

CHAPTER 4

Diskgroups and Failure Groups



he primary component of ASM is the diskgroup, which is the highest-level data structure in ASM (see Figure 4-1). A diskgroup is essentially a container that holds a logical grouping of disks that are managed together as a unit. The diskgroup is comparable to a logical volume manager's (LVM) volume group. Once ASM has discovered the disks, these disks can be used to create a diskgroup.

ASM diskgroups differ from typical LVM volume groups in the following ways:

- An ASM filesystem layer is implicitly created within a diskgroup.
- ASM diskgroups have inherent automatic file-level striping and mirroring capabilities. A database file created within an ASM diskgroup is distributed equally across all online disks in the diskgroup, which provides an even input/output (I/O) load.

Diskgroup Management

ASM has three diskgroups types: external redundancy, normal redundancy, and ASM high redundancy. The diskgroup type, which is defined at diskgroup creation, determines the level of mirroring performed by ASM. A diskgroup of external type

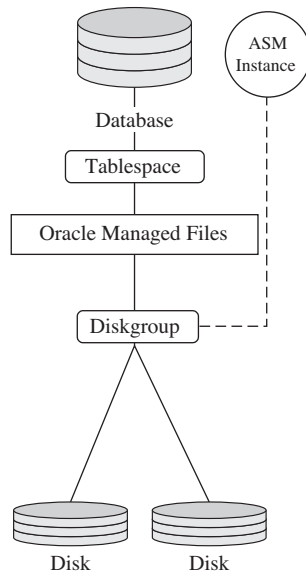


FIGURE 4-1. ASM layer

indicates that the mirroring will be handled and managed by the storage array and not by ASM. For example, a user may create an external type diskgroup where the storage array is an EMC DMX or Hitachi USP series. Because the core competency of these high-end arrays is mirroring, external redundancy is well suited for them.

With ASM redundancy, ASM performs and manages the mirroring. ASM redundancy is typically used over low-cost commodity storage, such as the Dell PowerVault MD1000 SAS storage array, or when deploying stretch clusters. For details on ASM redundancy, please see the “ASM Redundancy and Failure Groups” section later in this chapter. The next section focuses on creating diskgroups in an external redundancy environment.

Creating Diskgroups

The creation of a diskgroup involves validation of the disks to be added. The disks must have the following attributes:

- They cannot already be in use by another diskgroup.
- They must not have a preexisting valid ASM header. The `FORCE` option can be used to override this.
- They cannot have an Oracle file header. For example, from a raw Oracle datafile. The `FORCE` option can be used to override this for a raw Oracle datafile. Trying to create an ASM disk using a raw device datafile results in the following error:

```
SQL> CREATE DISKGROUP DATA DISK '/dev/sdd3';
create diskgroup data disk '/dev/sdd3'
ERROR at line 1:
ORA-15032: not all alterations performed
ORA-15201: disk /dev/sda3 contains a valid RDBMS file
```

The disk header validation prevents ASM from destroying any data device in use. Only disks with a header status of `CANDIDATE`, `FORMER`, or `PROVISIONED` are allowed to be diskgroup members. To add disks to a diskgroup with a header status of `MEMBER` or `FOREIGN`, use the `FORCE` option. To prevent gratuitous use of the `FORCE` option, ASM allows it only when using the `NOFORCE` option would fail. An attempt to use `FORCE` when it is not required results in an `ORA-15034` error (“disk ‘%s’ does not require the `FORCE` option”). Use the `FORCE` option with extreme caution, because it overwrites the data on the disk that was previously used as an ASM disk or database file.

A diskgroup can be created using SQL, Enterprise Manager (EM), or Database Configuration Assistant (DBCA) commands. In the following example, a `DATA` diskgroup is created using four disks that reside in a storage array, with the redundancy

92 Oracle Automatic Storage Management

being handled externally by the storage array. The following query lists the available that will be used in the diskgroup:

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

NAME	PATH	MODE_ST	STATE
	/dev/rds/c3t19d5s4	ONLINE	NORMAL
	/dev/rds/c3t19d16s4	ONLINE	NORMAL
	/dev/rds/c3t19d17s4	ONLINE	NORMAL
	/dev/rds/c3t19d18s4	ONLINE	NORMAL

```
SQL> CREATE DISKGROUP DATA EXTERNAL REDUNDANCY DISK
'/dev/rds/c3t19d5s4',
'/dev/rds/c3t19d16s4',
'/dev/rds/c3t19d17s4',
'/dev/rds/c3t19d18s4';
```

The output, from V\$ASM_DISKGROUP, shows the newly created diskgroup:

```
SQL> SELECT NAME, STATE, TYPE, TOTAL_MB, FREE_MB FROM V$ASM_DISKGROUP;
NAME                STATE          TYPE          TOTAL_MB    FREE_MB
-----
DATA                MOUNTED       EXTERN        34512       34101
```

After the diskgroup is successfully created, metadata information, which includes creation date, diskgroup name, and redundancy type, is stored in the System Global Area (SGA) and on each disk (in the disk header) within the diskgroup. The V\$ASM_DISK view reflects this disk header information. Once these disks are under ASM management, all subsequent mounts of the diskgroup reread and validate the ASM disk headers.

The following output shows how the V\$ASM_DISK view reflects the disk state change after the disk is incorporated into the diskgroup:

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

NAME	PATH	MODE_ST	STATE	DISK_NUMBER
DATA_0000	/dev/rds/c3t19d5s4	ONLINE	NORMAL	0
DATA_0001	/dev/rds/c3t19d16s4	ONLINE	NORMAL	1
DATA_0002	/dev/rds/c3t19d17s4	ONLINE	NORMAL	2
DATA_0003	/dev/rds/c3t19d18s4	ONLINE	NORMAL	3

The output that follows shows entries from the ASM alert log reflecting the creation of the diskgroup and the assignment of the disk names:

```
SQL> CREATE DISKGROUP DATA EXTERNAL REDUNDANCY DISK
'/dev/rds/c3t19d5s4', '/dev/rds/c3t19d16s4', '/dev/rds/c3t19d17s4',
'/dev/rds/c3t19d18s4'
NOTE: Assigning number (1,0) to disk (/dev/rds/c3t19d5s4)
```

```

NOTE: Assigning number (1,1) to disk (/dev/rdsd/c3t19d16s4)
NOTE: Assigning number (1,2) to disk (/dev/rdsd/c3t19d17s4)
NOTE: Assigning number (1,3) to disk (/dev/rdsd/c3t19d18s4)
NOTE: initializing header on grp 1 disk DATA_0000
NOTE: initializing header on grp 1 disk DATA_0002
NOTE: initializing header on grp 1 disk DATA_0003
NOTE: initializing header on grp 1 disk DATA_0004
NOTE: initiating PST update: grp = 1
Wed Mar 07 15:42:21 2007
NOTE: group DATA: initial PST location: disk 0000 (PST copy 0)
NOTE: group DATA: initial PST location: disk 0001 (PST copy 1)
NOTE: group DATA: initial PST location: disk 0001 (PST copy 3)
NOTE: group DATA: initial PST location: disk 0001 (PST copy 4)
NOTE: PST update grp = 1 completed successfully
NOTE: cache registered group DATA number=1 incarn=0xa311b052
NOTE: cache opening disk 0 of grp 1: DATA_0000 path: /dev/rdsd/c3t19d5s4
NOTE: cache opening disk 1 of grp 1: DATA_0001 path: /dev/rdsd/c3t19d16s4
NOTE: cache opening disk 2 of grp 1: DATA_0002 path: /dev/rdsd/c3t19d17s4
NOTE: cache opening disk 3 of grp 1: DATA_0003 path: /dev/rdsd/c3t19d18s4

```

When mounting diskgroups, either at ASM startup or for subsequent mounts, it is advisable to mount all required diskgroups at once. This minimizes the overhead of multiple ASM disk discovery scans.

ASM Disk Names

ASM disk names are usually assigned by default based on the diskgroup name and disk number, but names can be assigned during ASM diskgroup creation or when disks are added. The following example illustrates how to create a diskgroup where disk names are user-defined:

```

SQL> CREATE DISKGROUP DATA EXTERNAL REDUNDANCY DISK
'/dev/rdsd/c3t19d5s4' name DMX_disk1,
'/dev/rdsd/c3t19d16s4' name DMX_disk2,
'/dev/rdsd/c3t19d17s4' name DMX_disk3,
'/dev/rdsd/c3t19d18s4' name DMX_disk4;

```

If disk names are not provided, then ASM dynamically assigns a disk name with a sequence number to each disk added to the diskgroup:

```

SQL> CREATE DISKGROUP DATA EXTERNAL REDUNDANCY DISK
'/dev/rdsd/c3t19d5s4',
'/dev/rdsd/c3t19d16s4',
'/dev/rdsd/c3t19d17s4',
'/dev/rdsd/c3t19d18s4';

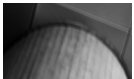
```

The ASM disk name is used when performing any disk management activities, such as `DROP DISK` or `RESIZE DISK`.

