A Deep Dive into ASM redundancy in Exadata

Emre Baransel
DOAG 2016, Nurnberg

Overview

As you probably know exadata boxes used to have several options like full, half, quarter and eighth.

A full box had 14 storage servers, half box had 7, quarter and eighth boxes had 3 storage servers.

Now it's possible to choose the number of storage servers and database servers in a box. You can have from 3 to 18 Storage Servers per rack

We also call them cell servers.

Minimum configuration has three cell servers, because each cell server is a failure group in ASM and high redundacy diskgroups require at least 3 failure groups.

We'll consider 3 storage servers in examples.

Except the Extreme Flash option which is an all flash Exadata Machine, every storage server in Exadata physically have 12 equal size and equal performance disks on them. For Exadata X6-2 these disks are 8 TB SAS disks. This is the High Capacity configuration of the Exadata machine.

There used to be an option to chose high performance or high capacity hard disks in an Exadata instalation.

But our point here is whatever option we used to choose, all the hard disks in the Exadata was the same type.

Exadata machine also has Flash Accelerator cards with flash memory but flash is almost always configured as a flash cache (Exadata Smart Flash Cache) in front of disk storage.

We'll also consider a configuration where harddisks are used as ASM disks and flashdisks are used for flashcache.

In Exadata language we call each of these disks 'Physical disk'.

In this case we have total 36 physical disks on our Exadata.

On the installation of Exadata we create logical partitions on the physical disks. Then use the same sized partitions to create ASM diskgroups.

Look at the green boxes. These are the system partitions where storage server operating system resides. In every storage server first two disks have the system partition for redundancy.

As a best practice, on the other disks of the cell servers, we create same sized logical partitions with system partition. You can see them with yellow color. And use them to create DBFS diskgroup to provide filesystem area and to keep clusterware files.

Then we create two more logical partitions on every disk and create RECO and DATA diskgroups. You can see the blue logical partitions build the RECO diskgroup and RED logical partitions build the DATA diskgroup.

We left the DATA diskgroup at the end of the disk, because partitions start from inner edge to outer and the outer edge of the disk provides more performance than the inner edges.

We call these logical partitions Grid Disks. Exadata Grid disks form the basis of ASM. ASM uses these disks in diskgroups so we also call them ASM disks.

Then we start to put data on diskgroups. ASM stripes files accross all the disks within the diskgroup. So we can see a uniform distribution of data in ASM diskgroups.

In Exadata, we can chose normal or high redundancy diskgroups, external redundancy is not supported.

In a normal redundancy diskgroup a primary extent has one mirrored extent (secondary extent). When reading a block, database reads from the primary extent and when writing, writes to both.

And if one of the extents is corrupted, ASM can fix it by using the healty extent. Which is called ‘bad block remapping’

In a high redundancy diskgroup a primary extent has two mirrored extents in different failgroups. In this case where there are 3 storage servers, each server has a copy of extent.

Failure

Now it’s time to investigate failures.

We should look into two types of failures: disk failures and cell server failures.

A disk failure can be transient which can be caused by disk path malfunctions, such as cable failures, host bus adapter failures, controller failures, or disk power supply interruptions. So the disk is actually healty but ASM cannot make I/O to the disk.

Other disk failure type is physical where disk itself has hardware errors.

In a transient disk failure, exadata takes the physical disk into OFFLINE status and grid disks on it will be marked as INACTIVE.

Then time starts ticking.

If the failure is corrected or the disk is replaced before DISK_REPAIR_TIME value exceeds,

Then the grid disks are resynced with ASM FAST MIRROR RESYNC feature. This is an ASM 11g feature andis very useful so that no complete rebalance of failed disk is required.

ASM FAST MIRROR RESYNC keeps track of pending changes to extents on an OFFLINE disk during an outage. The extents are resynced when the disk is brought back online.

But if we don't repair the failure or don't replace the disk, before DISK_REPAIR_TIME, then asm drops the disks from diskgroups and rebalance the data in order to provide redundancy again.

So in a transient failure, the value of DISK_REPAIR_TIME is very important.

It's also very important when we manually offline disks or shutdown storage servers.

Our maintenance task must finish before DISK_REPAIR_TIME in order to prevent an unnecessary rebalace.

We should remember that DISK_REPAIR_TIME is a diskgroup attribute and its default is 3.6 hours.

We can change the default with alter diskgroup command but changing it after a failure has no effect on offline disks.

In a physical disk failure case the story is different.

