ASM Internals

By

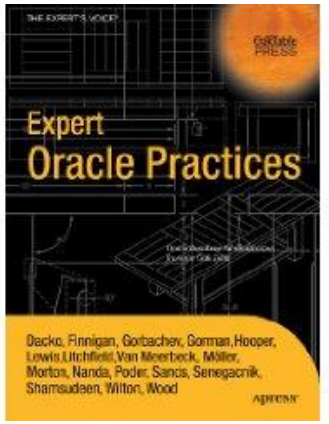
Riyaj Shamsudeen

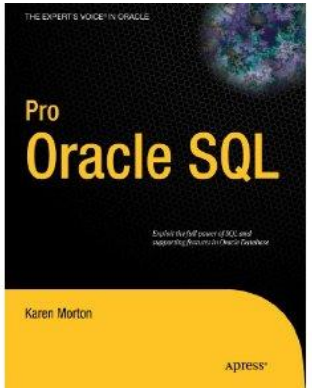
Me

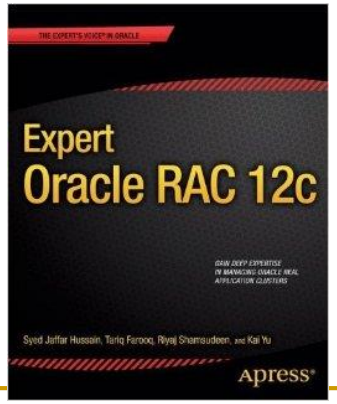
ORACLE ACE Director

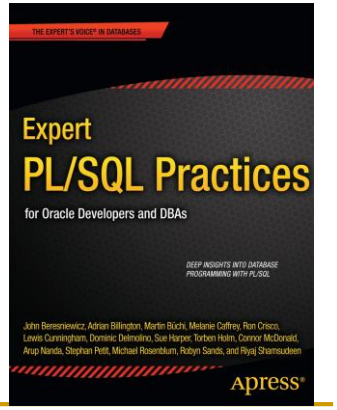
- *23+ years as DBA*
- *OakTable member*
- *Oracle ACE Director*
- *Specializes in RAC, performance tuning and Internals.*
- *Slowly in to BigData*

- *rshamsud@orainternals.com*
- *orainternals.wordpress.com*
- *Web: www.orainternals.com*









WARNING

Most of the topics in this presentations are from my research.

Writing about internals have issues:

a. I completely misunderstood the data and trace files.

b. Future version changed the feature, so, information is outdated.

Tested in version 11g, 12.1.0.2, Linux and Solaris 11 platform.

AGENDA

ASM overview: Instance, asmb etc

Tools: kfod, kfed, amdu

Disk group, redundancy, AU

ASM rebalance

Asmcmd

Conclusion

Architecture

- ASM is an Oracle Instance with instance_type='ASM'
- ASM manages disks, luns and externalizes files to RDBMS
- ASM instance is never opened. **Simply in a mount state**.

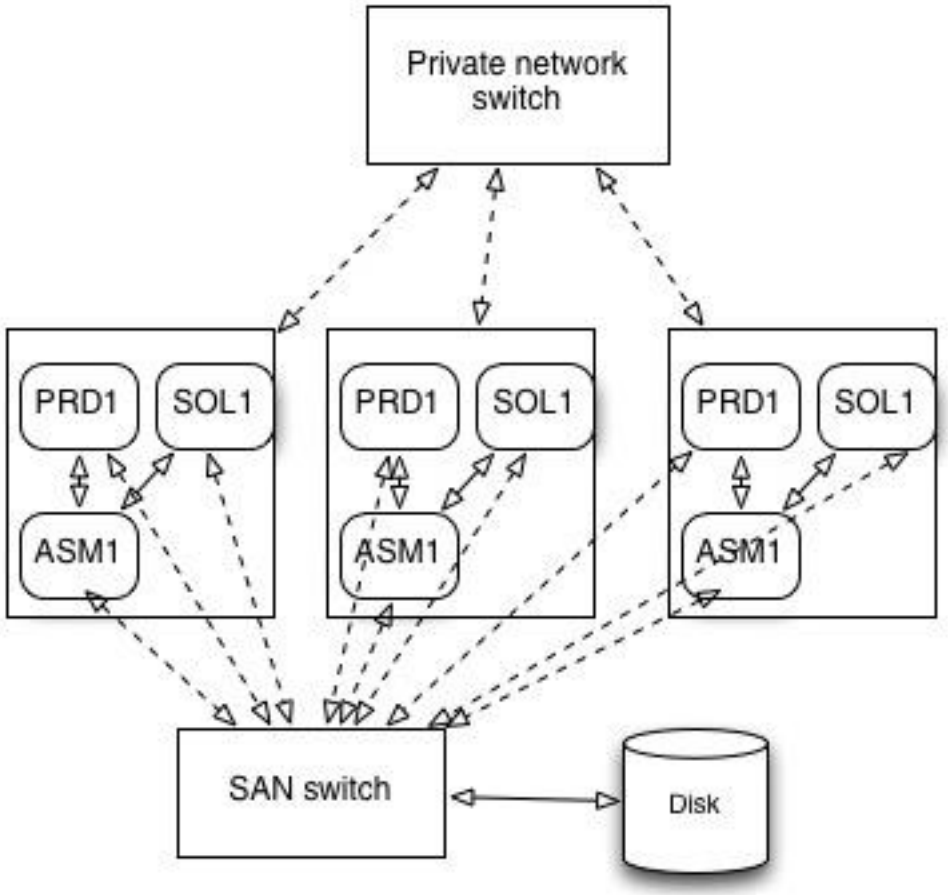
Demo: v$instance

Architecture

- ASM provides an extent map of files to RDBMS.
- **RDBMS directly accesses the disk to perform I/O**. ASM is not involved in I/O operation.
- Extending files or adding data files will involve refresh of extent map from ASM to RDBMS.

Demo: v$instance

Architecture: With ASM



RDBMS I/O

- Truss of DBWR: ASM is not involved for RDBMS I/O to the devices.

Write calls to file pointer 262: (truss output)

```
/1:     kaio(AIOWRITE, 262 , 0x6DD3C000, 8192, 0xFC17E6380F4E4000) = 0
...
/1:     kaio(AIOWRITE, 262 , 0x7DF3F000, 49152, 0xFC17D7080BD8A000) = 0
```

File pointer 262 is a SCSI device (pfiles output)

```
 262: S_IFCHR mode:0755 dev:291,0 ino:15728902 uid:601 gid:503 rdev:30,129
      O_RDWR|O_NONBLOCK|O_DSYNC|O_LARGEFILE FD_CLOEXEC
      /devices/iscsi/disk@0000iqm.demo.volumes-san0001,1:b,raw
```

RDBMS is a client (aka umbilical process)

- asmb process running in RDBMS instance makes a connection to ASM instance, as a foreground process for ASM instance.
- asmb process sleeps in a loop and a primary mechanism to detect ASM crash.
- If ASM instance crashes, asmb connection will die leading to an RDBMS instance crash.