The ASM disk name is different from the small computer system interface (SCSI) address, and this allows for consistent naming across Real Application Clusters (RAC) nodes. Additionally, ASM disk names persist even if the SCSI address name changes. SCSI address name changes occur due to array reconfigurations and or after reboots.

Diskgroup Numbers

The lowest nonzero available diskgroup number is assigned on the first mount of a diskgroup. However, in an ASM cluster, even if the diskgroups are mounted in a different order between cluster nodes, the diskgroup numbers will still be consistent across the cluster but the diskgroup name never changes. For example, if node 1 has dgA as group number 1 and dgB and group number 2, then if node 2 mounts only dgB, then it will be group number 2, even though 1 is not in use in node 2.



NOTE

Diskgroup numbers are never recorded persistently, thus there is no diskgroup number in a disk header. Only the diskgroup name is recorded in the disk header.

There is a hard-coded limitation of 63 diskgroups in a cluster. There is a six-bit field that holds the diskgroup number and zero is reserved. This is independent of the data in their header. A diskgroup may be mounted on some nodes but not others.

Disk Numbers

Although diskgroup numbers are never recorded persistently, disk numbers are recorded on the disk headers. However, there is no persistent binding of disk numbers to pathnames. When an ASM instance starts up, it discovers all the devices matching the pattern in `ASM_DISKSTRING` and for which it has read/write access. If it sees an ASM disk header, then it knows the ASM disk number.

Also, disks that are discovered but are not part of any mounted diskgroup are reported in diskgroup 0. A disk that is not part of any diskgroup, mounted or not, will be in diskgroup 0 until it is added to a diskgroup or mounted. When the disk is added to a diskgroup, it will be associated with the correct diskgroup.

ASM Redundancy and Failure Groups

For systems that do not use external redundancy, ASM provides its own internal redundancy mechanism and additional high availability. A diskgroup is divided into failure groups, and each disk is in exactly one failure group. A failure group is a

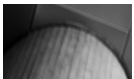
collection of disks that can become unavailable due to failure of one of its associated components. Possible failing components could be any of the following:

- Storage array controllers
- Host bus adapters (HBAs)
- Fibre Channel (FC) switches
- Disks
- Entire arrays

Thus disks in two separate failure groups (for a given diskgroup) must not share a common failure component. If you define failure groups for your diskgroup, ASM can tolerate the simultaneous failure of multiple disks in a single failure group.

ASM uses a unique mirroring algorithm. ASM does not mirror disks; rather, it mirrors extents. When ASM allocates a primary extent of a file to one disk in a failure group, it allocates a mirror copy of that extent to another disk in another failure group. Thus, ASM ensures that a primary extent and its mirror copy never reside in the same failure group.

Unlike other volume managers, ASM has no concept of a primary disk or a mirrored disk. As a result, to provide continued protection in event of failure, your diskgroup requires only spare capacity; a hot spare disk is unnecessary. Redundancy for diskgroups can be either *normal* (the default), where files are two-way mirrored (requiring at least two failure groups), or *high*, which provides a higher degree of protection using three-way mirroring (requiring at least three failure groups). After you create a diskgroup, you cannot change its redundancy level. If you want a different redundancy, then you must create another diskgroup with the desired redundancy, then move the datafiles (using Recovery Manager [RMAN] restore, ASMCMD copy command or `DBMS_FILE_TRANSFER`) from the original diskgroup to the newly created diskgroup.

**NOTE**

Diskgroup metadata are always triple mirrored with normal or high redundancy

Additionally, after you assign a disk to a failure group, you cannot reassign it to another failure group. If you want to move it to another failure group, then you must first drop it from the current failure group and then add it to the desired failure group. However, because the hardware configuration usually dictates the choice of a failure group, users generally do not need to reassign a disk to another failure group unless it is physically moved.

Creating ASM Redundancy Diskgroups

The following simple example shows how to create a normal redundancy disk group using two failure groups over a NetApp filer:

```
SQL> CREATE DISKGROUP DATA_NRM1 NORMAL REDUNDANCY
FAILGROUP F1GRP1 DISK '/dev/rdisk/c3t19d3s4', '/dev/rdisk/c3t19d4s4',
'/dev/rdisk/c3t19d5s4', '/dev/rdisk/c3t19d6s4', '/dev/rdisk/c4t20d3s4',
'/dev/rdisk/c4t20d4s4', '/dev/rdisk/c4t20d5s4', '/dev/rdisk/c4t20d6s4'
FAILGROUP FLGRP2 DISK '/dev/rdisk/c5t21d3s4', '/dev/rdisk/c5t21d4s4',
'/dev/rdisk/c5t21d5s4', '/dev/rdisk/c5t21d6s4', '/dev/rdisk/c6t22d3s4',
'/dev/rdisk/c6t22d4s4', '/dev/rdisk/c6t22d5s4', '/dev/rdisk/c6t22d6s4';
```

The same create diskgroup command can be executed using wildcard syntax:

```
SQL> CREATE DISKGROUP DATA_NRM1 NORMAL REDUNDANCY
FAILGROUP FL1GRP1 DISK '/dev/rdisk/c[34]*'
FAILGROUP FLGRP2 DISK '/dev/rdisk/c[56]*';
```

In this following example, ASM normal redundancy is being deployed over a low-cost commodity storage array. This storage array has four internal trays, with each tray having four disks. Because the failing component to isolate is the storage tray, the failure group boundary is set for the storage tray—that is, each storage tray is associated with a failure group:

```
SQL> CREATE DISKGROUP DATA_NRM1 NORMAL REDUNDANCY
FAILGROUP FLGRP1 DISK '/dev/rdisk/c3t19d3s4', '/dev/rdisk/c3t19d4s4',
'/dev/rdisk/c3t19d5s4', '/dev/rdisk/c3t19d6s4'
FAILGROUP FLGRP2 DISK '/dev/rdisk/c4t20d3s4', '/dev/rdisk/c4t20d4s4',
'/dev/rdisk/c4t20d5s4', '/dev/rdisk/c4t20d6s4'
FAILGROUP FLGRP3 DISK '/dev/rdisk/c5t21d3s4', '/dev/rdisk/c5t21d4s4',
'/dev/rdisk/c5t21d5s4', '/dev/rdisk/c5t21d6s4'
FAILGROUP FLGRP4 DISK '/dev/rdisk/c6t22d3s4', '/dev/rdisk/c6t22d4s4',
'/dev/rdisk/c6t22d5s4', '/dev/rdisk/c6t22d6s4';
```

Designing for ASM Redundancy Diskgroups

Note that with ASM redundancy, you are not restricted to having two failure groups for normal redundancy and three for high redundancy. In the preceding example, four failure groups are created to ensure that disk partners are not allocated from the same storage tray.

There may be cases where users want to protect against storage area network (SAN) array failures. This can be accomplished by putting each array in a separate failure group. For example, a configuration may include two NetApp filers and the deployment of ASM normal redundancy such that each filer—that is, all logical unit numbers (LUNs) presented through the filer—are part of an ASM failure group. In this scenario, ASM mirrors extent between the two filers.

If the database administrator (DBA) does not specify a failure group in the `CREATE DISKGROUP` command, then a failure group is automatically constructed for each disk. This method of placing every disk in its own failure group works well for most customers.

The choice of failure groups depends on the kinds of failures that need to be tolerated without loss of data availability. For small numbers of disks (for example, fewer than 20), it is usually best to put every disk in its own failure group. Nonetheless, this is also beneficial for large numbers of disks when the main concern is spindle failure. To protect against the simultaneous loss of multiple disk drives due to a single component failure, explicit failure group specification should be used. For example, a diskgroup may be constructed from several small modular disk arrays. If the system needs to continue operation when an entire modular array fails, then each failure group should consist of all the disks in one module. If one module fails, all the data on that module are relocated to other modules to restore redundancy. Disks should be placed in the same failure group if they depend on a common piece of hardware whose failure needs to be tolerated with no loss of availability.

It is much better to have several failure groups as long as the data is still protected against the necessary component failures. A small number of failure groups or failure groups of uneven capacity can lead to allocation problems that prevent full utilization of all available storage. Moreover, having failure groups of different sizes can waste disk space. There may be enough room to allocate primary extents, but no space available for secondary extents. For example, in a diskgroup with six disks and three failure groups, if two disks are in their own individual failure groups and the other four are in one common failure group, then there will be very unequal allocation. All the secondary extents from the big failure group can be placed on only two of the six disks. The disks in the individual failure groups fill up with secondary extents and block additional allocation even though plenty of space is left in the large failure group. This also places an uneven write load on the two disks that are full because they contain more secondary extents that are accessed only for writes or if the disk with the primary extent fails.

