ASM Metadata and Internals

A collection of facts on configuration and and diagnostic of Oracle ASM. More on RAC and ASM configuration and performance of CERN Physics DBs in Inside_Oracle_ASM_LC_CERN_UKOUG07.ppt and in HAandPerf.

ASM metadata, V$ and X$:

View Name	X$ Table name	Description
V$ASM_DISKGROUP	X$KFGRP	performs disk discovery and lists diskgroups
V$ASM_DISKGROUP_STAT	X$KFGRP_STAT	diskgroup stats without disk discovery
V$ASM_DISK	X$KFDSK, X$KFKID	performs disk discovery, lists disks and their usage metrics
V$ASM_DISK_STAT	X$KFDSK_STAT, X$KFKID	lists disks and their usage metrics
V$ASM_FILE	X$KFFIL	lists ASM files, including metadata/asmdisk files
V$ASM_ALIAS	X$KFALS	lists ASM aliases, files and directories
V$ASM_TEMPLATE	X$KFTMTA	lists the available templates and their properties
V$ASM_CLIENT	X$KFNCL	lists DB instances connected to ASM
V$ASM_OPERATION	X$KFGMG	lists rebalancing operations
N.A.	X$KFKLIB	available libraries, includes asmlib path
N.A.	X$KFDPARTNER	lists disk-to-partner relationships
N.A.	X$KFFXP	extent map table for all ASM files
N.A.	X$KFDAT	extent list for all ASM disks
N.A.	X$KFBH	describes the ASM cache (buffer cache of ASM in blocks of 4K (_asm_blksize)
N.A.	X$KFCCE	a linked list of ASM blocks. to be further investigated

This list is obtained querying v$fixed_view_definition where view_name like '%ASM%' which exposes all the v$ and gv$ views with their definition. Fixed tables are exposed by querying v$fixed_table where name like 'x$kf%' (ASM fixed tables use the 'X$KF' prefix). Note on 11g there are additional V$views: , and X$tables: *

New in 11g:

View Name	X$ Table name	Description
V$ASM_ATTRIBUTE	X$KFENV	ASM attributes, the X$ table shows also 'hidden' attributes
V$ASM_DISK_IOSTAT	X$KFNSDSKIOST	I/O statistics
N.A.	X$KFDFS	
N.A.	X$KFDDD	
N.A.	X$KFGBRB	
N.A.	X$KFMDGRP	
N.A.	X$KFCLLE	
N.A.	X$KFVOL	
N.A.	X$KFVOLSTAT	
N.A.	X$KFVOFS	
N.A.	X$KFVOFSV	

Striping and Mirroring with ASM, extents and allocation units

A basic example, using ASM and normal redundancy: the available storage, say 64 HDs over FC SAN, are used to create the main DB diskgroup: DATADG. DATADG is logically divided into 2 evenly sized groups of disks: 32 disks in failgroup N.1 and 32 in failgroup N.2. Oracle datafiles created in DATADG are 'striped' into smaller pieces, extents of 1MB in size. Extents are allocated to the storage in 2 (mirrored) allocation units (AU): one AU in failgroup N.1 the other in failgroup N.2.

X$KFFXP

This X$ table contains the mapping between files, extents and allocation units. It allows to track the position of all the extents of a given file striped and mirrored across storage. Note: RDBMS read operations access only the primary extent of a mirrored couple (unless there is an IO error) . Write operations instead write all mirrored extents to disk.

X$KFFXP Column Name	Description
ADDR	x$ table address/identifier
INDX	row unique identifier
INST_ID	instance number (RAC)
NUMBER_KFFXP	ASM file number. Join with v$asm_file and v$asm_alias
COMPOUND_KFFXP	File identifier. Join with compound_index in v$asm_file
INCARN_KFFXP	File incarnation id. Join with incarnation in v$asm_file
PXN_KFFXP	Progressive file extent number
XNUM_KFFXP	ASM file extent number (mirrored extent pairs have the same extent value)
GROUP_KFFXP	ASM disk group number. Join with v$asm_disk and v$asm_diskgroup
DISK_KFFXP	Disk number where the extent is allocated. Join with v$asm_disk
AU_KFFXP	Relative position of the allocation unit from the beginning of the disk. The allocation unit size (1 MB) in v$asm_diskgroup
LXN_KFFXP	0->primary extent, ->mirror extent, 2->2nd mirror copy (high redundancy and metadata)
FLAGS_KFFXP	N.K.
CHK_KFFXP	N.K.
SIZE_KFFXP	**11g**, to support variable size AU, integer value which marks the size of the extent in AU size units.