Demo: asm_connections.sql, asm_clients.sql

RDBMS as a client

- Truss of a RDBMS startup shows that a LOCAL connection was made to the ASM instance.

```
1821: 2.9102 execve("/u02/app/11.2.0/grid/bin/oracle",0x0E8E87F0,0x0E9ED510)
```

- Instance restart alone opens 6 different connections to ASM instance. You need to set processes parameter appropriately.

```
grep execve  truss_startup.lst |grep grid
1821:    2.9102  0.0015 execve("/u02/app/11.2.0/grid/bin/oracle", 0x0E8E87F0, 0x0E9ED510)  argc = 2
1941:    8.8772  0.0019 execve("/u02/app/11.2.0/grid/bin/oracle", 0x0E8E8090, 0x0EA11970)  argc = 2
1966:   10.0884  0.0019 execve("/u02/app/11.2.0/grid/bin/oracle", 0x0E8E8090, 0x0EA11970)  argc = 2
2002:   12.7198  0.0020 execve("/u02/app/11.2.0/grid/bin/oracle", 0x0E8E7550, 0x0E99B220)  argc = 2
2010:   13.1669  0.0020 execve("/u02/app/11.2.0/grid/bin/oracle", 0x0E8E7550, 0x0E99B220)  argc = 2
2066:   29.0296  0.0024 execve("/u02/app/11.2.0/grid/bin/oracle", 0x0E8E7550, 0x0E99B220)  argc = 2
```

Death of asmb process

- asmb process sleeps on "ASM background timer" with 5s sleep cycle.

***** 2011-10-05 00:38:11.486**

WAIT #0: nam='ASM background timer' ela= 5001967 p1=0 p2=0 p3=0 obj#=-1 tim=1247836548

- I killed the connection from ASM instance, resulting in asmb process death, followed by RDBMS instance crash

NOTE: ASMB terminating

Errors in file /u01/app/oracle/diag/rdbms/solrac/solrac1/trace/solrac1_asmb_1492.trc:

ORA-15064: communication failure with ASM instance

ORA-03113: end-of-file on communication channel

Process ID:

Session ID: 30 Serial number: 3

…

ASMB (ospid: 1492): **terminating** the instance due to error 15064

Demo: Killing asmb connection

ASM extent pointer array

- v$sgastat shows the extent pointer array in the RDBMS. This array is retrieved from ASM instance.

```
select * from gv$sgastat where name like '%ASM extent%';
   INST_ID POOL         NAME                            BYTES
---------- ------------ -------------------------- ----------
         1 shared pool  ASM extent pointer array       171824
         2 shared pool  ASM extent pointer array       171824
```

- For large databases, this area will be bigger.

- To improve instance startup performance, only minimal extent mapping is retrieved initially. More data added to this array on need basis.

Minimal ASM parameters

- Instance_type='ASM'

  ASM instances named +ASMx

- SGA components are:

  db_cache_size =64M # To cache metadata blocks

  shared_pool_size=128M # for various structures for ASM

  large_pool_size =64M # for extent map operations

- I usually, set processes parameter to 25 + 12*# of databases.

- 11g+ supports automatic memory management and you can set memory_target =512M and let Oracle manage it.

Demo: Parameters, v$sgastat, show sga

ASM disks

- During ASM startup, ASM instance scans the disks to identify all ASM disks.
- Parameter asm_diskstring identifies the disks to scan.
- asm_diskstring accepts wildcard parameters and null is default. To improve ASM startup time, set this parameter properly.
- For example,

  Following value for asm_diskstring will search for all devices matching the wildcard and has read write permissions.

  asm_diskstring = /dev/rdsk/c2t*d0s1

Demo: show parameter asm_diskstring

kfod

Kfod –h for help

- kfod utility can be used to check all devices that qualifies asm_diskstring.

```
 $ kfod status=TRUE asm_diskstring='/dev/mapper/' disks=ALL verbose=TRUE
--------------------------------------------------------------------------------
 Disk          Size Header    Path                                           User     Group
================================================================================
   1:      20473 Mb MEMBER    /dev/mapper/asmdisk1p1                         oracle   oinstall
   2:      20473 Mb MEMBER    /dev/mapper/asmdisk2p1                         oracle   oinstall
--------------------------------------------------------------------------------
ORACLE_SID ORACLE_HOME
================================================================================
     +ASM1 /u01/app/12.1.0/grid
KFOD-00311: Error scanning device /dev/mapper/control
ORA-27041: unable to open file
Linux-x86_64 Error: 13: Permission denied
Additional information: 42
KFOD-00311: Error scanning device /dev/mapper/36000c29d5fb1e04764ebbedd94bb6acd
ORA-27041: unable to open file
Linux-x86_64 Error: 13: Permission denied
```

…

ASM disks - RAC

- A lun must be visible in all nodes of a cluster with proper permissions for ASM to consider a lun.
- This means that lun path need not be the same, but lun should exist and visible through asm_diskstring parameter.

- For example, same device have different names in two nodes:

```
node1 /dev/rdsk/c2t9d0s1
node2 /dev/rdsk/c2t11d0s1
```

- ASM identifies Lun even if configuration changes later

- Metadata kept in every disk header.

Demo: show parameter asm_diskstring

kfed disk header

- kfed utility can be used to dump the metadata block(s) of the device.
  - Without any parameter, kfed reads disk header.

```
$ kfed read  /dev/rdsk/c2t9d0s1
kfbh.endian:                          1 ; 0x000: 0x01
..
kfbh.type:                            1 ; 0x002: KFBTYP_DISKHEAD
..
kfbh.block.blk:                       0 ; 0x004: T=0 NUMB=0x0
kfbh.block.obj:              2147483655 ; 0x008: TYPE=0x8 NUMB=0x7…
kfdhdb.compat:                186646528 ; 0x020: 0x0b200000
kfdhdb.dsknum:                        7 ; 0x024: 0x0007
kfdhdb.grptyp:                        1 ; 0x026: KFDGTP_EXTERNAL
kfdhdb.hdrsts:                        3 ; 0x027: KFDHDR_MEMBER
kfdhdb.dskname:              DATA_0007 ; 0x028: length=9
kfdhdb.grpname:                   DATA ; 0x048: length=4
kfdhdb.fgname:               DATA_0007 ; 0x068: length=9
kfdhdb.capname:                       ; 0x088: length=0
```

Demo: kfed read

kfed ..2

```
kfdhdb.secsize:                 512 ; 0x0b8: 0x0200
kfdhdb.blksize:                4096 ; 0x0ba: 0x1000
kfdhdb.ausize:              1048576 ; 0x0bc: 0x00100000
kfdhdb.mfact:                113792 ; 0x0c0: 0x0001bc80
kfdhdb.dsksize:                2000 ; 0x0c4: 0x000007d0
kfdhdb.pmcnt:                     2 ; 0x0c8: 0x00000002
kfdhdb.fstlocn:                   1 ; 0x0cc: 0x00000001
kfdhdb.altlocn:                   2 ; 0x0d0: 0x00000002
kfdhdb.f1b1locn:                  0 ; 0x0d4: 0x00000000
kfdhdb.redomirrors[0]:            0 ; 0x0d8: 0x0000
kfdhdb.redomirrors[1]:            0 ; 0x0da: 0x0000
kfdhdb.redomirrors[2]:            0 ; 0x0dc: 0x0000
kfdhdb.redomirrors[3]:            0 ; 0x0de: 0x0000
kfdhdb.dbcompat:          168820736 ; 0x0e0: 0x0a100000
```

Demo: kfed read

kfed other blocks

- kfed can be used to read other blocks in the lun also.

```
$kfed read  /dev/rdsk/c2t9d0s1 aun=0 blkn=1 |grep kfbh.type
kfbh.type:                               2 ; 0x002: KFBTYP_FREESPC

$kfed read  /dev/rdsk/c2t9d0s1 aun=0 blkn=2 |grep kfbh.type
kfbh.type:                               3 ; 0x002: KFBTYP_ALLOCTBL

# ASM also stores backup disk header in the second allocation
   unit, last 2 blocks.
$kfed read  /dev/rdsk/c2t9d0s1 aun=1 blkn=254 |more
kfbh.type:                               1 ; 0x002: KFBTYP_DISKHEAD
kfbh.datfmt:                             1 ; 0x003: 0x01
…
```