Allocating ASM Extent Sets

With ASM redundancy, the first file extent allocated is chosen as primary extent, and the mirrored extent is called the secondary extent. In the case of high redundancy, there will be two secondary extents. This logical grouping of primary and secondary extents is called an *extent set*. When a block is read from disk, it is always read from the primary extent, unless the primary extent cannot be read. In Oracle Database 11g, the preferred read feature allows the database to read the secondary extent first instead of reading the primary extent. This is especially important for RAC Extended Cluster implementations. See the section “ASM and Extended Clusters” later in this chapter for more details on this feature.

Keep in mind that each disk in a diskgroup (and in failure groups) contains nearly the same number of primary and secondary extents. This provides an even distribution of read I/O activity across all the disks.

All the extents in an extent set always contain the exact same data because they are mirrored versions of each other. The only exception is when temporary tablespaces are used. For example, if the relational database management system (RDBMS) instance crashes while writing to a temporary tablespace, there is no need to resilver the mirrors because they will not be read again before they are written.

When a block is to be written to a file, each extent in the extent set is written in parallel. ASM requires that all writes complete before acknowledging the write to the client. Otherwise, the unwritten side could be read before it is written. If one write I/O fails, then that side of the mirror must be made unavailable for reads before the write can be acknowledged.

Disk Partnering

In ASM redundancy diskgroups, ASM protects against a double-disk failure, (which can lead to data loss) by mirroring copies of data on disks that are partners of the disk containing the primary data extent. A *disk partnership* is a symmetric relationship between two disks in a high or normal redundancy diskgroup, and ASM automatically creates and maintains these relationships. ASM selects partners for a disk from failure groups other than the failure group to which the disk belongs. This ensures that a disk with a copy of the lost disk's data will be available following the failure of the shared resource associated with the failure group. ASM limits the number of disk partners to 10 for any single disk.

To illustrate disk partnership, the following example uses the normal redundancy diskgroup DATA_NRML, created previously. The following query shows a disk partnership for a disk (disk 2) that is part of failure group FLGRP1:

```
SQL> SELECT NUMBER_KFDPARTNER, FAILGROUP FROM X$KFDPARTNER A, V$ASM_DISK
A WHERE DISK=2 AND GRP=1 AND A.NUMBER_KFDPARTNER= B.DISK_NUMBER
NUMBER_KFDPARTNER FAILGROUP
-----
          7 FLGRP2
          8 FLGRP2
         10 FLGRP2
         12 FLGRP2
         16 FLGRP3
         17 FLGRP3
         19 FLGRP3
         24 FLGRP4
         25 FLGRP4
         26 FLGRP4
```

Notice that ASM did not choose partner disks from its own failure group (FLGRP1); rather, 10 partners were chosen from the other three failure groups. Disk partnership is detailed in Chapter 11, "ASM Operations."

Recovering Failure Groups

Returning to the example in the previous `CREATE DISKGROUP DATA_NRM1` command, in the event of a disk failure in failure group `FLGRP1`, which will induce a rebalance, the contents (data extents) of the failed disk are reconstructed using the redundant copies of the extents from partner disks. These partner disks are from failure group `FLGRP2`, `FLGRP3`, or both. If the database instance needs to access an extent whose primary extent was on the failed disk, then the database will read the mirror copy from the appropriate disk. After the rebalance is complete and the disk contents are fully reconstructed, the database instance returns to reading primary copies only.

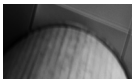
ASM and Extended Clusters

The last two years have seen the emergence of extended clusters. An extended cluster, also called a *stretch cluster*, *geocluster*, *campus cluster*, or *metro-cluster*, is essentially a RAC environment deployed across two data center locations. Many customers implement extended RAC to marry disaster recovery with the benefits of RAC, all in an effort to provide higher availability.

Within Oracle, the term *extended clusters* is used to refer to all of the stretch cluster implementations.

The distance for extended RAC can be anywhere between several meters to several hundred kilometers. Because Cluster Ready Services (CRS)-RAC cluster group membership is based on the ability to communicate effectively across the interconnect, extended cluster deployment requires a low-latency network infrastructure. For close proximity, users typically use Fibre Channel, whereas for large distances, Dark Fiber is used.

For normal redundancy diskgroups in extended RAC, there should be only one failure group on each site of the extended cluster. High-redundancy diskgroups should not be used in extended cluster configurations unless there are three sites. In this scenario, there should one failure group at each site. Note that it is best to name the failure groups explicitly based on the site name.



NOTE

If a diskgroup contains an asymmetrical configuration, such that there are more failure groups on one site than another, then an extent could get mirrored to the same site and not to the remote failure group. This could cause the loss of access to the entire diskgroup if the site containing more than one failure group fails.

As stated earlier, ASM in Oracle Database 10g always reads the primary copy of a mirrored extent set. Thus, a read for a specific block may require a read of the primary extent at the remote site across the interconnect. Accessing a remote disk through a metropolitan area or wide area storage network is substantially slower than accessing a local disk. This can tax the interconnect as well as result in high I/O and network latency.

To assuage this, Oracle Database 11g provides a feature called *preferred reads*. This feature enables ASM administrators to specify a failure group for local reads—that is, provide preferred reads. In a normal or high-redundancy diskgroup, when a secondary extent is on a preferred disk and the primary extent is remote, the secondary extent is read rather than the primary one on that node. This feature is especially beneficial for extended cluster configurations.

ASM Preferred Read

The `ASM_PREFERRED_READ_FAILURE_GROUP` initialization parameter is used to specify a list of failure group names that will provide local reads for each node in a cluster. The format of the `ASM_PREFERRED_READ_FAILURE_GROUP` is as follows:

```
ASM_PREFERRED_READ_FAILURE_GROUPS = DISKGROUP_NAME.FAILUREGROUP_NAME, . . .
```

Each entry is comprised of `DISKGROUP_NAME`, which is the name of the diskgroup, and `FAILUREGROUP_NAME`, which is the name of the failure group within that diskgroup, with a period separating these two variables. Multiple entries can be specified using commas as a separator. This parameter can be dynamically changed.

In an extended cluster, the failure groups that you specify with settings for the `ASM_PREFERRED_READ_FAILURE_GROUPS` parameter should contain only disks that are local to the instance. `V$ASM_DISK` indicates the preferred disks with a `Y` in the `PREFERRED_READ` column.

Note that when adding or dropping a disk, it is a best practice to issue the `add` or `drop` command from the site where storage change is occurring. This enables a more efficient rebalance method because the extent relocation is localized within the same failure group—that is, within the same local site.

The following example shows how to deploy the preferred read feature and demonstrates some of its inherent benefits. This example illustrates I/O patterns when the `PREFERRED_READ_FAILURE_GROUPS` parameter is not set, and then demonstrates how changing the parameter affects I/O.

1. First create a diskgroup with two failure groups:

```
CREATE DISKGROUP MYDATA NORMAL REDUNDANCY
-- these disks are local access from node1/remote from node2
FAILGROUP FG1 DISK '/dev/sda1', '/dev/sdb1', '/dev/sdc1'
-- these disks are local access from node2/remote from node 1
FAILGROUP FG2 DISK '/dev/sdf1', '/dev/sdg1', '/dev/sdh1';
```

Oracle Enterprise Manager (SYS) - Disk Group I/O Cumulative Statistics - Microsoft Internet Explorer

Automatic Storage Management: +ASM1_mangrid1.uk.oracle.com > Logged in As SYS / SYSDBA

Disk Group I/O Cumulative Statistics

Data Retrieved March 8, 2007 10:34:52 AM GMT Refresh Real Time: Manual Refresh Refresh

Expand All | Collapse All

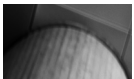
Disk Groups	Average Response Time (ms)	Average Throughput (MB per second)	Total I/O Calls	Reads		Writes	
				Total	Errors	Total	Errors
Automatic Storage Management							
-							
+ASM1_mangrid1.uk.oracle.com	2.53	3.53	10375	8083	0	2292	0
DATA	n/a	n/a	n/a	n/a	n/a	n/a	n/a
MYDATA	2.53	3.53	10375	8083	0	2292	0
MYDATA_0000	2.45	3.28	1686	1187	0	499	0
MYDATA_0001	3.03	3.04	1660	1341	0	319	0
MYDATA_0002	2	4.88	2004	1444	0	560	0
MYDATA_0003	2.52	3.11	1485	1142	0	343	0
MYDATA_0004	3.18	2.69	1443	1115	0	328	0
MYDATA_0005	2.27	4.24	2097	1854	0	243	0

FIGURE 4-2. The EM Disk Group I/O Cumulative Statistics screen

- In this test case, note that under the Reads subheading of the Total table column heading shown in Figure 4-2, and 4-3. The I/Os are evenly distributed across all disks—that is, these are nonlocalized I/Os.
- Note that you can achieve the same effect by entering the following query:

```
SQL> SELECT INST_ID, FAILGROUP, SUM(READS), SUM(WRITES) FROM GV$ASM_DISK
WHERE FAILGROUP IN ('FG1','FG2') GROUP BY INST_ID, FAILGROUP;
```

INST_ID	FAILGROUP	SUM(READS)	SUM(WRITES)
1	FG1	3796	2040
1	FG2	5538	2040
2	FG1	4205	1874
2	FG2	5480	1874



NOTE

V\$ASM_DISK includes I/Os that are performed by the ASM instance for Partnership Status Table (PST) heartbeats, discovery, and so on. The Oracle Database 11g *V\$ASM_DISK_IOSTAT* view was introduced to highlight preferred read. The *V\$ASM_DISK_IOSTAT* tracks I/O on a per-database basis. This view can be used to verify that an RDBMS instance never does any I/O to a nonpreferred disk.