Example1 - reading ASM files with direct OS access

- Find the 2 mirrored extents of an ASM file (the spfile in this example)

```
sys@+ASM1> select GROUP_KFFXP,DISK_KFFXP,AU_KFFXP from x$kffxp where
   number_kffxp=(select file_number from v$asm_alias where name='spfiletest1.ora');

GROUP_KFFXP DISK_KFFXP   AU_KFFXP
----------- ---------- ----------
          1         20        379
          1          3         101
```

- find the diskname

```
sys@+ASM1> select disk_number,path from v$asm_disk where
    GROUP_NUMBER=1 and disk_number in  (3,20);

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

```
          3    /dev/mpath/itstor417_2p1
         20   /dev/mpath/itstor419_2p1
```

- access the data directly from disk with dd

```
dd if=/dev/mpath/itstor417_2p1 bs=1024k count=1 skip=101|strings|more
```

See also:

- https://twiki.cern.ch/twiki/pub/PSSGroup/HAandPerf/ASM_metadata_30012006.html
- http://www.freelists.org/archives/oracle-l/05-2006/msg00395.html

X$KFDAT

This X$ table contains details of **all allocation units** (free and used).

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

Example2 - list allocation units of a given file from x$kfdat

- similarly to example 1 above, another way to retrieve ASM file allocation maps:

```
sys@+ASM1> select GROUP_KFDAT,NUMBER_KFDAT,AUNUM_KFDAT from x$kfdat where
   fnum_kfdat=(select file_number from v$asm_alias where name='spfiletest1.ora');

GROUP_KFDAT NUMBER_KFDAT AUNUM_KFDAT
----------- ------------ -----------
          1            3         101
          1           20         379
```

Example3 - from strace data of an oracle user process

- from the strace file of a user (shadow) process identify IO operations:
  - ex: `strace -p 30094 2>&1|grep -v time`
  - read64(15, "#\242\0\0\33\0@\2\343\332\177\303s\5\1\4\211\330\0\0\0"..., 8192, 473128960) = 8192
  - it is a read operation of 8KB (oracle block) at the offset 473128960 (=451 MB + 27*8KB) from file descriptor FD=15
- using /proc/30094/fd -> find FD=15 is /dev/mpath/itstor420_1p1
- I find the group and disk number of the file:

```
sys@+ASM1> select GROUP_NUMBER,DISK_NUMBER from v$asm_disk
```

```
where path='/dev/mpath/itstor420_1p1';

GROUP_NUMBER DISK_NUMBER
------------ -----------
           1          30
```

- using the disk number, group number and offset (from strace above) I find the file number and extent number:

```
sys@+ASM1> select number_kffxp, XNUM_KFFXP from x$kffxp where group_kffxp=1 and disk_kffxp=20 and

NUMBER_KFFXP XNUM_KFFXP
------------ ----------
       268          17
```

- from v$asm_file fnum=268 is file of the users' tablesspace:

```
sys@+ASM1> select name from v$asm_alias where FILE_NUMBER=268

NAME
------------------------------
USERS.268.612033477

sys@DB> select file#,name from v$datafile where upper(name) like '%USERS.268.612033477';

     FILE# NAME
---------- ----------------------------------------------------------
         9 +TEST1_DATADG1/test1/datafile/users.268.612033477
```

- from dba extents finally find the owner and segment name relative to the original IO operation:

```
sys@TEST1> select owner,segment_name,segment_type from dba_extents
where FILE_ID=9 and 27+17*1024*1024 between block_id and block_id+blocks;

OWNER                          SEGMENT_NAME                   SEGMENT_TYPE
------------------------------ ------------------------------ ------------------
SCOTT                          EMP                            TABLE
```

X$KFDPARTNER

This X$ table contains the disk-to-partner (1-N) relationship. Two disks of a given ASM diskgroup are partners if they each contain a mirror copy of the same extent. Therefore partners must belong to different failgroups of the same diskgroup. From a few live examples I can see that **typically disks have 10 partners each** at diskgroup creation and fluctuate around 10 partners following ASM operations. This mechanism is in place to reduce the chance of losing both sides of the mirror in case of double disk failure.

X$KFDPARTNER Column Name	Description
ADDR	x$ table address/identifier
INDX	row unique identifier
INST_ID	instance number (RAC)
GRP	diskgroup number, join with v$asm_diskgroup
DISK	disk number, join with v$asm_disk
COMPOUND	disk identifier. Join with compound_index in v$asm_disk
NUMBER_KFDPARTNER	partner disk number, i.e. disk-to-partner (1-N) relationship
MIRROR_KFDPARNER	=1 in a healthy normal redundancy config
PARITY_KFDPARNER	=1 in a healthy normal redundancy config
ACTIVE_KFDPARNER	=1 in a healthy normal redundancy config

X$KFFIL and metadata files

Three types of metadata:

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

Example: list all files, system and users' with their sizes:

- `select group_kffil group#, number_kffil file#, filsiz_kffil filesize_after_mirr, filspc_kffil raw_file_size from x$kffil;`

Example: List all files including metadata allocated in the ASM diskgroups

- `select group_kfdat group#,FNUM_KFDAT file#, sum(1) AU_used from x$kfdat where v_kfdat='V' group by group_kfdat,FNUM_KFDAT,v_kfdat;`

Description of metadata files

This paragraph is from: Oracle Automatic Storage Management, Oracle Press Nov 2007, N. Vengurlekar, M. Vallath, R.Long

- File#0, AU=0: disk header (disk name, etc), Allocation Table (AT) and Free Space Table (FST)
- File#0, AU=1: Partner Status Table (PST)
- File#1: File Directory (files and their extent pointers)
- File#2: Disk Directory
- File#3: Active Change Directory (ACD) The ACD is analogous to a redo log, where changes to the metadata are logged. Size=42MB * number of instances
- File#4: Continuing Operation Directory (COD). The COD is analogous to an undo tablespace. It maintains the state of active ASM operations such as disk or datafile drop/add. The COD log record is either committed or rolled back based on the success of the operation.
- File#5: Template directory
- File#6: Alias directory
- 11g, File#9: Attribute Directory
- 11g, File#12: Staleness registry, created when needed to track offline disks

Tnsnames entries and ASM

TIP: An example of tnsnames entry to be used to connect to ASM instances via Oracle*NET (note the extra keyword (UR=A)). More generally UR=A allows to connect to 'blocked services'. Example connect sys/pass@ASM1 as sysdba (an asm password file is also needed on the server). The extra keyword (UR=A) applies to 10g, it is not needed in 11g.

```
ASM1 =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = [hostname])(PORT = [portN]))
    (CONNECT_DATA =
      (SERVER = DEDICATED) (SERVICE_NAME = +ASM)   (INSTANCE_NAME = +ASM1)
      (UR=A)
```