Demo: kfed read

Corrupting header

- Minor header related repair possible

```
$ kfed read /dev/mapper/asmdisk4p1 |more
kfbh.endian:                          1 ; 0x000: 0x01
kfbh.hard:                          130 ; 0x001: 0x82
kfbh.type:                            1 ; 0x002: KFBTYP_DISKHEAD

$ dd if=/dev/zero of=/dev/mapper/asmdisk4p1  bs=1M count=1
1+0 records in
1+0 records out

$ kfed read /dev/mapper/asmdisk4p1 |more
kfbh.endian:                          0 ; 0x000: 0x00
kfbh.hard:                            0 ; 0x001: 0x00
kfbh.type:                            0 ; 0x002: KFBTYP_INVALID
```

Demo: kfed read

Kfed repair

```
$ kfed repair /dev/mapper/asmdisk4p1

$ kfed read /dev/mapper/asmdisk4p1 |more
kfbh.endian:                          1 ; 0x000: 0x01
kfbh.hard:                          130 ; 0x001: 0x82
kfbh.type:                            1 ; 0x002: KFBTYP_DISKHEAD
```

Demo: kfed read

amdu

amdu can be used to extract files, even when the disks are corrupt.

```
$ amdu -diskstring=/dev/mapper/asmdisk3p1
amdu_2017_01_14_07_36_15/

$ ls -lt amdu_2017_01_14_07_36_15/
total 4
-rw-r--r--. 1 oracle oinstall 1834 Jan 14 07:36 report.txt

$ more amdu_2017_01_14_07_36_15/report.txt
-*-amdu-*-
******************************** AMDU Settings
   ********************************
ORACLE_HOME = /u01/app/12.1.0/grid
System name:    Linux
Node name:      rac1.localdomain
Release:        3.8.13-44.el6uek.x86_64
Version:        #2 SMP Fri Aug 8 21:59:01 PDT 2014
Machine:        x86_64
amdu run:       14-JAN-17 07:36:15
Endianess:      1
```

amdu

```
---------------------------- DISK REPORT N0001 ------------------------------
                Disk Path: /dev/mapper/asmdisk3p1
           Unique Disk ID:
               Disk Label:
    Physical Sector Size: 512 bytes
               Disk Size: 2047 megabytes
              Group Name: TEST
               Disk Name: TEST_0000
     Failure Group Name: TEST_0000
             Disk Number: 0
          Header Status: 3
     Disk Creation Time: 2017/01/11 23:49:59.434000
        Last Mount Time: 2017/01/14 07:32:17.969000
   Compatibility Version: 0x0a100000(10010000)
        Disk Sector Size: 512 bytes
        Disk size in AUs: 2047 AUs
        Group Redundancy: 2
     Metadata Block Size: 4096 bytes
                 AU Size: 1048576 bytes
                  Stride: 113792 AUs
```

...

V$asm_disk

- V$asm_disk shows all the disks that ASM has visibility and access.
- Header_status shows the state of the disk.

```
select header_status, name from v$asm_disk;

HEADER_STATU NAME
------------ ------------------------------
MEMBER       DATA_0001
MEMBER       DATA_0002
```

Header status	Meaning
Member	Disk is part of the disk group
Candidate	Available to add
Former	Was part of another disk group
Provisioned	Linux specific, ASMLIB configured

Demo: asm_disks.sql

Multipathing & ASM

- ASM does not provide any multi-pathing solutions, but leverages the implemented solution.

- Multi-pathing solution should:

  1. Provide single block device interface to a lun with multiple paths.

  2. Handle the failover and load balancing between multiple paths.

  3. externalize just one path to ASM.

- ASM does not handle it properly if *a* disk is seen twice while scanning the devices.

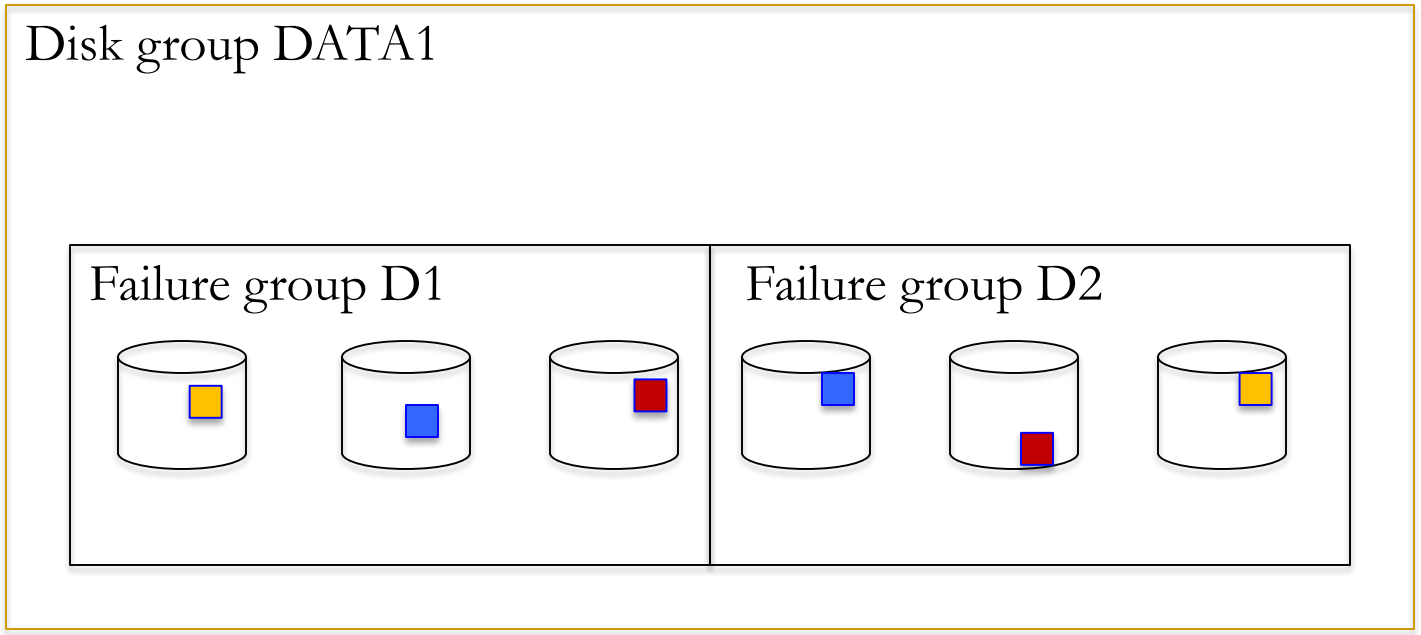
ASM disk group

- As the name suggests, it is a group of ASM disks ☺
- Essentially, ASM hides the disks underneath as an abstraction layer and provides files to the RDBMS/ACFS clients.
- Three types of redundancy implementations: External, normal, and high.
- With external redundancy ASM assumes that SAN takes care of redundancy.
- With normal redundancy, there are two copies managed by ASM. Three copies managed by ASM in the case of high redundancy.

Demo: asm_disk_group.sql, asm_disks.sql

ASM disk group

- Picture of a Disk group with Normal redundancy. Two failure groups are allocated since this is a mirrored disk group.
- ASM does not mirror disks, rather extents are kept in two separate failure groups.