Disk Groups	Average Response Time (ms)	Average Throughput (MB per second)	Total I/O Calls	Reads		Writes	
				Total	Errors	Total	Errors
Automatic Storage Management							
- +ASM1_mangrid1.uk.oracle.com	2.54	8.4	13585	7855	0	5730	0
DATA	n/a	n/a	n/a	n/a	n/a	n/a	n/a
MYDATA	2.54	8.4	13585	7855	0	5730	0
MYDATA_0000	2.54	7.79	3549	2663	0	886	0
MYDATA_0001	2.98	7.21	3533	2699	0	834	0
MYDATA_0002	1.19	21.03	1170	39	0	1131	0
MYDATA_0003	1.4	21.32	762	35	0	727	0
MYDATA_0004	3.19	6.1	3529	2384	0	1145	0
MYDATA_0005	1.16	18.73	1042	35	0	1007	0

FIGURE 4-3. The EM Disk Group I/O Cumulative Statistics screen

- Now set the appropriate ASM parameters for preferred read. Note that you need not dismount or remount the diskgroup, as this the parameter is dynamic.

Enter the following for Node1 (site1):

```
+ASM1.asm_preferred_read_failure_groups=MYDATA.FG1
```

Enter this code for Node2 (site2):

```
+ASM2.asm_preferred_read_failure_groups='MYDATA.FG2'
```

- Verify that the parameter took effect by querying GV\$ASM_DISK. From Node1, observe the following:

```
SELECT INST_ID, FAILGROUP, NAME, PREFERRED_READ FROM G$ASM_DISK
ORDER BY INST_ID, FAILGROUP;
```

INST_ID	FAILGROUP	NAME	PREFERRED_READ
1	FG1	MYDATA_0000	Y
1	FG1	MYDATA_0001	Y
1	FG1	MYDATA_0004	Y
1	FG2	MYDATA_0002	N
1	FG2	MYDATA_0003	N
1	FG2	MYDATA_0005	N

```

2 FG1          MYDATA_0000 N
2 FG1          MYDATA_0001 N
2 FG1          MYDATA_0004 N
2 FG2          MYDATA_0002 Y
2 FG2          MYDATA_0003 Y
2 FG2          MYDATA_0005 Y

```

Keep in mind that disks MYDATA_0000, MYDATA_0001, and MYDATA_0004 are part of the FG1 failure group, and disks MYDATA_0002, MYDATA_0003, MYDATA_0005 are in failure group FG2.

- Put a load on the system and check I/O calls via EM or using V\$ASM_DISK_IOSTAT. Notice in the “Reads-Total” column that reads have a strong affinity to the disks in FG1. This is because FG1 is local to node1 where +ASM1 is running. The remote disks in FG2 have very few reads.
- Using SQLPlus, the same behavior can be observed. Notice the small number of reads that instance 1 is making to FG2 and the small number of reads that instance 2 is making to FG1:

```
SQL> SELECT INST_ID, FAILGROUP, SUM(READS), SUM(WRITES) FROM GV$ASM_DISK
WHERE FAILGROUP IN ('FG1','FG2') GROUP BY INST_ID, FAILGROUP;
```

INST_ID	FAILGROUP	SUM(READS)	SUM(WRITES)
1	FG1	8513	3373
1	FG2	118	3373
2	FG1	72	1756
2	FG2	5731	1756

Recovering from Transient and Permanent Disk Failures

This section reviews how ASM handles transient and permanent disk failures in normal- and high-redundancy diskgroups. Additionally, this section describes the differences in processing of these failures between Oracle Database 10g and Oracle Database 11g.

Recovering from Disk Failures in Oracle Database 10g

In Oracle Database 10g, as described previously, restoring the redundancy of all extents in the diskgroup following a disk failure is a relatively costly operation. This can be especially expensive in the case of transient errors, which can include

cable disconnections, host bus adapter or controller failures, or even disk power interruptions, which cause the entire failure to be dropped. Additionally, a transient failure of the storage interconnect between sites would appear as a failure of an entire failure group. When an ASM disk fails, the ASM disk is taken offline and dropped.

If the ASM diskgroup is created with two failure groups, the loss of one entire failure group is tolerated with no downtime. Messages in the alert log and EM describe the situation. The diskgroup continues to operate normally, allocating space in the one surviving failure group. The failed disks show up in `V$ASM_DISK` twice (as of 10.1.0.4): once as *missing* and *candidate* statuses for `HEADER_STATUS` and `STATE`, respectively, and another as *offline hung*. The reason that you have missing, candidate, and offline hung disks is that you have had a failure group crash, and an insufficient number of surviving failure groups exists to complete a rebalance successfully to restore the contents from the failed disks. The offline hung entries in `V$ASM_DISK` essentially track the fact that there are extents whose redundancy has not yet been restored.

To restore the disks to a normal status, you need to add back the missing disks. To add the disks back, you must specify the `FORCE` flag, as in the following example, because they still have their old disk headers:

```
ALTER DISKGROUP DATA_NRM1 ADD FAILGROUP FLGRP1 DISK
'/dev/rdisk/c3t19d3s4' FORCE, '/dev/rdisk/c3t19d4s4' FORCE,
'/dev/rdisk/c3t19d5s4' FORCE, '/dev/rdisk/c3t19d6s4' FORCE;
```

When adding back the disks, be sure to specify the previously used failure group name. Adding these disks initiates a rebalance. Once the rebalance completes from adding back the disks from the failed failure group, the offline hung entries should go away.

In cases where it is appropriate, you can also use a disk add pattern such as `/dev/rdisk/c*`; just make sure that you have the right pattern for the disks you are adding and the appropriate failure group name.

The following are some points to consider when adding back disks into the failgroup:

- Make sure to add the disk back using the same failure group as before the storage failure.
- If upon disk failure the disk to be added back to the failgroup has been physically replaced, then the ASM disk header no longer exists. In this case, the disk cannot be added back to the failgroup using the `FORCE` option; it must be added using the standard add disk command:

```
ALTER DISKGROUP DATA_NRM1 ADD FAILGROUP FLGRP1 DISK '/dev/rdisk/c3t13d3s4';
```

- If you are providing a disk name when adding back to the failgroup, ensure that the disk name is different from the previous disk name. Currently, ASM does not support adding a disk back using the same disk name in the diskgroup. It is a best practice to let ASM generate a new disk name for you. If you are using ASMLIB, the default name is provided by the ASMLIB disk name stamped by `oracledasm`. In such cases, with Oracle Database 10g the user must explicitly specify a new disk name as part of the `add disk`.

When disks are added back to reform a second failure group, the rebalance restores redundancy and the hung disks are dropped. Note that adding back an insufficient number of disks can result in the diskgroup running out of space during rebalance. The added storage must be of sufficient capacity to hold a copy of all the data allocated in the surviving failure group. It is also possible to get in a similar state if all but a few disks in a failure group fail. For example, if there are 10 disks in each failure group and 9 of them fail in one failure group, the surviving disk is unlikely to have the capacity to hold a copy of every extent, so the rebalance runs out of space. The easy way to avoid this circumstance is to drop force the one surviving disk.

On some occasions, DBAs need to store data offline in a stretch cluster as part of a planned outage. This essentially drops all the disks in that failure group. However, keep in mind that the rebalance cannot move all of the storage from the dropped disks following a normal drop of all of the disks in a failure group in a two-failure-group diskgroup, because there is nowhere to move the data. Nevertheless, it is part of the design of ASM that any storage reconfiguration—that is, the adding or dropping of disks—always invokes a rebalance. The behavior in Oracle Database 10g is as described previously—that is, the disks in the failure group have hung, cached, or member status. Note that the drop disk operation is an asynchronous operation. The “Statement Processed” message indicates only that the drop has been initiated and that no new allocations will occur on that disk if it is in dropping state; the `V$ASM_DISK` view reveals whether or not a disk is still a member of the diskgroup. Note that a disk is not dropped until all of its contents have been moved to another disk. If such a disk has failed, it will be considered missing. The hung status is documented as a drop disk operation that cannot continue because there is insufficient space to relocate the data from the disk being dropped.

The hung state has been omitted from Oracle Database 11g. Disks that are effectively hung continue to be displayed as dropping or forcing.

Recovering from Disk Failures in Oracle Database 11g—Fast Disk Resync

The Oracle Database 11g feature ASM Fast Disk Resync significantly reduces the time to recover from transient disk failures in failure groups. The feature accomplishes this speedy recovery by quickly resynchronizing the failed disk with its partnered disks.