If a physicaldisk fails, if the disk itself has hardware failure, all griddisks on that physicaldisk will be immediately DROPPED with FORCE option from ASM. Which is called Pro-Active Disk Quarantine

ASM will not wait DISK_REPAIR_TIME to drop disks in this case.

After dropping disks, rebalance will start on diskgroups if there's enough disk space on diskgroups.

Then when we replace the disk, celldisk and griddisks will be recreated automatically and those griddisks will be automatically added to ASM.

Then a new rebalance will start automatically and write extents to new disks.

These automatic operations will occur if the griddisk was previously dropped by the auto disk managment.

If the griddisk was manually dropped, we will need to add disks back into ASM manually .

If we have a closer look into Auto Disk Management feature in Exadata, we see two processes working for it on ASM instances.

The 'Exadata Automation Manager' and 'Exadata Automation Worker'.

Manager process initiates the automation tasks and Worker performs the tasks.

And there are some hidden parameters which controls this feature.

_AUTO_MANAGE_EXADATA_DISKS can enable and disable the feature.

_AUTO_MANAGE_NUM_TRIES controls the maximum number of attempts to perform an automatic operation.

_AUTO_MANAGE_MAX_ONLINE_TRIES controls maximum number of attempts to ONLINE a disk.

These parameters are hidden and you know needs Oracle support advise.

Now let's look into storage server failures. A storage server can fail for some reason and ASM continues working in this case both in normal and high redundancy.

High redundancy can handle two storage servers.

A storage server failure is actually the failure of the whole failgroup in asm.

In this case ASM does not drop disks and it waits for disk_repair_time.

remember this behaviour is also same when rebooting the storage server

If the fault is corrected and server is alive before DISK_REPAIR_TIME, then grid disks will be synched and there will be no rebalance operation.

But if the server is not alive after DISK_REPAIR_TIME, ASM will drop grid disks from diskgroups and a rebalance will start to build redundancy.

The rebalance will start if there is enough space in the diskgroups.

Here normal redundancy only guarantees that system will be alive after one cell server failure, building the redundancy again is possible only if there's enough disk space.

We'll see how to calculate the required space to build reduncy after a cell server fault in next slides.

When the fault is corrected and cell server comes back, there will be a second rebalance and data will be distributed back to all disks.

Second Failure

First failure doesn't cause an ASM service loss in Exadata because there is normal or high redundancy. Now let's look what happens at second failure.

In a normal redundancy diskgroup what happens at second failure is first related with when it occurs.

If it happens after rebalance or sync is completed, then procedure is same with the first failure. It'll be like a first failure case.

But if it happens before the rebalance or sync is completed, then what happens is related with which disk is failed.

If first & second failed disks are not partner disks, which means they don't store two copies of an extent, then a new rebalance will occur, of course if there's enough space.

But if first & second failed disks are partners then ASM will stop and failure will result with data loss.

Ok, this is a small possibility but we have to consider especially depending on the value of the data.

Partner disks are on different storage servers, which means if the second failure is in the same cell, there's no data loss.

Also we say first failure, second failure but first incident doesn't have to be a failure, storage server reboot causes the same, if the sync operation not yet finished.

In some cases it's useful to find out the partner disks of a specific ASM disk. In this MOS note two methods are given for this purpose. It's possible to use X$KFDPARTNER table joined with v$asm_disk view if the diskgroup is mounted. If diskroup is not mounted it's possible to use GMON trace which is on the background_dump_dest of ASM Instance. A shell command with grep and sed is given to extract the partnership information from the trace file.

In High Redundancy we know there are three copies of each extent. So we can say that a second failure never cause a data loss in High Redundancy

Here i want to mention a feature which came with 11.2.0.4 BP16 and 12.1.0.2 BP4. The feature is 'MOUNT RESTRICTED FORCE FOR RECOVERY' and it's applicable to normal redundancy. This feature is useful in the following cases.

First, during an Exadata cell rolling upgrade/patching or any other maintenance work, one of the cell server is not in use and a partner disk failure occurs.

Or a transient disk failure in a cell followed by a permanent partner disk failure before the first failed disk comes back online.

How we use this feature? Let's look into an example.

First we shutdown all services on first cell and ASM takes these disks offline. Just like we're doing maitenance on the cell.

Then simulate a disk failure in second cell with this alter physicaldisk command.

Then ASM stops and database crashes.

Now we mount the diskgroup with 'restricted force for recovery' option.

We start services on first cell and make the disks online with this command. Now a SYNC operation will start to update modified extents in the first cell. We must wait until SYNC finishes. In this resync the process will not be able to read some of the required extents because of the second disk failure and mark them with BADFDA7A.