Example

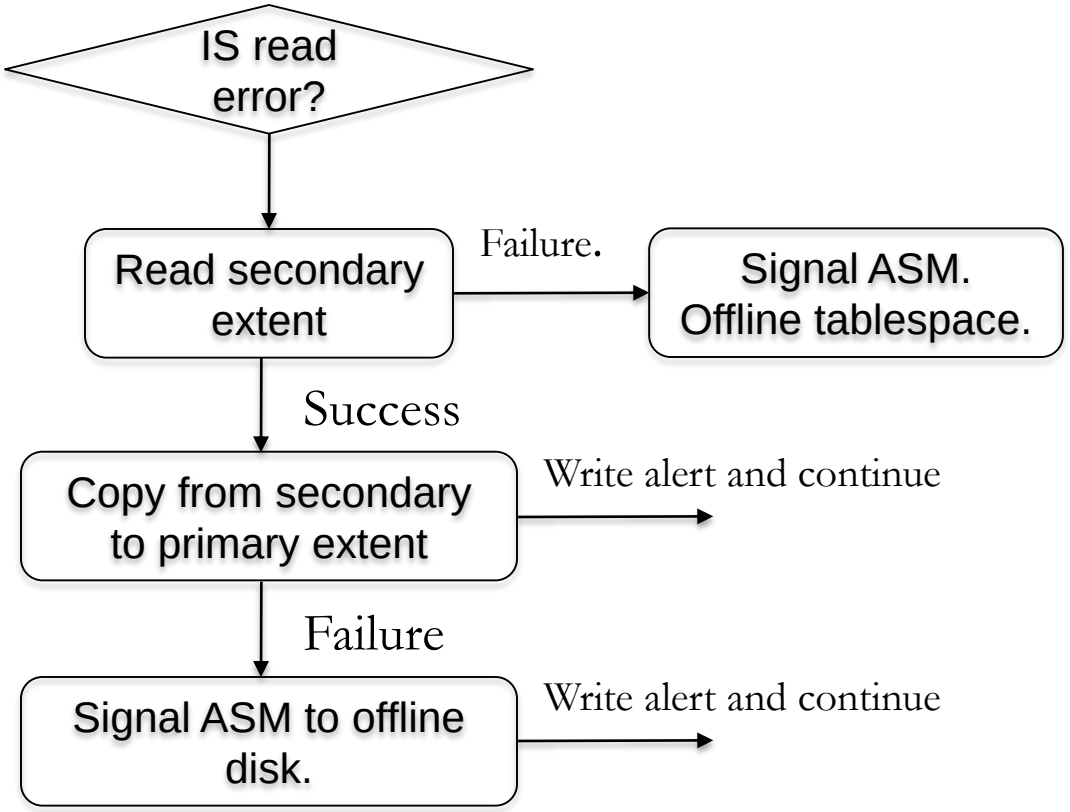
Construct the failure groups such a way that one component failure affects at the most one failure group.

```
create diskgroup DATA normal redundancy
    Failure group fl1 disk
    '/dev/rdsk/c3t11d3s4','/dev/rdsk/c3t11d4s4','/dev/rdsk/c3t11d5s4',
    '/dev/rdsk/c3t11d6s4'
    Failure group fl2 disk
    '/dev/rdsk/c4t12d3s4','/dev/rdsk/c4t12d4s4','/dev/rdsk/c4t12d5s4',
    '/dev/rdsk/c4t11ds4'
    Failure group fl3 disk
    /dev/rdsk/c5t13d3s4','/dev/rdsk/c5t13d4s4','/dev/rdsk/c5t13d5s4',
    '/dev/rdsk/c5t13ds4'
    Failure group fl4 disk
    /dev/rdsk/c6t14d3s4','/dev/rdsk/c6t14d4s4','/dev/rdsk/c6t14d5s4',
    '/dev/rdsk/c6t14ds4';
```

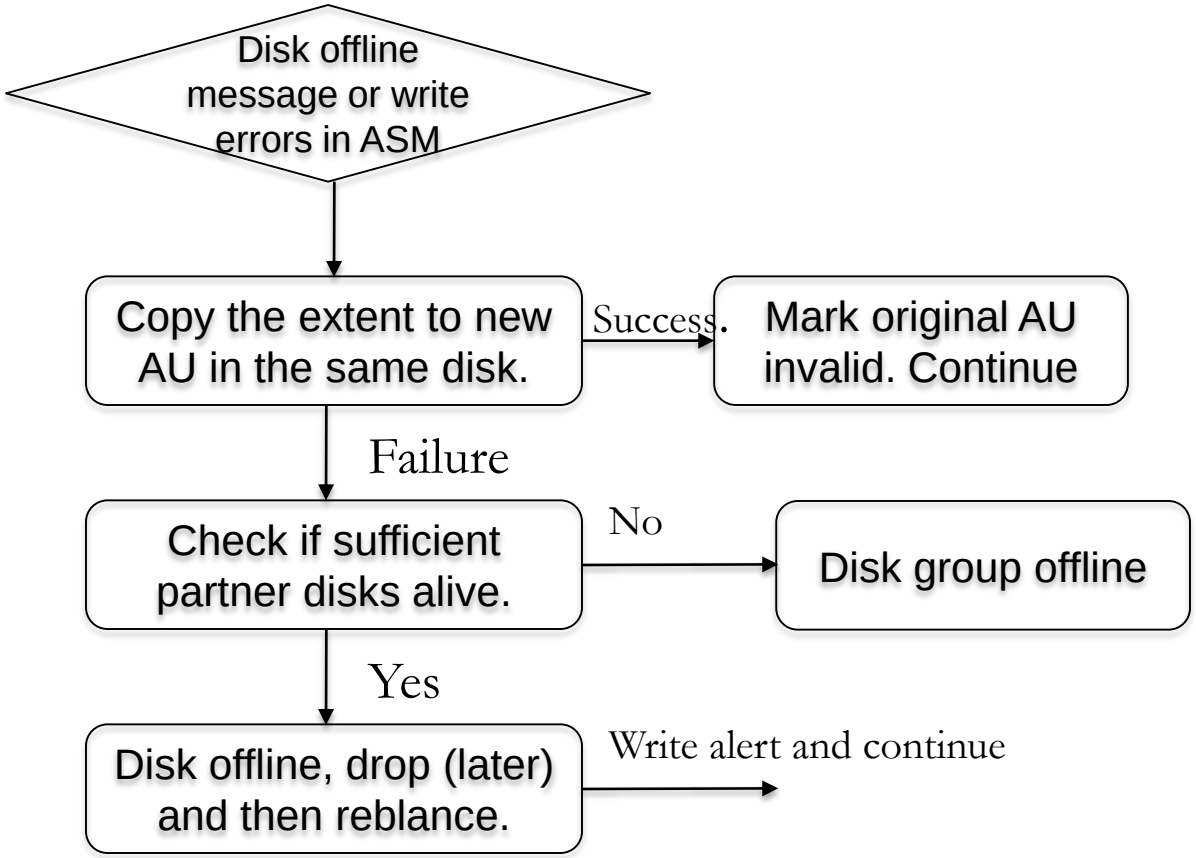
Redundancy & I/O

- In the case of Normal redundancy, there will be two write calls from the host side (by database).
- This could potentially be an issue if you go from external to normal redundancy.
- ASM tries to keep nearly same number of primary and secondary extents in each disk(lun).
- This provides an uniform distribution of I/O activity in all luns.
- But, ASM does not know anything about striping & mirroring in the SAN. Double SAME methodology in play, generally.