In Oracle ASM Database 10g, disks that go offline because of disk failures are immediately dropped from the diskgroup. To reconstruct a disk's contents, ASM must add the disk back to the diskgroup and perform a full rebalance. This problem can be exacerbated if all disks in a failure group go offline.

With Fast Disk Resync, the repair time is proportional to the number of extents that have been written or modified since the failure. This feature can significantly reduce the time that it takes to repair a failed diskgroup from hours to minutes.

The Fast Disk Resync feature allows the user a grace period to repair the failed disk and return it online. This time allotment is dictated by the ASM diskgroup attribute `DISK_REPAIR_TIME`. This attribute dictates maximum time of the disk outage that ASM can tolerate before dropping the disk. If the disk is repaired before this time is exceeded, then ASM resynchronizes the repaired disk when the user places the disk online. The command `ALTER DISKGROUP DISK ONLINE` is used to place the repaired disk online and initiate disk resynchronization.

Fast Disk Resync requires that the `COMPATIBLE.ASM` and `COMPATIBLE.RDBMS` attributes of the ASM diskgroup be set to at least Oracle 11.1.0.0. In the following example, the current ASM 11g diskgroup has a compatibility of 10.1.0.0.0 and is modified to 11.1. To validate the attribute change, the `V$ASM_ATTRIBUTE` view is queried.

```
SQL> SELECT NAME, COMPATIBILITY, DATABASE_COMPATIBILITY FROM
V$ASM_DISKGROUP_STAT;
NAME      COMPATIBILITY      DATABASE_COMPATIBILITY
-----
DATA      10.1.0.0.0        10.1.0.0.0

SQL> ALTER DISKGROUP DATA SET ATTRIBUTE 'COMPATIBLE.ASM' = '11.1';
SQL> ALTER DISKGROUP DATA SET ATTRIBUTE 'COMPATIBLE.RDBMS' = '11.1';

SQL> SELECT NAME, VALUE FROM V$ASM_ATTRIBUTE;
NAME      VALUE
-----
disk_repair_time 12960
compatible.asm  11.1.0.0.0
compatible.rdbms 11.1.0.0.0
```

After you correctly set the compatibility to Oracle Database version 11.1, you can set the `DISK_REPAIR_TIME` attribute accordingly. Notice that the default repair time is 12,960 sec or 3.6 hours. It is a best practice to leave this value at the default, as it should support most disk outages.

If the value of `DISK_REPAIR_TIME` needs to be changed, you can enter the following command:

```
ALTER DISKGROUP DATA SET ATTRIBUTE 'DISK_REPAIR TIME' = '4 H'
```

If the `DISK_REPAIR_TIME` parameter is not 0 and an ASM disk fails, that disk is taken offline but not dropped. During this outage, ASM tracks any modified

extents using a bitmap that is stored in diskgroup metadata. (See Chapter 11, “ASM Operations,” for more details on the algorithms used for resynchronization.)

ASM’s GMON process will periodically (every 3 min) inspect all mounted diskgroups for offline disks. If GMON finds any, it sends a message to a slave process to decrement their timer values (by 3 min) and initiate a drop for the offline disks when the timer expires. This timer display is shown in REPAIR_TIMER column of V\$ASM_DISK.

The ALTER DISK GROUP DISK OFFLINE SQL command or the EM ASM Target page can also be used to take the ASM disks offline manually for preventative maintenance. The following describes this scenario using SQLPlus:

```
SQL> ALTER DISKGROUP DATA OFFLINE DISK DATA_0000 DROP AFTER 20 m;
```

```
SQL> SELECT NAME, HEADER_STATUS, MOUNT_STATUS, MODE_STATUS, STATE,
REPAIR_TIMER FROM V$ASM_DISK WHERE GROUP_NUMBER=1;
```

NAME	HEADER_STATU	MOUNT_S	MODE_ST	STATE	REPAIR_TIMER
DATA_0003	MEMBER	CACHED	ONLINE	NORMAL	0
DATA_0002	MEMBER	CACHED	ONLINE	NORMAL	0
DATA_0001	MEMBER	CACHED	ONLINE	NORMAL	0
DATA_0000	UNKNOWN	MISSING	OFFLINE	NORMAL	840

Notice that the offline disk’s MOUNT_STATUS and MODE_STATUS are set to the MISSING and OFFLINE states, and also that the REPAIR_TIMER begins to decrement from the drop timer.

Figure 4-4 shows the EM method for taking disks offline. Figures 4-5 and 4-6 show EM screens confirming the offline operation.

After the maintenance is completed, you can use the ALTER DISK GROUP DISK ONLINE command to bring the disk(s) online:

```
SQL> ALTER DISKGROUP DATA ONLINE DISK DATA_0000
```

Or

```
SQL> ALTER DISKGROUP DATA ONLINE ALL
```

This statement brings all the repaired disks back online to bring the stale contents up to date and to enable new contents. See Chapter 11, “ASM Operations,” for more details on how to implement resynchronization.

The following is an excerpt from the ASM alert log showing a disk being brought offline and online:

```
SQL> ALTER DISKGROUP DATA OFFLINE DISK DATA_0000
NOTE: DRTimer CodCreate: of disk group 2 disks 0
WARNING: initiating offline of disk 0.3915947593 (DATA_0000) with mask 0x7e
NOTE: initiating PST update: grp = 2, dsk = 0, mode = 0x15
NOTE: PST update grp = 2 completed successfully
NOTE: initiating PST update: grp = 2, dsk = 0, mode = 0x1
```

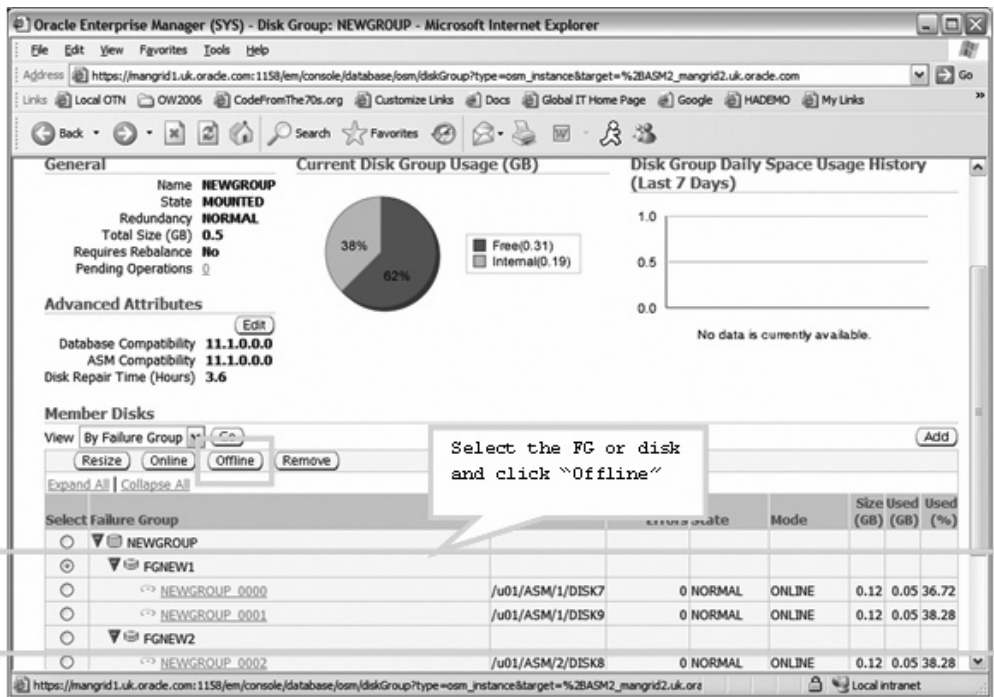


FIGURE 4-4. EM taking a disk offline for preventive maintenance

```
NOTE: PST update grp = 2 completed successfully
Tue Mar 20 08:15:37 2007
NOTE: cache closing disk 0 of grp 2: DATA_0000
Tue Mar 20 08:17:50 2007
GMON SlaveB: Deferred DG Ops completed.
Tue Mar 20 08:19:06 2007
.....
```

After fixing the disk, you can bring it online using the following command:

```
SQL> ALTER DISKGROUP DATA ONLINE DISK DATA_0000;
SQL> SELECT NAME, HEADER_STATUS, MOUNT_STATUS, MODE_STATUS, STATE,
REPAIR_TIMER FROM V$ASM_DISK WHERE GROUP_NUMBER=1;
```

NAME	HEADER_STATU	MOUNT_S	MODE_ST	STATE	REPAIR_TIMER
DATA_0003	MEMBER	CACHED	ONLINE	NORMAL	0
DATA_0002	MEMBER	CACHED	ONLINE	NORMAL	0
DATA_0001	MEMBER	CACHED	ONLINE	NORMAL	0
DATA_0000	MEMBER	CACHED	ONLINE	NORMAL	0

