files

db_recovery_file_dest

log_archive_dest_1

指定到文件系统

--登录 rman

[oracle@localhost tmp]$ rman target /

Recovery Manager: Release 10.2.0.3.0 - Production on Mon Jun 27 12:48:26 2011

Copyright (c) 1982, 2005, Oracle. All rights reserved.

connected to target database: TOS (DBID=1569606545)

--执行 backup as copy datafile

RMAN> backup as copy datafile '+DATA/tos/datafile/users.276.754906035' format '/u01/oradata/tos/USERS01.dbf';

Starting backup at 27-JUN-11

using target database control file instead of recovery catalog

allocated channel: ORA_DISK_1

channel ORA_DISK_1: sid=141 devtype=DISK

channel ORA_DISK_1: starting datafile copy

input datafile fno=00004 name=+DATA/tos/datafile/users.276.754906035

output filename=/u01/oradata/tos/USERS01.dbf tag=TAG20110627T124853 recid=17 stamp=754922939

channel ORA_DISK_1: datafile copy complete, elapsed time: 00:00:07

Finished backup at 27-JUN-11

RMAN> backup as copy datafile '+DATA/tos/datafile/sysaux.271.754905929' format '/u01/oradata/tos/SYSAUX01.dbf';

Starting backup at 27-JUN-11

using channel ORA_DISK_1

channel ORA_DISK_1: starting datafile copy

input datafile fno=00003 name=+DATA/tos/datafile/sysaux.271.754905929

output filename=/u01/oradata/tos/SYSAUX01.dbf tag=TAG20110627T124929 recid=18 stamp=754923029

channel ORA_DISK_1: datafile copy complete, elapsed time: 00:01:05

Finished backup at 27-JUN-11

RMAN> backup as copy datafile '+DATA/tos/datafile/undotbs1.273.754906021' format '/u01/oradata/tos/UNDOTBS101.dbf';

Starting backup at 27-JUN-11

using channel ORA_DISK_1

channel ORA_DISK_1: starting datafile copy

input datafile fno=00002 name=+DATA/tos/datafile/undotbs1.273.754906021

output filename=/u01/oradata/tos/UNDOTBS101.dbf tag=TAG20110627T125049 recid=19 stamp=754923057

channel ORA_DISK_1: datafile copy complete, elapsed time: 00:00:15

Finished backup at 27-JUN-11

RMAN> backup as copy datafile '+DATA/tos/datafile/system.270.754905833' format '/u01/oradata/tos/SYSTEM01.dbf';

Starting backup at 27-JUN-11

using channel ORA_DISK_1

channel ORA_DISK_1: starting datafile copy

input datafile fno=00001 name=+DATA/tos/datafile/system.270.754905833

output filename=/u01/oradata/tos/SYSTEM01.dbf tag=TAG20110627T125112 recid=20 stamp=754923150

channel ORA_DISK_1: datafile copy complete, elapsed time: 00:01:25

channel ORA_DISK_1: starting datafile copy

copying current control file

RMAN-03009: failure of backup command on ORA_DISK_1 channel at 06/27/2011 12:52:39

ORA-01580: error creating control backup file /u01/oradata/tos/SYSTEM01.dbf

ORA-27038: created file already exists

Additional information: 1

continuing other job steps, job failed will not be re-run

channel ORA_DISK_1: starting full datafile backupset

channel ORA_DISK_1: specifying datafile(s) in backupset

including current SPFILE in backupset

channel ORA_DISK_1: starting piece 1 at 27-JUN-11

RMAN-00571: ===

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

RMAN-00571: ===

RMAN-03009: failure of backup command on ORA_DISK_1 channel at 06/27/2011 12:52:42

ORA-19504: failed to create file "/u01/oradata/tos/SYSTEM01.dbf"

ORA-27038: created file already exists

Additional information: 1

注：因为默认情况下，备份 system 数据文件是，会自动备份控制文件，这里因为 system01.dbf 已经备份好，而控制文件再次备份为该名称所以失败

RMAN> backup as copy datafile '+DATA/tos/datafile/example.272.754905995' format '/u01/oradata/tos/EXAMPLE01.dbf';

Starting backup at 27-JUN-11

using channel ORA_DISK_1

channel ORA_DISK_1: starting datafile copy

input datafile fno=00005 name=+DATA/tos/datafile/example.272.754905995

output filename=/u01/oradata/tos/EXAMPLE01.dbf tag=TAG20110627T125341 recid=21 stamp=754923244

channel ORA_DISK_1: datafile copy complete, elapsed time: 00:00:25

Finished backup at 27-JUN-11

RMAN> backup as copy datafile '+DATA/tos/datafile/xff.274.754906027' format '/u01/oradata/tos/XFF01.dbf';