I/O Errors – Normal redundancy (DB)



I/O Errors – Normal redundancy (ASM)



Fast Mirror Resync

- Disk goes offline, if ASM encounters errors.
- But, in 11g, ASM doesn't drop the disk for 3.6 hours. After 3.6 hours, disks are dropped if it is not available.
- You can modify disk_repair_time from 3.6 hours.

```
ALTER DISKGROUP DATA SET ATTRIBUTE 'DISK_REPAIR_TIME'='10H';
```

- Idea here is that transient failures do not trigger massive resilvering activities.
- Changes to the extents are tracked in a bitmap, and this bitmap is used to copy the extents once the disks are available.
- This is truly useful, say, if a controller fails, as the disks are fine.

Failures and corruption

- ASM also reads only a primary extent normally. This means that the **corruption in the secondary extent will not be noticed** until the primary extent is not accessible.
- But, writes will write to both extents and so, can detect corruption.
- Hardware failures will be detected immediately though since each disk will have an approximately equal number of primary & secondary extents.

Diskgroup check

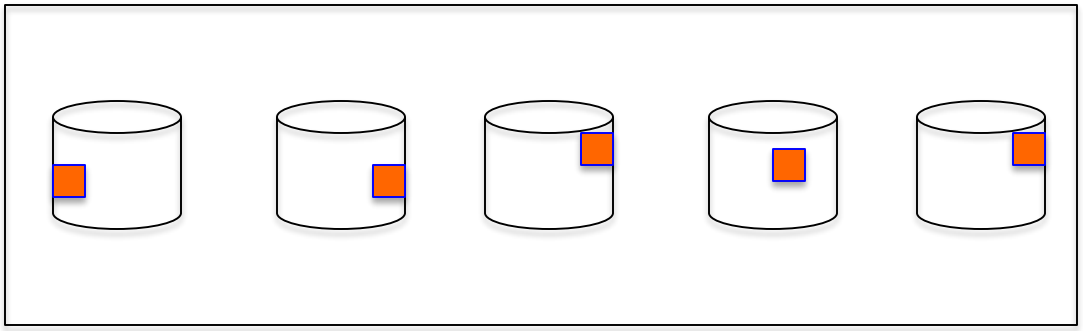
- If there are any disk errors, checking diskgroup might be a first step to take.
- Returns with no errors if the disk group is good.
- Checks for ASM metadata consistency:
  - Verifies file extent maps and allocation tables.
  - Verifies the directories, files, and aliases are correct.
  - Reads metadata and backup, and verifies them.

Use same size luns

- If a lun fails, then ASM will induce rebalance and will copy the extents from primary or secondary.
- Database will continue to read from the available mirror and will not see any errors.
- For these reasons, it is important to have same size luns in a disk group.
- We will discuss rebalance operation later.

How many disk groups?

- 2 or 3 (DATA, FRA,CRS)
- One disk group for database files (say DATA) and another group for flash recovery area (say FRA) is the recommended approach.
- ASM follows SAME methodology. For example, If there are 5 disks in a disk group (assuming external redundancy), file will be spread on all the available luns.

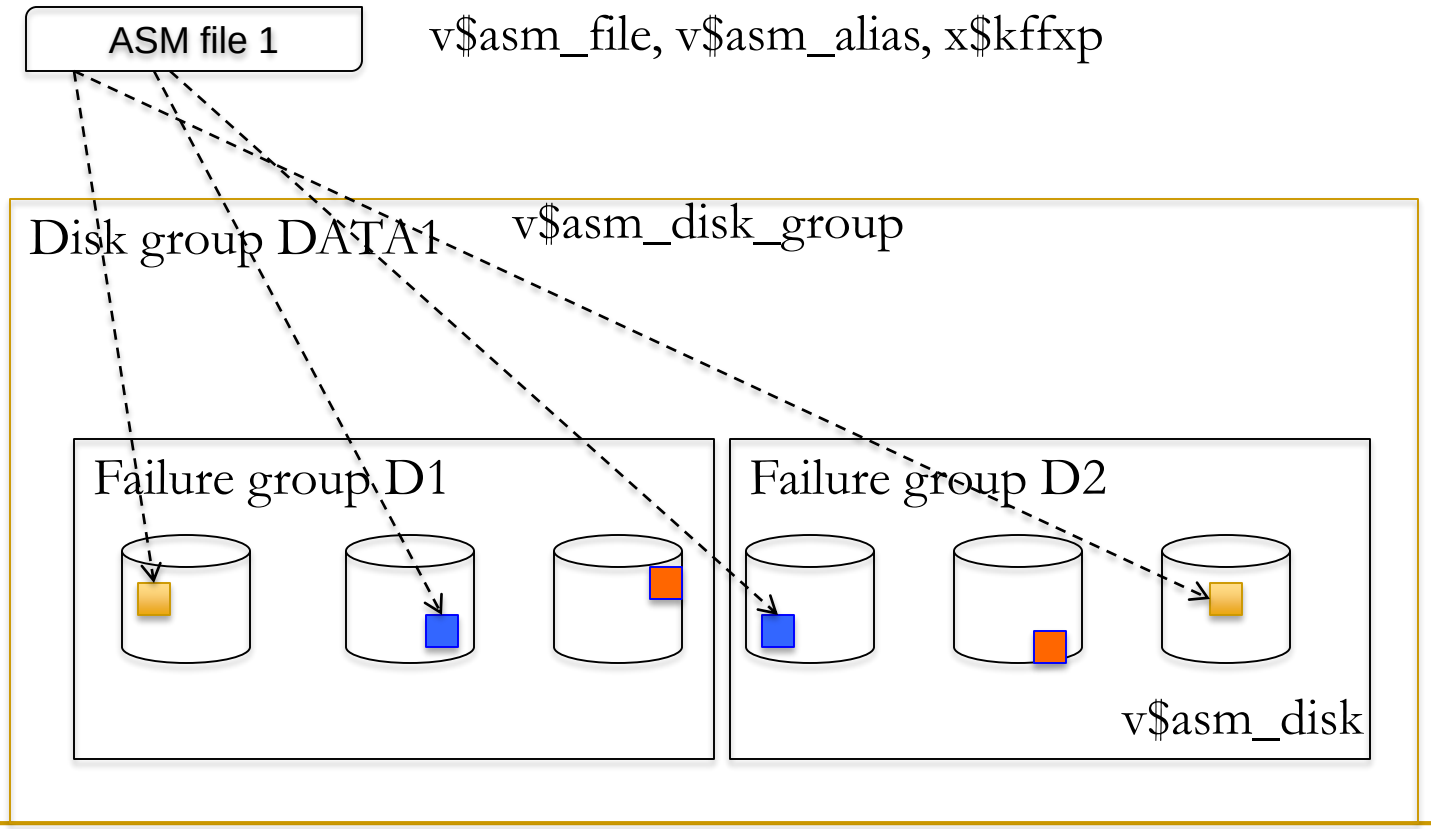


Redundant copies

- If there are two disk groups configured at DB creation time, a control file and a redo log file member will be placed automatically in both disk groups.
- You could do this manually too, later.
- Even if you have many database instances using that ASM, still, just 2 or 3 ASM disk group is the recommended approach.
- There is an exception: If you have tier 1 and tier 2 storage architecture, then it makes sense to have more disk groups.