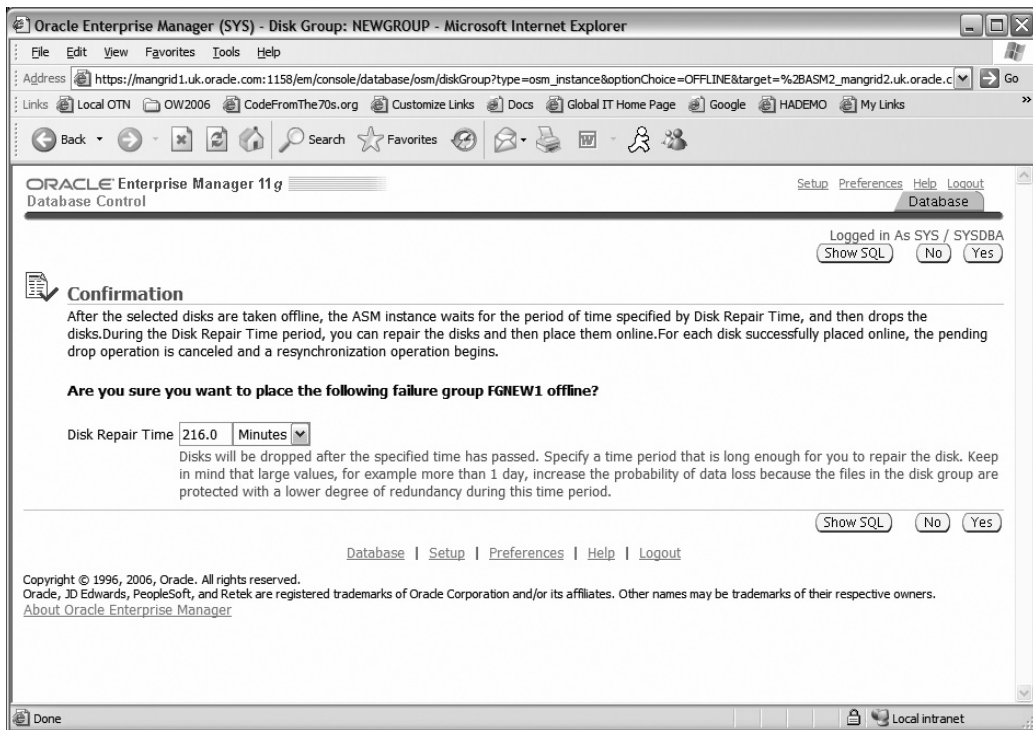


FIGURE 4-5. EM confirming that it has taken disks offline

```
SQL> ALTER DISKGROUP DATA ONLINE disk DATA_0000
Tue Mar 20 08:29:06 2007
NOTE: initiating online of disk group 2 disks 0
NOTE: initiating PST update: grp = 2, dsk = 0, mode = 0x19
NOTE: disk validation pending for group 2/0x62087046 (DATA)
NOTE: cache opening disk 0 of grp 2: DATA_0000 path:/u01/ASM/1/DISK7
SUCCESS: validated disks for 2/0x62087046 (DATA)
NOTE: initiating PST update: grp = 2, dsk = 0, mode = 0x1d
NOTE: PST update grp = 2 completed successfully
NOTE: initiating PST update: grp = 2, dsk = 0, mode = 0x5d
NOTE: PST update grp = 2 completed successfully
NOTE: initiating PST update: grp = 2, dsk = 0, mode = 0x7d
NOTE: PST update grp = 2 completed successfully
Tue Mar 20 08:29:17 2007
NOTE: initiating PST update: grp = 2, dsk = 0, mode = 0x7f
NOTE: PST update grp = 2 completed successfully
NOTE: completed online of disk group 2 disks 0
```

Database Compatibility **11.1.0.0.0**
 ASM Compatibility **11.1.0.0.0**
 Disk Repair Time (Hours) **3.6**

No data is currently available.

Member Disks

View By Failure Group

Expand All | Collapse All

Select	Failure Group	Path	Read/Write Errors	State	Mode	Size (GB)	Used (GB)	Used (%)
<input type="radio"/>	NEWGROUP							
<input type="radio"/>	NEWGROUP_0000		0	NORMAL	OFFLINE	0.00	0.00	0.00
<input type="radio"/>	NEWGROUP_0001		0	NORMAL	OFFLINE	0.00	0.00	0.00
<input type="radio"/>	FGNEW2							
<input type="radio"/>	NEWGROUP_0002	/u01/ASM/2/DISK8	0	NORMAL	ONLINE	0.12	0.05	38.28
<input type="radio"/>	NEWGROUP_0003	/u01/ASM/2/DISK0	0	NORMAL	ONLINE	0.12	0.05	36.72

General Performance Templates Files

[Database](#) | [Setup](#) | [Preferences](#) | [Help](#) | [Logout](#)

Copyright © 1996, 2006, Oracle. All rights reserved.
 Oracle, JD Edwards, PeopleSoft, and Retek are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.
[About Oracle Enterprise Manager](#)

FIGURE 4-6. The EM Member Disks screen indicates which disks are offline.

Once the disk is brought back online, the REPAIR_TIMER is reset to 0 and the MODE_STATUS is set to ONLINE.

At first glance, the Fast Disk Resync feature may seem to be a substitute for Dirty Region Logging (DRL), which several logical volume managers such as Veritas VxVM implement. However, Fast Disk Resync and DRL are distinctly different.

DRL is a mechanism to track blocks that have writes in flight. A mirrored write cannot be issued unless a bit in the DRL is set to indicate there may be a write in flight. Because DRL itself is on disk and also mirrored, it may require two DRL writes before issuing the normal mirrored write. This is mitigated by having each DRL bit cover a range of data blocks such that setting one bit will cover multiple mirrored block writes. There is also some overhead for I/O to clear DRL bits for blocks that are no longer being written. You can often clear these bits while setting another bit in DRL.

If a host dies while it has mirrored writes in flight, then it is possible that one side of the mirror is written and the other is not. Most applications require that they

get the same data every time if they read a block multiple times without writing it. If one side was written but not the other, then different reads may get different data. DRL mitigates this by constructing a set of blocks that must be copied from one side to the other to ensure that all blocks are the same on both sides of the mirror. Usually this set of blocks is much larger than those that were being written at the time of the crash and it takes a while to create the copies.

During the copy, the storage is unavailable, which increases overall recovery times. Additionally, it is also possible that the failure caused a partial write to one side, resulting in a corrupt logical block. The copying may write the bad data over the good data because the volume manager has no way of knowing which side is good.

Fortunately, when you use ASM, you need not maintain a DRL or blindly copy from one mirror to the other before beginning recovery. The only application that uses ASM is the Oracle database, and Oracle knows how to recover its data so that the mirror sides are the same for the cases that matter. It is not always necessary to make the mirror sides the same. For example, if a file is being initialized before it is part of the database, then it will be reinitialized after a failure, so that file does not matter for the recovery process. For data that does matter, Oracle must always have a means of tolerating a write that was started but which might not have been completed. The redo log is an example of one such mechanism in Oracle. Because Oracle already has to reconstruct such interrupted writes, it is simple to rewrite both sides of the mirror even if it looks like the write completed successfully. The number of extra writes can be small, because Oracle is excellent at determining exactly which blocks need recovery.

Another benefit of not using a DRL is that a corrupt block, which does not report an I/O error on read, can be recovered from the good side of the mirror. When a block corruption is discovered, each side of the mirror is read to determine whether one of them is valid. If the sides are different and one is valid, then the valid copy is used and rewritten to both sides. This can repair a partial write at host death. This mechanism is used all the time, not just for recovery reads. Thus an external corruption that affects only one side of an ASM mirrored block can also be recovered.

I/O Error-Failure Management and ASM

Whereas the previous section covers ASM handling of transient and permanent disk failures in ASM redundancy diskgroups, this section discusses how ASM processes I/O errors, such as read and write errors, and also discusses in general how to handle I/O failures in external redundancy diskgroups.

General Disk Failure Overview

Disk drives are mechanical devices and thus tend to fail. As drives begin to fail or have sporadic I/O errors, database failures become more likely.

The ability to detect and resolve device path failures is a core component of Path Managers as well as HBAs. A disk device can be in the following states or have the following issues:

- **Media sense errors** These include hard read errors and unrecoverable positioning errors. In this situation, the disk device is still functioning and responds to `SCSI_INQUIRY` requests.
- **Device too busy** A disk device can become so overwhelmed with I/O requests that it will not respond to the `SCSI_INQUIRY` within a reasonable amount of time.
- **Failed device** In this case, the disk has actually failed and will not respond to a `SCSI_INQUIRY` request, and when the `SCSI_INQUIRY` timeout occurs, the disk and path will be taken offline.
- **Path failure** The disk device may be intact, but a path component—such as a port or a fiber adapter—has failed.