Starting backup at 27-JUN-11

using channel ORA_DISK_1

channel ORA_DISK_1: starting datafile copy

input datafile fno=00006 name=+DATA/tos/datafile/xff.274.754906027

output filename=/u01/oradata/tos/XFF01.dbf tag=TAG20110627T125415 recid=22 stamp=754923257

channel ORA_DISK_1: datafile copy complete, elapsed time: 00:00:03

Finished backup at 27-JUN-11

RMAN> backup as copy datafile '+DATA/tos/datafile/xff.275.754906031' format '/u01/oradata/tos/XFF02.dbf';

Starting backup at 27-JUN-11

using channel ORA_DISK_1

channel ORA_DISK_1: starting datafile copy

input datafile fno=00007 name=+DATA/tos/datafile/xff.275.754906031

output filename=/u01/oradata/tos/XFF02.dbf tag=TAG20110627T125507 recid=23 stamp=754923309

channel ORA_DISK_1: datafile copy complete, elapsed time: 00:00:04

Finished backup at 27-JUN-11

RMAN> exit

Recovery Manager complete.

--登录 sqlplus

[oracle@localhost tmp]$ sqlplus / as sysdba

SQL*Plus: Release 10.2.0.3.0 - Production on Mon Jun 27 12:55:29 2011

Connected to:
Oracle Database 10g Enterprise Edition Release 10.2.0.3.0 - Production
With the Partitioning, OLAP and Data Mining options

--备份控制文件
SQL> alter database backup controlfile to '/tmp/control.ctl';

Database altered.

--关闭数据库
SQL> shutdown immediate;
Database closed.
Database dismounted.
ORACLE instance shut down.

--启动数据库只 nomount 状态
SQL> startup pfile='/tmp/pfile' nomount;
ORACLE instance started.

Total System Global Area 167772160 bytes
Fixed Size 1260672 bytes
Variable Size 79692672 bytes
Database Buffers 79691776 bytes
Redo Buffers 7127040 bytes
SQL> exit
Disconnected from Oracle Database 10g Enterprise Edition Release 10.2.0.3.0 - Production
With the Partitioning, OLAP and Data Mining options

[oracle@localhost tmp]$ rman target /

Recovery Manager: Release 10.2.0.3.0 - Production on Mon Jun 27 12:58:22 2011

connected to target database: tos (not mounted)

--恢复控制文件
RMAN> restore controlfile from '/tmp/control.ctl';

Starting restore at 27-JUN-11
using target database control file instead of recovery catalog
allocated channel: ORA_DISK_1
channel ORA_DISK_1: sid=156 devtype=DISK

channel ORA_DISK_1: copied control file copy
output filename=/u01/oradata/tos/control01.ctl
output filename=/u01/oradata/tos/control02.ctl
Finished restore at 27-JUN-11

--启动数据库只 mount 状态
RMAN> alter database mount;

database mounted
released channel: ORA_DISK_1

--修改数据文件在控制文件中位置
RMAN> switch tablespace SYSTEM to copy;

Starting implicit crosscheck backup at 27-JUN-11
allocated channel: ORA_DISK_1
channel ORA_DISK_1: sid=156 devtype=DISK
Finished implicit crosscheck backup at 27-JUN-11

Starting implicit crosscheck copy at 27-JUN-11

using channel ORA_DISK_1
Crosschecked 15 objects
Finished implicit crosscheck copy at 27-JUN-11

searching for all files in the recovery area
cataloging files...
no files cataloged

datafile 1 switched to datafile copy "/u01/oradata/tos/SYSTEM01.dbf"

RMAN> switch tablespace UNDOTBS1 to copy;

datafile 2 switched to datafile copy "/u01/oradata/tos/UNDOTBS101.dbf"

RMAN> switch tablespace SYSAUX to copy;

datafile 3 switched to datafile copy "/u01/oradata/tos/SYSAUX01.dbf"

RMAN> switch tablespace USERS to copy;

datafile 4 switched to datafile copy "/u01/oradata/tos/USERS01.dbf"

RMAN> switch tablespace EXAMPLE to copy;

datafile 5 switched to datafile copy "/u01/oradata/tos/EXAMPLE01.dbf"

RMAN> switch tablespace XFF to copy;

datafile 6 switched to datafile copy "/u01/oradata/tos/XFF01.dbf"
datafile 7 switched to datafile copy "/u01/oradata/tos/XFF02.dbf"

--恢复数据库
RMAN> recover database;

Starting recover at 27-JUN-11

using channel ORA_DISK_1

starting media recovery

archive log thread 1 sequence 9 is already on disk as file +DATA/tos/onlinelog/group_6.279.754906321

archive log filename=+DATA/tos/onlinelog/group_6.279.754906321 thread=1 sequence=9

media recovery complete, elapsed time: 00:00:03

Finished recover at 27-JUN-11

--打开数据库

RMAN> alter database open;

RMAN-00571: ===

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

RMAN-00571: ===

RMAN-03002: failure of alter db command at 06/27/2011 13:00:36

ORA-01589: must use RESETLOGS or NORESETLOGS option for database open

RMAN> alter database open resetlogs;

database opened

注：不能直接使用 open 打开

RMAN> exit

Recovery Manager complete.