ASM files – Normal redundancy

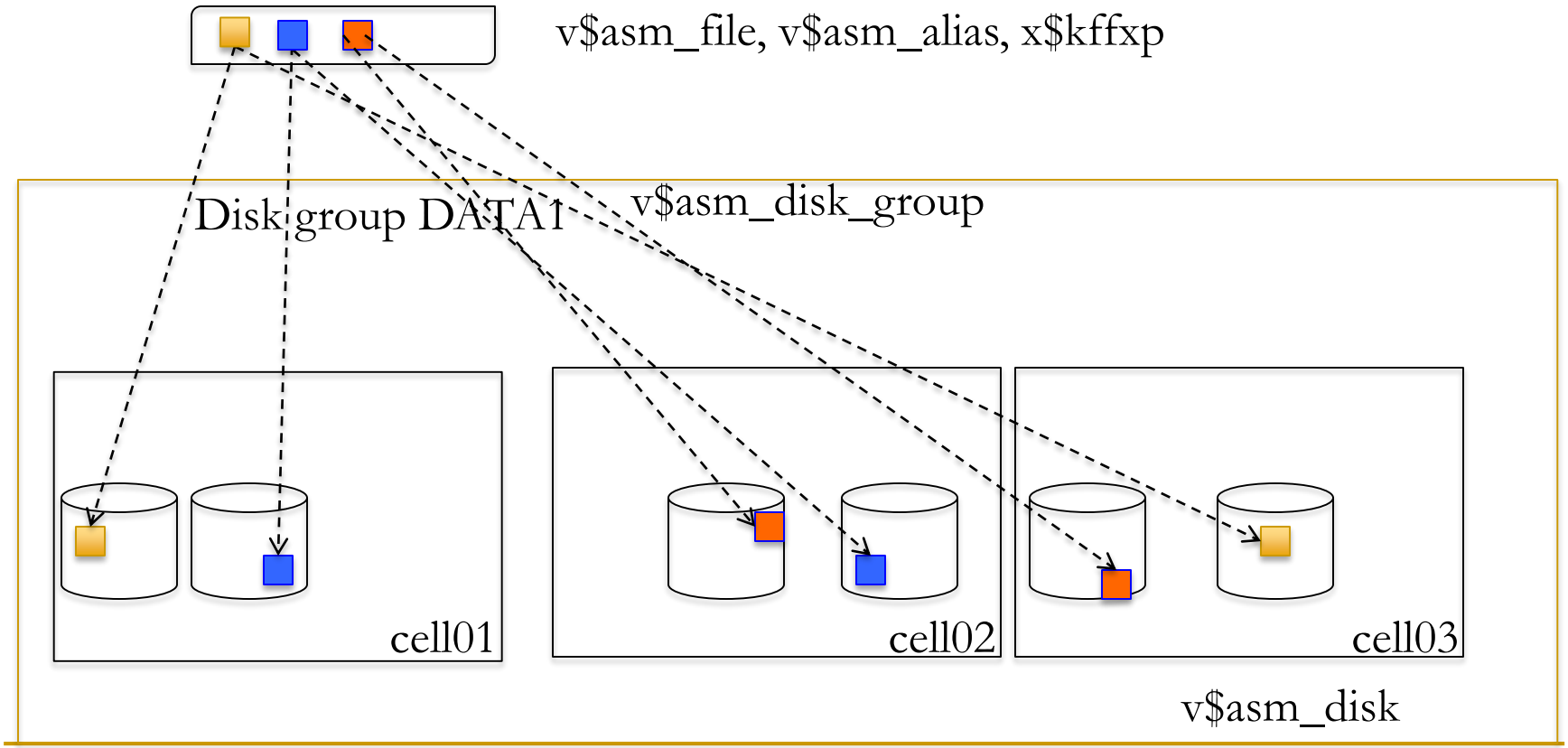
- ASM files are allocated from mirrored extents between the failure groups.



Demo: asm_file_analysis.sql

ASM files – Normal redundancy - Exadata

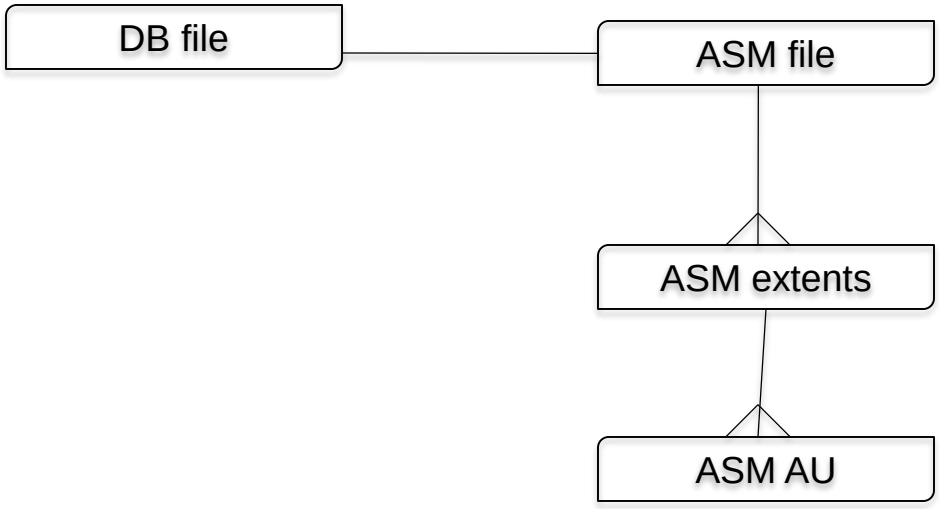
- ASM files are allocated from mirrored extents between the failure groups.



Demo: asm_file_analysis.sql

Extents vs Files

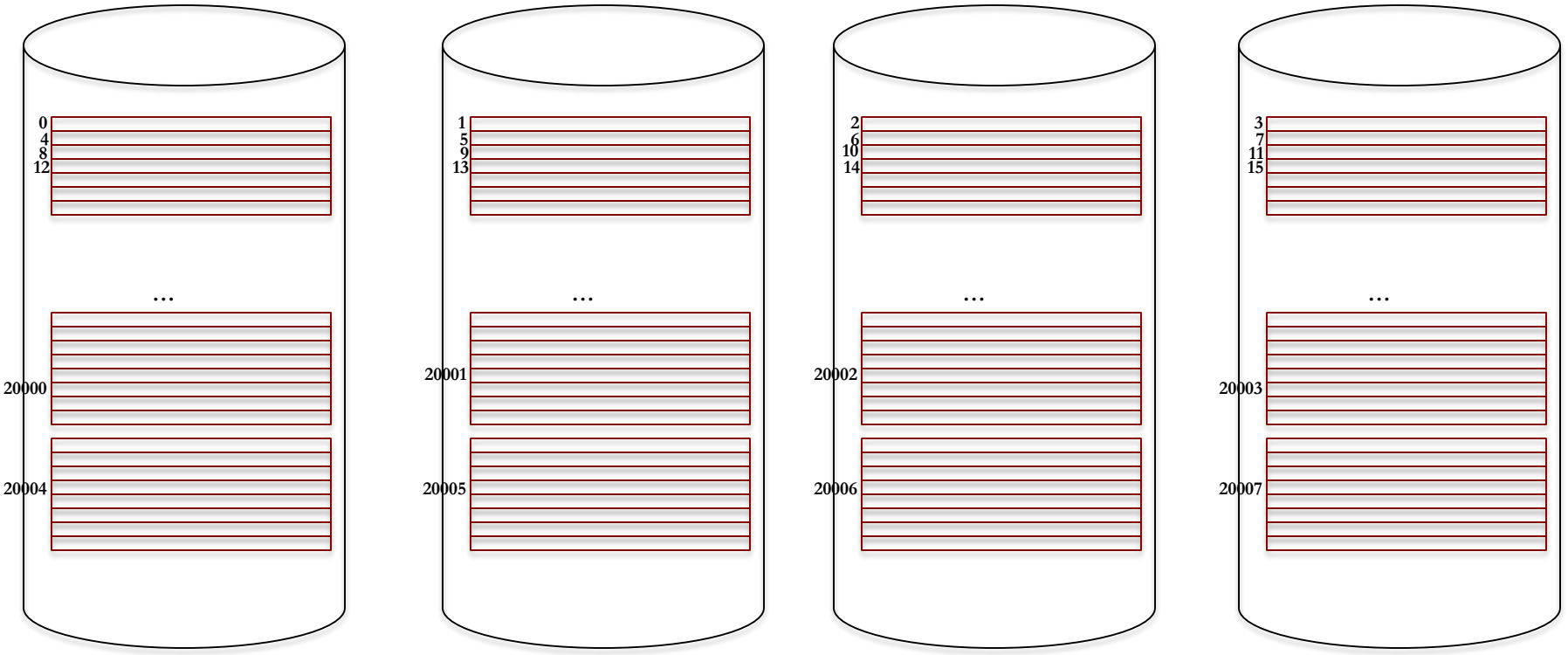
- ASM files are allocated as series of extents.
- ASM extents are made up of one or more allocation units.