In general, I/O requests can time out because either the SCSI driver device is unable to respond to a host message within the allotted time or the path on which a message was sent has failed. To detect this path failure, HBAs typically enable a timer each time that a message is received from the SCSI driver. A link failure is thrown if the timer exceeds the link-down timeout without receiving the I/O acknowledgment. After the link-down event occurs, the Path Manager determines that the path is dead and evaluates whether to reroute queued I/O requests to alternate paths.

ASM and I/O Failures

The method that ASM uses to handle I/O failures depends on the context in which the I/O failure occurred. If the I/O failure occurs in the database instance, then it notifies ASM, and ASM decides whether to take the disk offline. ASM takes whatever action is appropriate based on the redundancy of the diskgroup and the number of disks that are already offline.

If the I/O error occurs while ASM is trying to mount a diskgroup, the behavior depends on the release. In Oracle Database 10g Release 1, if the instance mounting the diskgroup is the first instance in the cluster to mount the diskgroup, it attempts to take offline any disks that it cannot discover, such as those that threw I/O errors while trying to read the disk header for mounting. In Oracle Database 10g Release 2, if the instance is not the first to mount the diskgroup in the cluster, it will not attempt to take any disks offline that are online in the diskgroup mounted by other instances. If none of the disks can be found, the mount will fail. The rationale here is that if the disk in question has truly failed, the running instances will very quickly take the disk offline.

If the error is local and you want to mount the diskgroup on the instance that cannot access the disk, you need to drop the disk from a node that the diskgroup mounted. Note that a drop force command will allow the mount immediately. Often in such scenarios, the disk cannot be found on a particular node because of errors in the `ASM_DISKSTRING` or the permissions on the node.

In Oracle Database 11g, these two behaviors are still valid, but rather than choosing one or the other based on whether the instance is first to mount the diskgroup, the behavior is based on the type of mount. The diskgroup `MOUNT [NOFORCE]`, which is the default, requires that all disks in the diskgroup be found at mount time. If any disks are missing (or have I/O errors), the mount will fail. `MOUNT FORCE` attempts to take disks offline as necessary allow the mount to complete. Note that to discourage the excessive use of `FORCE`, `MOUNT FORCE` succeeds only if a disk needs to be taken offline.

ASM, as well as the database, takes proactive measures to handle I/O failures or data corruptions.

When the database reads a data block from disk, it validates the checksum, the block number, and some other fields. If the block fails the consistency checks, then an attempt is made to reread the block to get a valid block read. Oracle can read individual mirror sides to resolve corruption since Oracle Database version 7.3. It was originally implemented with the Veritas volume manager, but now works with a number of volume managers including ASM. For corrupt blocks in datafiles, the database code reads each side of the mirror and looks for a good copy. If it finds a good copy, then the read succeeds and the good copy is written back to disk to repair the corruption, assuming that the database is holding the appropriate locks to perform a write. If the mirroring is done in a storage array (external redundancy), then there is no interface to select mirror sides for reading. In that case, the RDBMS simply rereads the same block and hopes for the best; however, with a storage array, this process will most likely return the same data from the array cache unless the original read was corrupted. If the RDBMS cannot find good data, then an error is signaled. The corrupt block is kept in buffer cache (if it is a cache-managed block) to avoid repeated attempts to reread the block and to avoid excessive error reporting. Note that the handling of corruption is different for each file type and for each piece of code that accesses the file. For example, the handling of datafile corruption during a RMAN backup is different from that described in this section, and the handling of archive logfile corruption.

ASM, like most volume managers, does not do any proactive polling of the hardware looking for faults. Servers usually have enough I/O activity to make such polling unnecessary. Moreover, ASM cannot tell whether an I/O error is due to a cable being pulled or a disk failing. It is up to the operating system (OS) to decide when to return an error or continue waiting for an I/O completion. ASM has no control over how the OS handles I/O completions. The OS signals a permanent I/O error to the caller (the Oracle I/O process) after several retries in the device driver.

**NOTE**

Starting with Oracle Database 11g, In the event of a disk failure, ASM polls disk partners and the other disks in the failure group of the failed disk. This is done to efficiently detect a pathological problem that may exist in the failure group.

ASM takes disks offline from the diskgroup only on a write operation I/O error, not for read operations. For example, in Oracle Database 10g, if a permanent disk I/O error is incurred during an Oracle write I/O operation, then ASM takes the affected disk offline and immediately drops it from the diskgroup, thus preventing stale data reads. In Oracle Database 11g, if the `DISK_REPAIR_TIMER` attribute is enabled, then ASM takes the disk offline but does not drop it. However, ASM does drop the disk if the `DISK_REPAIR_TIMER` expires. This feature is covered in the section “Recovering from Disk Failures in Oracle Database 11g—Fast Disk Resync,” earlier in this chapter.

In Oracle Database 11g, ASM attempts to remap bad blocks if a read fails. This remapping can lead to a write, which could lead to ASM taking the disk offline. For read errors, the block is read from the secondary extents (only for normal or high redundancy). If the loss of a disk would result in data loss, as in the case where a disk’s partner disk is also offline, ASM automatically dismounts the diskgroup to protect the integrity of the diskgroup data.

**NOTE**

Read failures from disk header and other unmirrored, physically addressed reads also cause ASM to take the disk offline.

In Oracle Database 11g, if a disk fails, ASM proactively reads all the disk headers of all the partners of the failing disk in an effort to avoid taking disk offline unnecessarily and to ensure partnership availability. After taking a disk offline, ASM checks the disk headers of all disks in that failure group to drop a failure group proactively in the case of failure group. ASM dismounts a diskgroup rather than taking some disks offline and then dismounting the diskgroup in case of apparent failures of disks in multiple failure groups. Also, ASM takes disks in a failure group offline all at once to allow for more efficient repartnering.

If the heartbeat cannot be written to a copy of the PST in a normal- or high-redundancy diskgroup, then ASM takes the disk containing the PST copy offline and moves the PST to another disk in the same diskgroup. In an external redundancy diskgroup, the diskgroup is dismounted if the heartbeat write fails. The heartbeat block is not normally read except at diskgroup mount. At mount time, it is read twice, at least 6 sec apart, to determine whether an instance outside the local

cluster mounts the diskgroup. If the two reads show different contents, then the diskgroup has mounted an unseen instance.

In the following example, ASM detects I/O failures as shown from the alert log:

```

Wed May 10 08:13:47 2006
NOTE: cache initiating offline of disk 1 group 1
WARNING: offlining mode 3 of disk 1/0x0 (DATA_1_0001)
NOTE: halting all I/Os to diskgroup DATA_1
NOTE: active pin found: 0x0x546d24cc
Wed May 10 08:13:52 2006
ERROR: PST-initiated MANDATORY DISMOUNT of group DATA_1

```

The following warning indicates that ASM detected an I/O error on a particular disk:

```

WARNING: offlining mode 3 of disk 1/0x0 (DATA_1_0001)

```

This error message alerts the user that trying to take the disk offline would cause data loss, so ASM is dismounting the diskgroup instead:

```

ERROR: PST-initiated MANDATORY DISMOUNT of group DATA_1

```

Messages should also appear in the OS log indicating problems this same disk (DATA_1_0001).

Many users want to simulate corruption in an ASM file in order to test failure and recovery. Two types of failure injection tests that customers induce are block corruption and disk failure. Unfortunately, overwriting an ASM disk simulates corruption, *not* a disk failure. Note further that overwriting the disk will corrupt ASM metadata as well as database files. This may not be the user's intended fault injection testing. You must be cognizant of the redundancy type deployed before deciding on the suite of tests run in fault injection testing. In cases where a block or set of blocks is physically corrupted, ASM, like some other host volume managers, attempts to reread all mirror copies of a corrupt block to find one copy that is not corrupt. So redundancy does matter when recovering a corrupt block. The source of the corruption also matters. If bad data are written to disk through ASM or any other volume manager, they will go to all copies of the mirror. An example of this is a logical corruption, such as bad redo generation. Additionally, normal redundancy would not help if the corruption was systematic so that it affected multiple disks (for example, if a storage administrator changed multiple disks).

Space Management Views for ASM Redundancy

Two V\$ ASM views provide more accurate information on free space usage: `USABLE_FREE_SPACE` and `REQUIRED_MB_FREE`.

In Oracle Database 10g Release 1, the `FREE_MB` value that is reported in `V$ASM_DISKGROUP` does not take into account mirrored extents. Oracle Database 10g Release 2 introduced a new column in `V$ASM_DISKGROUP` called `USABLE_FILE_MB` to indicate the amount of free space that can be “safely” utilized taking mirroring into account. The column provides a more accurate view of usable space in the diskgroup. Note that for external redundancy, the column `FREE_MB` is equal to `USABLE_FREE_SPACE`.

Be careful of cases in which `USABLE_FILE_MB` has negative values in `V$ASM_DISKGROUP`. `USABLE_FILE_MB` can go negative due to the relationship among `FREE_MB`, `REQUIRED_MIRROR_FREE_MB`, and `USABLE_FILE_MB`. 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 may result in files with reduced redundancy or can result in an out-of-space condition, which can hang the database. If `USABLE_FILE_MB` becomes negative, it is strongly recommended that you add more space to the diskgroup as soon as possible.