```
    )  )
```

DBMS_DISKGROUP, an internal ASM package

dbms_diskgroup is an Oracle 'internal package' (it doesn't show up as an object being that ASM has no dictionary) called dbms_diskgroup. It is used to access the ASM with filesystem-like calls. 11g asmcmd uses this package to implement the cp command. A list of procedures:

dbms_diskgroup.open(:fileName, :openMode, :fileType, :blkSz, :hdl,:plkSz, :fileSz)
dbms_diskgroup.createfile(:fileName, :fileType, :blkSz, :fileSz, :hdl, :plkSz, :fileGenName)
dbms_diskgroup.close(:hdl)
dbms_diskgroup.read(:hdl, :offset, :blkSz, :data_buf)
dbms_diskgroup.commitfile(:handle)
dbms_diskgroup.resizefile(:handle,:fsz)
dbms_diskgroup.remap(:gnum, :fnum, :virt_extent_num)
dbms_diskgroup.getfileattr(:fileName, :fileType, :fileSz, :blkSz)
dbms_diskgroup.checkfile(?)
dbms_diskgroup.patchfile(?)

ASM parameters and underscore parameters

```
select a.ksppinm "Parameter", c.ksppstvl "Instance Value"
  from x$ksppi a, x$ksppcv b, x$ksppsv c
 where a.indx = b.indx and a.indx = c.indx
   and ksppinm like '%asm%'
order by a.ksppinm;
```

Parameter Name	Value
_asm_acd_chunks	1
_asm_allow_only_raw_disks	TRUE
_asm_allow_resilver_corruption	FALSE
_asm_ausize	1048576
_asm_blksize	4096
_asm_disk_repair_time	14400
_asm_droptimeout	60
_asm_emulmax	10000
_asm_emultimeout	0
_asm_kfdpevent	0
_asm_libraries	ufs (may differ if asmlib is used)
_asm_maxio	1048576
_asm_stripesize	131072
_asm_stripewidth	8
_asm_wait_time	18
_asmlib_test	0
_asmsid	asm
asm_diskgroups	list of diskgroups to be mounted at startup
asm_diskstring	search path for physical disks to be used with ASM
asm_power_limit	default rebalance power value

new in 11g:

Parameter Name	Value
_asm_compatibility	10.1

_asm_dbmsdg_nohdrchk	FALSE
_asm_droptimeout	removed in 11g
_asm_kfioevent	0
_asm_repairquantum	60
_asm_runtime_capability_volume_support	FALSE
_asm_skip_resize_check	FALSE
_lm_asm_enq_hashing	TRUE
asm_preferred_read_failure_groups	

ASM-related acronyms

- **PST** - Partner Status Table. Maintains info on disk-to-diskgroup membership.
- **COD** - Continuing Operation Directory. The COD structure maintains the state of active ASM operations or changes, such as disk or datafile drop/add. The COD log record is either committed or rolled back based on the success of the operation. (source Oracle whitepaper)
- **ACD** - Active Change Directory. The ACD is analogous to a redo log, where changes to the metadata are logged. The ACD log record is used to determine point of recovery in the case of ASM operation failures or instance failures. (source Oracle whitepaper)
- **OSM** Oracle Storage Manager, legacy name, synonymous of ASM
- **CSS** Cluster Synchronization Services. Part of Oracle clusterware, mandatory with ASM even in single instance. CSS is used to heartbeat the health of the ASM instances.
- **RBAL** - Oracle backgroud process. In an ASM instance coordinated rebalancing operations. In a DB instance, opens and mount diskgroups from the local ASM instance.
- **ARBx** - Oracle backgroud processes. In an ASM instance, a slave for rebalancing operations
- **PSPx** - Oracle backgroud processes. In an ASM instance, Process Spawners
- **GMON** - Oracle backgroud processes. In an ASM instance, diskgroup monitor.
- **ASMB** - Oracle backgroud process. In an DB instance, keeps a (bequeath) persistent DB connection to the local ASM instance. Provides hearthbeat and ASM statistics. During a diskgroup rebalancing operation ASM communicates to the DB AU changes via this connection.
- **O00x** - Oracle backgroud processes. Slaves used to connected from the DB to the ASM instance for 'short operations'.

Revisions:
Additions and corrections, Nov 2007, L.C.
Added examples, Feb 2007, L.C.
Major additions, Jan 2007, L.C.
V1.0 Jan 2006, Luca.Canali@cernNOSPAMPLEASE.ch

This topic: PSSGroup > ASM_Internals
Topic revision: r22 - 2009-01-28 - LucaCanali