- ASM extents are contained within an ASM disk though.

Extent vs AU

- 1 extent = 1 AU up to 20000 extents. 1 extent=8 AUs after 20000 extents.
- This is one asm file and so extents are distributed between the devices (striping).



Allocation_unit (AU)

- Allocation unit defines a smallest size disk segment that can be allocated, at disk group level.
- Allocation_unit defaults to 1MB. It can be increased in multiples of 2 i.e. 2,4,8,16MB etc while creating a diskgroup. (11g).
- Once a disk group is created with an allocation unit it can not be altered.
- In 10g, underscore parameters _asm_ausize can be used to modify the allocation_unit.
- Increased allocation_unit is useful in VLDB daabases.

Striping

- File extents are striped. There are two types of striping: coarse and fine.
- With coarse striping, one allocation unit is the size of stripe. This is used for database files.
- With fine striping, 128KB is interleaved with 8 allocation units. This type of striping is used for online redo log files, control files, and spfiles.
- Striping is controlled by templates.
- Template can be altered, but be careful of implications.

Demo: asm_templates.sql

ASM files

- You don't need to specify complete file name while creating file from the database.

- ASM will generate a system defined unique file name if you don't specify complete path.

```
create tablespace ts_small datafile '+DATA' size 10M;
select file_name from dba_data_files where tablespace_name='TS_SMALL'
FILE_NAME
--------------------------------------------------------------------------------
+DATA/solrac/datafile/ts_small.281.764615081
```

Demo: cr_ts_small.sql

ASM Directory

- You can create directory structure in ASM and use that for file names (ASM instance).

```
SQL>alter diskgroup data add  directory '+DATA/app';
SQL>alter diskgroup data add  directory '+DATA/app/oracle';
```

- A new file with user defined file name can be added to the database.

```
SQL> alter tablespace ts_small add datafile
        '+DATA/app/oracle/ts_small_02.dbf' size 10M;
```

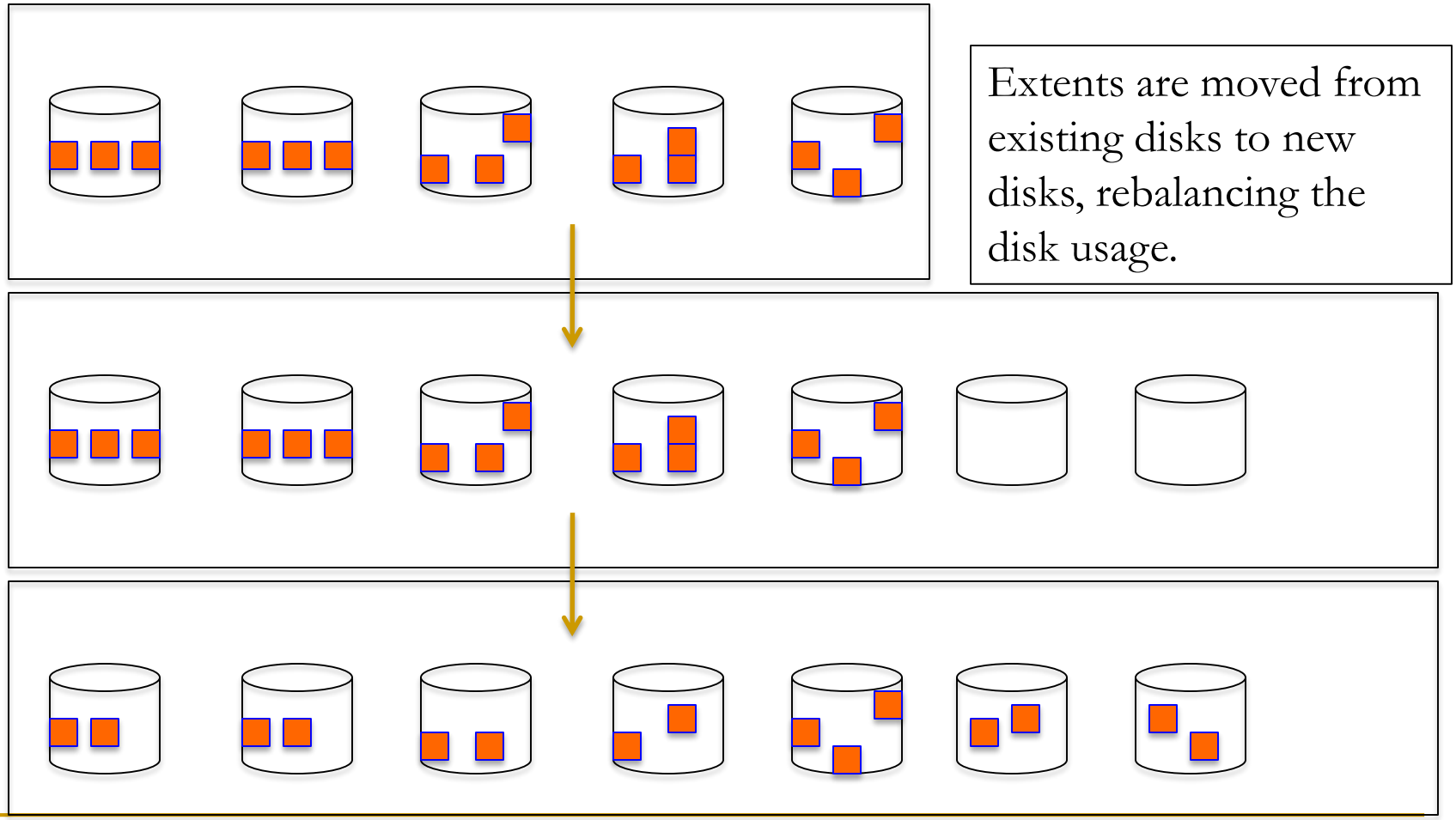
- User defined files are simply alias:

```
$ asmcmd ls -lt  '+DATA/app/oracle/ts_small_02.dbf'
Type      Redund  Striped  Time             Sys  Name
                                            N    ts_small_02.dbf => +DATA/SOLRAC/DATAFILE/TS_SMALL.280.764616401
```

Demo:add_directory, al_ts_small, drop_directory, drop tablespace,

Rebalance

- Addition or deletion of asm disk from a disk group will trigger a rebalance operation.



Extents are moved from existing disks to new disks, rebalancing the disk usage.

Processing details

- RBAL is triggered when there is addition/deletion/resize of disks.
- RBAL acts as a co-ordinator process, updates metadata that ASM rebalance is underway.
- Determines the extent to move and the target disk. Hands off the work to ARBx process.
- ARBx process moves the extent and replies back to RBAL after the successful completion.
- This goes on until RBAL completes the rebalance operation.