Along with `USEABLE_FREE_SPACE`, another column, `REQUIRED_MB_FREE`, has been added to `V$ASM_DISKGROUP` to indicate more accurately the amount of space that is required to be available in a given diskgroup to restore redundancy after one or more disk failures. The amount of space displayed in this column takes into account mirroring.

Diskgroups and Attributes

Oracle Database 11g introduced the concept of ASM attributes. Unlike initialization parameters, which are instance-specific but apply to all diskgroups, ASM attributes are diskgroup-specific and apply to all instances. The ASM diskgroup attributes are shown in the `V$ASM_ATTRIBUTES` view. However, this view is not populated until the diskgroup compatibility—that is, `COMPATIBLE.ASM`—is set to 11.1.0.

In Oracle Database 11g, the following attributes can be set:

- `COMPATIBLE.ASM`
- `COMPATIBLE.RDBMS`
- `DISK_REPAIR_TIME`
- `AU_SIZE`

The diskgroup attributes can be set at diskgroup creation or by using the `ALTER DISKGROUP` command. For example, a diskgroup can be created with 10.1 diskgroup compatibility and then advanced to 11.1 by setting the `COMPATIBLE.ASM` attribute to 11.1. The discussion on compatibility attributes is covered in the next section.

The following example shows a CREATE DISKGROUP command that results in a diskgroup with 10.1 compatibility (the default):

```
SQL> CREATE DISKGROUP DATA DISK '/dev/rdisk/c3t19d16s4',
'/dev/rdisk/c3t19d17s4' ;
```

This diskgroup can then be advanced to 11.1 using the following command:

```
SQL> ALTER DISKGROUP DATA SET ATTRIBUTE 'compatible.asm' = '11.1.0.0.0';
```

On successful advancing of the diskgroup, the following message appears:

```
SUCCESS: Advancing ASM compatibility to 11.1.0.0.0 for grp 1
```

In another example, the AU_SIZE attribute, which dictates the allocation unit size, and the COMPATIBLE.ASM attributes are specified at diskgroup creation. Note that the AU_SIZE attribute can only be specified at diskgroup creation and cannot be altered using the ALTER DISKGROUP command:

```
SQL> CREATE DISKGROUP FLASH DISK '/dev/raw/raw15', '/dev/raw/raw16',
'/dev/raw/raw17' ATTRIBUTE 'au_size' = '16M', 'compatible.asm' = '11.1';
```

The V\$ASM_ATTRIBUTE view can be queried to get the DATA diskgroup attributes:

```
SQL> SELECT NAME, VALUE FROM V$ASM_ATTRIBUTE WHERE GROUP_NUMBER=1;
NAME                                VALUE
-----
disk_repair_time                    4 H
compatible.asm                      11.1.0.0.0
compatible.rdbms                    11.1.0.0.0
```

In the previous example, the COMPATIBLE.ASM was advanced, this next example advances the COMPATIBLE.RDBMS attribute. Notice that the version is set to simply 11.1, which is equivalent to 11.1.0.0.0.

```
SQL> ALTER DISKGROUP DATA SET ATTRIBUTE 'COMPATIBLE.RDBMS' = '11.1';
NAME                                VALUE
-----
disk_repair_time                    4 H
compatible.asm                      11.1.0.0.0
compatible.rdbms                    11.1
```

The range of values for COMPATIBLE.RDBMS or COMPATIBLE.ASM can be from one version to five versions—for example, from 11.1 to 11.1.0.0.0:

```
SQL> ALTER DISKGROUP ORADATA SET ATTRIBUTE 'DISK_REPAIR_TIME' = '4 H';
```

This is covered in more detail in the next section. For more details on `DISK_REPAIR_TIME`, see the section “Recovering from Disk Failures in Oracle Database 11g—Fast Disk Resync” earlier in this chapter.

ASM and Database Compatibility

When a database instance first connects to an ASM instance, it negotiates the highest Oracle version that can be supported between the instances. There are two types of compatibility settings between ASM and the RDBMS: instance-level software compatibility settings and diskgroup-specific settings.

Instance-level software compatibility is defined using the `init.ora` parameter `COMPATIBLE`. This `COMPATIBLE` parameter, which can be set to 11.1, 10.2, or 10.1 at the ASM or database instance level, defines what software features are available to the instance. Setting the `COMPATIBLE` parameter in the ASM instance to 10.1 precludes the use of any new features that are introduced in Oracle Database 11g, such as disk online/offline and variable extents. Using lower values of the `COMPATIBLE` parameter for an ASM instance is not useful, because ASM is compatible with multiple database versions. Note that the `COMPATIBLE.ASM` value must be greater than or equal to that of `COMPATIBLE.RDBMS`.

The other compatibility settings are specific to a diskgroup and control which attributes are available to the ASM diskgroup and which are available to the database. This is defined by the ASM compatibility (`COMPATIBLE.ASM`) and RDBMS compatibility (`COMPATIBLE.RDBMS`) attributes, respectively. These compatibility attributes are persistently stored in the diskgroup metadata.

RDBMS Compatibility

RDBMS diskgroup compatibility is defined by the `COMPATIBLE.RDBMS` attribute. This attribute, which defaults to 10.1 in Oracle Database 11g, is the minimum `COMPATIBLE` version setting of a database that can mount the diskgroup. RDBMS compatibility also dictates the format of the messages exchanged between ASM and RDBMS instances. After the diskgroup attribute of `COMPATIBLE.RDBMS` is advanced to 11.1, it cannot be reversed.

ASM Compatibility

ASM diskgroup compatibility, as defined by `COMPATIBLE.ASM`, controls the persistent format of the on-disk ASM metadata structures. The ASM compatibility defaults to 10.1 and must always be greater than or equal to the RDBMS compatibility level. After the compatibility is advanced to 11.1, it cannot be reset to lower versions. Any value up to the current software version can be set and will be enforced. All five parts of the version number may be specified. For example, valid values for compatibility can be 11.1.0.0, 11.1.0, 10.1.0.0, or 10.1. Oracle Releases 10.1.0.2 through 10.2.0 all use the same ASM and RDBMS compatibility number.

COMPATIBLE.RDBMS and COMPATIBLE.ASM together control the persistent format of the on-disk ASM metadata structures. The combination of the compatibility parameter setting of the database, the software version of the database, and the RDBMS compatibility setting of a diskgroup determines whether a database instance is permitted to mount a given diskgroup. The compatibility setting also determines which ASM features are available for a diskgroup.

The following query shows an ASM instance that was recently software upgraded from Oracle Database 10g to Oracle Database 11g:

```
SQL> SELECT NAME, BLOCK_SIZE, ALLOCATION_UNIT_SIZE "AU_SIZE", STATE,
COMPATIBILITY "ASM COMP", DATABASE_COMPATIBILITY "DB COMP" FROM V$ASM_DISKGRP;

NAME   BLOCK_SIZE AU_SIZE STATE   ASM_COMP  DB_COMP
-----
DATA   4096       1048576 MOUNTED 10.1.0.0.0 10.1.0.0.0
```

Notice that the ASM compatibility and RDBMS compatibility are still at the default (for upgraded instances) of 10.1. The 10.1 setting is the lowest attribute supported by ASM.

NOTE

An ASM instance can support different RDBMS clients with different compatibility settings, as long as the database COMPATIBLE init.ora parameter setting of each database instance is greater than or equal to the RDBMS compatibility of all diskgroups.

See the section “Diskgroups and Attributes,” earlier in this chapter, for examples on advancing the compatibility.

The ASM compatibility of a diskgroup can be set to 11.0 whereas its RDBMS compatibility could be 10.1, as in the following example:

```
SQL> SELECT DB_NAME, STATUS, SOFTWARE_VERSION, COMPATIBLE_VERSION FROM
V$ASM_CLIENT;

DB_NAME  STATUS      SOFTWARE_V  COMPATIBLE
-----
YODA     CONNECTED   11.1.0.6.0 11.1.0.0.0

SQL> SELECT NAME, COMPATIBILITY "COMPATIBLE", DATABASE_COMPATIBILITY
"DATABASE_COMP" FROM V$ASM_DISKGROUP

NAME           COMPATIBLE  DATABASE_COMP
-----
DATA           10.1.0.0.0 10.1.0.0.0
```

This implies that diskgroup can be managed only by ASM software version 11.0 or higher whereas any database software version must be 10.1 or higher.

Summary

A diskgroup is the fundamental object managed by ASM. It is composed of multiple ASM disks. Each diskgroup is self-describing—that is, all the metadata about the usage of the space in the diskgroup are completely contained within the diskgroup. If ASM can find all the disks in a diskgroup, it can provide access to the diskgroup without any additional metadata.

A given ASM file is completely contained within a single diskgroup. However, a diskgroup may contain files belonging to several databases and a single database may use files from multiple diskgroups. Most installations include only a small number of diskgroups—usually two and rarely more than three.