[oracle@localhost tmp]$ sqlplus / as sysdba

SQL*Plus: Release 10.2.0.3.0 - Production on Mon Jun 27 13:02:53 2011

Connected to:
Oracle Database 10g Enterprise Edition Release 10.2.0.3.0 - Production
With the Partitioning, OLAP and Data Mining options

--增加 redo log
SQL> alter database add logfile group 1 '/u01/oradata/tos/redo01.log' size 10m;

Database altered.

SQL> alter database add logfile group 2 '/u01/oradata/tos/redo02.log' size 10m;

Database altered.

SQL> alter database add logfile group 3 '/u01/oradata/tos/redo03.log' size 10m;

Database altered.

--切换日志
SQL> alter system switch logfile;

System altered.

SQL> /

System altered.

SQL> /

System altered.

SQL> /

System altered.

--内存中数据写入硬盘
SQL> alter system checkpoint;

System altered.

--查询当前日志组状态
SQL> select group#,status from v$log;

GROUP# STATUS
---------- ----------
1 CURRENT
2 INACTIVE
3 INACTIVE
4 INACTIVE
5 INACTIVE
6 INACTIVE

6 rows selected.

--删除 asm 中日志
SQL> alter database drop logfile group 4;

Database altered.

SQL> alter database drop logfile group 5;

Database altered.

SQL> alter database drop logfile group 6;

Database altered.

--添加临时文件

SQL> alter tablespace temp add tempfile '/u01/oradata/tos/temp01.dbf' size 30m autoextend on maxsize 1g;

Tablespace altered.

--查看临时表空间中临时文件

SQL> select name from v$tempfile;

NAME

/u01/oradata/tos/temp01.dbf

+DATA/tos/tempfile/temp.280.754906369

--删除 asm 中临时文件

SQL> alter tablespace temp drop tempfile '+DATA/tos/tempfile/temp.280.754906369';

Tablespace altered.

--查看迁移结果

SQL> set pagesize 100

SQL> select name from v$datafile

2 union

3 select member from v$logfile

4 union

5 select name from v$controlfile

6 union

7 select name from v$tempfile;

NAME

/u01/oradata/tos/EXAMPLE01.dbf

/u01/oradata/tos/SYSAUX01.dbf

/u01/oradata/tos/SYSTEM01.dbf

/u01/oradata/tos/UNDOTBS101.dbf

/u01/oradata/tos/USERS01.dbf

/u01/oradata/tos/XFF01.dbf

/u01/oradata/tos/XFF02.dbf

/u01/oradata/tos/control01.ctl

/u01/oradata/tos/control02.ctl

/u01/oradata/tos/redo01.log

/u01/oradata/tos/redo02.log

/u01/oradata/tos/redo03.log

/u01/oradata/tos/temp01.dbf

13 rows selected.

--创建 spfile 文件

SQL> create spfile from pfile='/tmp/pfile';

File created.