Asm_power_limit

- Asm_power_limit controls the speed of rebalance operation.
- This parameter controls number of ARBx process performing the rebalance operation.
- Each ARBx process locks just one extent at a time and moves the extent to another disk.
- You can increase asm_power_limit parameter to improve rebalance operation speed.
- It is not uncommon to disable the rebalance during busy hours and increase the limit to higher value during off hours.

Rebalance miscellaneous

- Rebalance operation can be manually controlled using 'alter diskgroup .. Rebalance';
- Command will return immediately, rebalance will continue asynchronously.
- You could specify wait keyword to wait for the rebalance to complete.
- Only one disk group can participate in a rebalance activity at any point in time, in an ASM instance.
- Potentially, rebalance can be triggered in another disk group from a different node.

Adding disks

- If you are adding many disks, then rebalance only after adding all disks.
- For example, to add 10 disks to an disk group, then:

  (i) Disable asm_power_limit by setting that to 0 with sid='*'.

  (ii) Add all the disks as you wish.

  (iii)Enable asm_power_limit to, say 5, during non-busy hours and let the rebalance work.
- V$asm_operation can be used to monitor rebalance activity.
- In RAC, only one node will be performing the rebalancing activity. Improved in 12c.

Migrating from one disk array to another..

- Same principle applies if you are migrating from one disk array to another.
- High level tasks in the case will be:

  (i) Set asm_power_limit to 0

  (ii) Add the disks from the new array.

  (iii) Drop the disks from the old array. (Dropping simply marks them to drop and you can't drop until rebalancing completes).

  (iv) At this point, set asm_power_limit to 5 or 10, let the rebalance completes the move of extents.

  (v) After the rebalance you can remove the luns.

V$ views

- Access to v$asm_disk will do a discovery of disks and could be costly.
- Oracle provides v$asm_disk_stat to do performance measurement.
- V$asm_disk_stat does not do discovery and so much lighter to access.
- V$asm_diskgroup_stat does not asm diskgroup discovery.

asmcmd

- Asmcmd provides an interface similar to a file system interface.
- Type 'asmcmd help' to see the syntax. Command 'asmcmd –p' to see the current path in asmcmd.
- In a logical view, ASM Disk groups are the root directories:

```
$ asmcmd
ASMCMD> ls -lt +data/
Type  Redund  Striped  Time             Sys  Name
                                        Y    solrac-cluster/
                                        Y    SOLRAC/
```

- Many flags in the ls are supported here too. Use ls –ls to see the size of files.

Demo: asmcmd commands, help etc

Asmcmd cd

- cd is also supported.

```
ASMCMD> cd +data
ASMCMD> ls -lt
Type  Redund  Striped  Time             Sys  Name
                                        Y    solrac-cluster/
                                        Y    SOLRAC/

ASMCMD> cd  SOLRAC
ASMCMD> ls -lt
Type           Redund  Striped  Time             Sys  Name
                                                 Y    TEMPFILE/
                                                 Y    PARAMETERFILE/
                                                 Y    ONLINELOG/
                                                 Y    DATAFILE/
                                                 Y    CONTROLFILE/
                                                 N    spfilesolrac.ora =>
   +DATA/SOLRAC/PARAMETERFILE/spfile.273.731450101
```

Demo: asmcmd commands

Asmcmd lsdg

- Other disk group levels commands are also available.

```
ASMCMD> lsdg
State    Type    Rebal  Sector  Block       AU  Total_MB  Free_MB  Req_mir_free_MB  Usable_file_MB
    Offline_disks  Voting_files  Name
MOUNTED  EXTERN  N         512   4096  1048576     16000     4989                0            4989
    0             Y  DATA/
ASMCMD> lsof
DB_Name  Instance_Name  Path
+ASM     +ASM2          +data.255.4294967295
solrac   solrac1        +data/solrac/controlfile/current.260.731449169
solrac   solrac1        +data/solrac/controlfile/current.261.731449167
solrac   solrac1        +data/solrac/datafile/example.267.731449265
solrac   solrac1        +data/solrac/datafile/sysaux.257.731448845
solrac   solrac1        +data/solrac/datafile/system.256.731448831
```

Demo: asmcmd commands: lsct, lsdg, lsof, du, iostat etc

Asmcmd find

- Find is also available, very similar to UNIX find utility.

```
ASMCMD> find --type DATAFILE +DATA *
+DATA/SOLRAC/DATAFILE/EXAMPLE.267.731449265
+DATA/SOLRAC/DATAFILE/SYSAUX.257.731448845
+DATA/SOLRAC/DATAFILE/SYSTEM.256.731448831
+DATA/SOLRAC/DATAFILE/SYSTEM.275.732461065
+DATA/SOLRAC/DATAFILE/TS_LMT_HW.277.758566281
+DATA/SOLRAC/DATAFILE/UNDOTBS1.258.731448847
+DATA/SOLRAC/DATAFILE/UNDOTBS2.268.731449831
…
ASMCMD> find --type CONTROLFILE +DATA *
+DATA/SOLRAC/CONTROLFILE/Current.260.731449169
+DATA/SOLRAC/CONTROLFILE/Current.261.731449167
..
SMCMD> find --type DATAFILE +DATA UNDO*
+DATA/SOLRAC/DATAFILE/UNDOTBS1.258.731448847
+DATA/SOLRAC/DATAFILE/UNDOTBS2.268.731449831
```

Demo: asmcmd commands: lsct, lsdg, lsof, du, iostat etc

Asmcmd cp

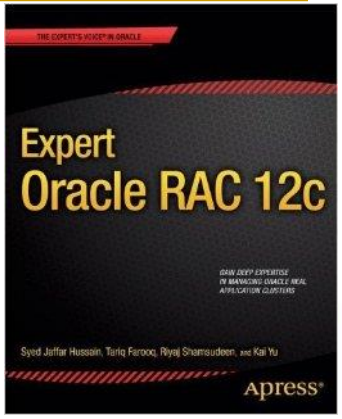
- Cp is another tool to copy from one asm-> asm or asm-> file system.

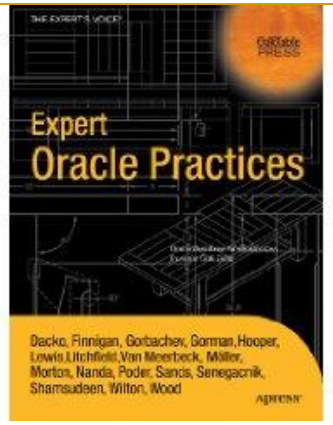
```
ASMCMD> cp SYSTEM.275.732461065 /tmp/
copying +DATA/SOLRAC/DATAFILE/SYSTEM.275.732461065 -> /tmp//SYSTEM.275.732461065
```

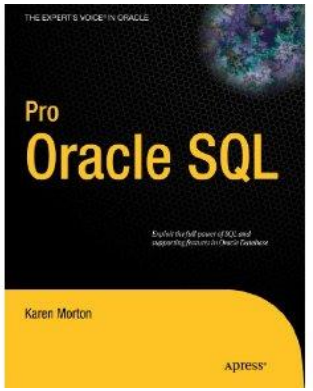
- You can also copy the file to a compressed pipe and transmit to a different server.
- ASM instance should be up and running in both sides for ASM to ASM copy to work.
- Asmcmd cp does not co-ordinate with the database. So, you should alter the database /tablespace to backup mode before cp operation.

Demo: asmcmd cp

THANK YOU







- *Email: rshamsud@orainternals.com*
- *Blog : orainternals.wordpress.com*
- *Web: www.orainternals.com*