We'll see this in rebalance process trace file.

Now we have a mounted diskgroup with some extents corrupted.

We can remount the diskgroup and perform RMAN block recovery to fix those corruptions.

Usable Space

Now we'll look at the 'usable space' expression on an Exadata diskgroup.

Here the question is what kind of usable space.

What a person wants to express with the term 'usable space' in Exadata is generally different from what people understand.

I listed some disk size terms in Exadata to make the 'usable space' expression clear.

Total Raw Size (TRS)

Used Raw Size (URS)

Free Raw Size (FRS)

Total Allocatable Size (TAS)	➔	**TRS / Redundancy Factor**
Used Allocatable Size (UAS)	➔	**URS / Redundancy Factor**
Free Allocatable Size (FAS)	➔	**FRS / Redundancy Factor**
Size Needed for Disk Failure Coverage (SNDFC)	➔	**Largest Disk (or 2 Disks for High R.)**
Size Needed for Cell Failure Coverage (SNCFC)	➔	**Largest Cell (or 2 Cells for High R.)**
Total Disk Failure Safe Allocatable Size	➔	**(TRS - SNDFC) / Redundancy Factor**
Total Cell Failure Safe Allocatable Size	➔	**(TRS - SNCFC) / Redundancy Factor**
Free Disk Failure Safe Allocatable Size	➔	**(FRS - SNDFC) / Redundancy Factor**
Free Cell Failure Safe Allocatable Size	➔	**(FRS - SNCFC) / Redundancy Factor**

Let’s make an example with a total raw size of 360. Let’s say used raw size is 120 and free raw size is 240.

If we don’t care about building the redundancy after a failure then we can use 180.

If we want ASM to be available to build redundancy after a disk failure, we should not use more 175 at total.

If we want ASM to be available to build redundancy after a cell failure, we should not use more 120 at total.

	Normal Redundancy	High Redundancy
Total Raw Size (TRS)	**360**	**360**
Used Raw Size (URS)	**120**	**120**
Free Raw Size (FRS)	**240**	**240**
Total Allocatable Size (TAS)	**TRS / 2 = 180**	**TRS / 3 = 120**
Used Allocatable Size (UAS)	**URS / 2 = 60**	**URS / 3 = 40**
Free Allocatable Size (FAS)	**FRS / 2 = 120**	**FRS / 3 = 80**
Size Needed for Disk Failure Coverage (SNDFC)	**10**	**20**
Size Needed for Cell Failure Coverage (SNCFC)	**120**	**240**
Total Disk Failure Safe Allocatable Size	**(TRS - SNDFC) / 2 = 175**	**(TRS - SNDFC) / 3 = 113.3**
Total Cell Failure Safe Allocatable Size	**(TRS - SNCFC) / 2 = 120**	**N/A for Quarter Rack**
Free Disk Failure Safe Allocatable Size	**(FRS - SNDFC) / 2 = 115**	**(FRS - SNDFC) / 3 = 73.3**
Free Cell Failure Safe Allocatable Size	**(FRS - SNCFC) / 2 = 60**	**N/A for Quarter Rack**

In the ASMCMD lsdg output there are information like redundancy type, rebalance status, allocation unit etc. What’s important for us in terms of usable space is in the yellow background. Total_mb, free_mb, req_mir_free_mb and usable_file_mb. Let’s look what they mean.

Total_mb is what we said as ‘total raw size’ and free_mb is ‘free raw size’

What Req_mir_free_mb means changes depending on the version. With and after 11.2.0.4.9 and 12.1.0.2 it means ‘Size Needed for Disk Failure Coverage’. Before these versions it means ‘Size Needed for Cell Failure Coverage’.

In the same way, what usable_file_mb means also changed. It’s ‘Free Disk Failure Safe Allocatable Size’ in the new versions and ‘Free Cell Failure Safe Allocatable Size’ in the older versions.

Oracle changed the calculation from cell to disk because usable_file_mb becomes negative much earlier when calculating cell failure. And this leads to some misunderstandings.

What to Pay Attention to

- **Replacing Disks**
    - Before replacement,

  Be sure that rebalance is completed and disk is ready for replacement

    - After replacement,

  Be sure that cell/grid disks are created and rebalance started

- **Rolling Shutdown/Upgrade of Storage Servers**
    - Check DISK_REPAIR_TIME and increase if required
    - Check if ASM will be OK if the grid disks go OFFLINE
    - Be sure that there’s enough space for rebalance if something goes wrong

  (Free Cell Failure Safe Allocatable Size)