9.6. Transportable Tablespaces

```
--------------------------------------------------------------------------------
-------------------------------- source database ---------------------------------
--------------------------------------------------------------------------------

SQL> SHOW PARAMETER DB_CREATE_FILE_DEST
NAME                                 TYPE                 VALUE
------------------------------------ ----------- ------------------------------
db_create_file_dest     string       +DATA

SQL> CREATE TABLESPACE TTS_1;
SQL> CREATE TABLESPACE TTS_2;

SQL> CREATE TABLE EMP_COPY TABLESPACE TTS_1 AS SELECT * FROM EMP;
SQL> CREATE INDEX EMP_COPY_I ON EMP_COPY (EMPNO) TABLESPACE TTS_2;

SQL> EXECUTE DBMS_TTS.TRANSPORT_SET_CHECK ('TTS_1, TTS_2' TRUE);
SQL> SELECT * FROM TRANSPORT_SET_VIOLATIONS;

SSKYDB =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = host2)(PORT = 1521))
    (CONNECT_DATA =
```

```
      (SERVER = DEDICATED)
      (SERVICE_NAME = SSKYDB)
    )
  )

SQL> CREATE DATABASE LINK SSKYDB CONNECT TO SYSTEM IDENTIFIED BY MANAGER1 USING 'SSKYDB';

--该操作需要 source 和 target 都进行
SQL> CREATE DIRECORY TTS_DUMP AS '+DATA/';
SQL> CREATE DIRECTORY TTS_DUMP_LOG AS '/export/home/tts_log/';
SQL> CREATE DIRECTORY TTS_DATAFILE AS '+DATA/db1/datafile/';

SQL> GRANT READ, WRITE ON DIRECTORY TTS_DUMP TO SYSTEM;
SQL> GRANT READ, WRITE ON DIRECTORY TTS_DUMP_LOG TO SYSTEM;
SQL> GRANT READ, WRITE ON DIRECTORY TTS_DATAFILE TO SYSTEM;

SQL> ALTER TABLESPACE tts_1 READ ONLY;
SQL> ALTER TABLESPACE tts_2 READ ONLY;

[ora10g@host1]$ expdp system/manager1 directory=tts_dump dumpfile=tts1_db1.dmp logfile=tts_dump_log:tts.log
transport_tablespaces=tts_1,tts_2 transport_full_check=y

--transfer dump file to target
[ora10g@host1]$ sqlplus system/manager

SQL>
BEGIN
    DBMS_FILE_TRANSFER.PUT_FILE(SOURCE_DIRECTORY_OBJECT         => 'TTS_DUMP',
                                SOURCE_FILE_NAME                => 'tts1.db1.dmp',
                                DESTINATION_DIRECTORY_OBJECT => 'TTS_DUMP',
                                DESTINATION_FILE_NAME           => 'tts1.db1.dmp',
                                DESTINATION_DATABASE             => 'SSKYDB');
END;
/

SQL> SELECT FILE_NAME FROM DBA_DATA_FILES WHERE TABLESPACE_NAME LIKE 'TTS%';
FILE_NAME
-----------------------------------------------------
+DATA/sskydb/datafile/tts_1.294.590721319
+DATA/sskydb/datafile/tts_2.295.586721335

SQL>
BEGIN
    DBMS_FILE_TRANSFER.PUT_FILE(SOURCE_DIRECTORY_OBJECT         => 'TTS_DATAFILE',
                                SOURCE_FILE_NAME                => 'tts_1.294.570721319',
                                DESTINATION_DIRECTORY_OBJECT => 'TTS_DATAFILE',
```

```
                                    DESTINATION_FILE_NAME           => 'tts1_db1.dbf',
                                    DESTINATION_DATABASE            => 'SSKYDB');
END;
/

SQL>
BEGIN
    DBMS_FILE_TRANSFER.PUT_FILE(SOURCE_DIRECTORY_OBJECT         => 'TTS_DATAFILE',
                                    SOURCE_FILE_NAME                => 'tts_2.295.586721335',
                                    DESTINATION_DIRECTORY_OBJECT => 'TTS_DATAFILE',
                                    DESTINATION_FILE_NAME           => 'tts2_db1.dbf',
                                    DESTINATION_DATABASE            => 'SSKYDB');
END;
/

--------------------------------------------------------------------------------
-------------------------------- target database -------------------------------
--------------------------------------------------------------------------------

[ora10g@host2]$ impdp system/oracle parfile=imp.par

--imp.par
DIRECTORY=TTS_DUMP
DUMPFILE=TTS1_DB1.DMP
LOGFILE=TTS_DUMP_LOG:TTS1.LOG
TRANSPORT_DATAFILES='+DATA1/tts1_db1.dbf','+DATA1/tts2_db1.dbf'

SQL> ALTER TABLESPACE tts_1 READ WRITE;
SQL> ALTER TABLESPACE tts_2 READ WRITE;

SQL> SELECT NAME FROM V$DATAFILE;
SQL> SELECT COUNT(*) FROM EMP_COPY;
```

10. ASM 故障修复

AS

11. 参考文献

http://www.killdb.com/2013/01/07/oracle-asm-%E5%89%96%E6%9E%90%E7%B3%BB%E5%88%971.html

http://www.askmaclean.com/archives/asm-file-number-1-the-file-directory.html

http://www.itpub.net/thread-1597605-1-1.html

http://blog.itpub.net/321157/viewspace-721325/

http://www.xifenfei.com/5867.html

https://twiki.cern.ch/twiki/bin/view/PDBService/ASM_Internals

http://czmmiao.iteye.com/blog/1749971

http://asmsupportguy.blogspot.fr/2010/04/asm-metadata.html

Master Note for Automatic Storage Management (ASM) (Doc ID 1187723.1)