IBM XIV® Storage System

Administrating Oracle Database and ASM in XIV environment

White Paper

May 2009

askxiv@us.ibm.com
www.xivstorage.com

Table of Contents

Introduction

The purpose of this document is to provide guidelines and recommendations on how to configure and Administrate Oracle database on XIV storage system when Oracle Automatic Storage Management (ASM) is used as underplaying volume manager. The intended audience is the XIV technical sales teams and their customers, creating or migrating Oracle database/s with ASM on/to XIV storage systems.

Currently, all procedures in this paper were tested and documented on Linux Red Hat OS. It is our intention to expand this document to cover other OSs (AIX, Windows, HP-UX and Solaris) in the future.

This document assumes an intimate familiarity with the XIV architecture and basic understanding of Oracle database technology. It combines the features and best practices of both technologies and provides recommendations on how to achieve:

- Performance favorable and simplest data layout
- Minimal down time for administrative processes, like data migration and storage allocation
- Optimized backup and recovery processes

ASM Overview

ASM is Oracle's storage management solution introduces with Oracle 10g. It provides an alternative to conventional volume managers, file systems, and raw devices.

ASM uses disk groups to store Oracle datafiles; an ASM disk group is a collection of disks that ASM manages as a unit. Within a disk group, ASM exposes a file system interface for Oracle database files. The content of files that are stored in a disk group is striped across all disks in the disk group.

You can add or remove disks from a disk group while a database continues to access files from the disk group. When you add or remove disks from a disk group, ASM automatically redistributes the file contents and eliminates the need for downtime when redistributing the content.

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

Oracle 10g officially supported only 1MB size AU, although there was a hidden parameter _asm_ausize which enabled larger sizes for AU.

With Oracle 11g, when you create a disk group, you can set the ASM AU size to be between 1 MB and 64 MB in powers of two, such as, 1, 2, 4, 8, 16, 32, or 64.

To distribute data evenly across the disk group, ASM separates files into stripes. There are two types of stripe size supported: coarse and fine. Coarse-grained striping provides load balancing for disk groups while fine-grained striping reduces latency for certain file types by spreading the load more widely.

The coarse-grained stripe size is always equal to the AU size. The fine-grained stripe size always equals 128 KB; this provides lower I/O latency for small I/O operations such as redo log writes.

For more details about Oracle ASM please refer to:
http://download.oracle.com/docs/cd/B28359_01/server.111/b31107.pdf.

Database Layout on ASM in XIV Environment

In general the XIV architecture does not require the use of any logical volumes with host based striping. The data is already striped across all the drives in the IBM XIV Storage System. The recommendation is not to use any logical volumes or any other method that stripes the data unless it is required for other reasons than performance. Striping on the logical volume side might adversely affect XIV caching algorithms, for example a sequential pre-fetch might not occur since the reads from the host will spread across the volumes. This is true for other storage systems as well.

In addition, it might generate unnecessary internal IOs if ASM AUs won't be aligned with XIV's internal striping.

Benchmark description

To come-up with recommendation regarding the preferred AU size and number of volumes, we have conducted a series of tests (see detailed description below) manipulating AU size and number of mapped volumes (preserving same capacity) for each set of tests. We have also executed a test on database using LVM instead of ASM.

The purpose of this benchmark ***is not*** to achieve the optimal performance numbers of Oracle with ASM on IBM XIV storage, but to analyze relative performance characteristics in this environment.

To simplify performance data collection all tests have utilized two disk groups, first for data files and second for logs. This configuration enabled isolation of IO statistics on the data volumes from log volumes. In real customer environment this separation is not necessary, since both data and log files will be automatically distributed and mirrored across all spindles in the XIV system.

Following list describes different test sets used in this benchmark (excluding the log volume):

- 1 vol, no ASM
- 1 vol, ASM, AU size 1M, fine striping (128k stripes)
- 1 vol, ASM, AU size 1M, course striping
- 1 vol, ASM, AU size 8M, course striping
- 1 vol, ASM, AU size 64M, course striping
- 2vol, ASM, AU size 8M, course striping
- 2 vol, ASM, AU size 64M, course striping
- 6 vol, ASM, AU size 8M, course striping

Each test set executed the workloads described below. These workloads represent common IO profiles in any database environment:

- Data load – large sequential writes to populate the database;
- Table scan – large sequential reads across all database tables (no index);
- Index creation – mix of reads and writes;
- Merge – this test scans two large tables (sequential reads) in the database and creates a third table (writes) by merging their content

The following chart summarizes the results of benchmark:

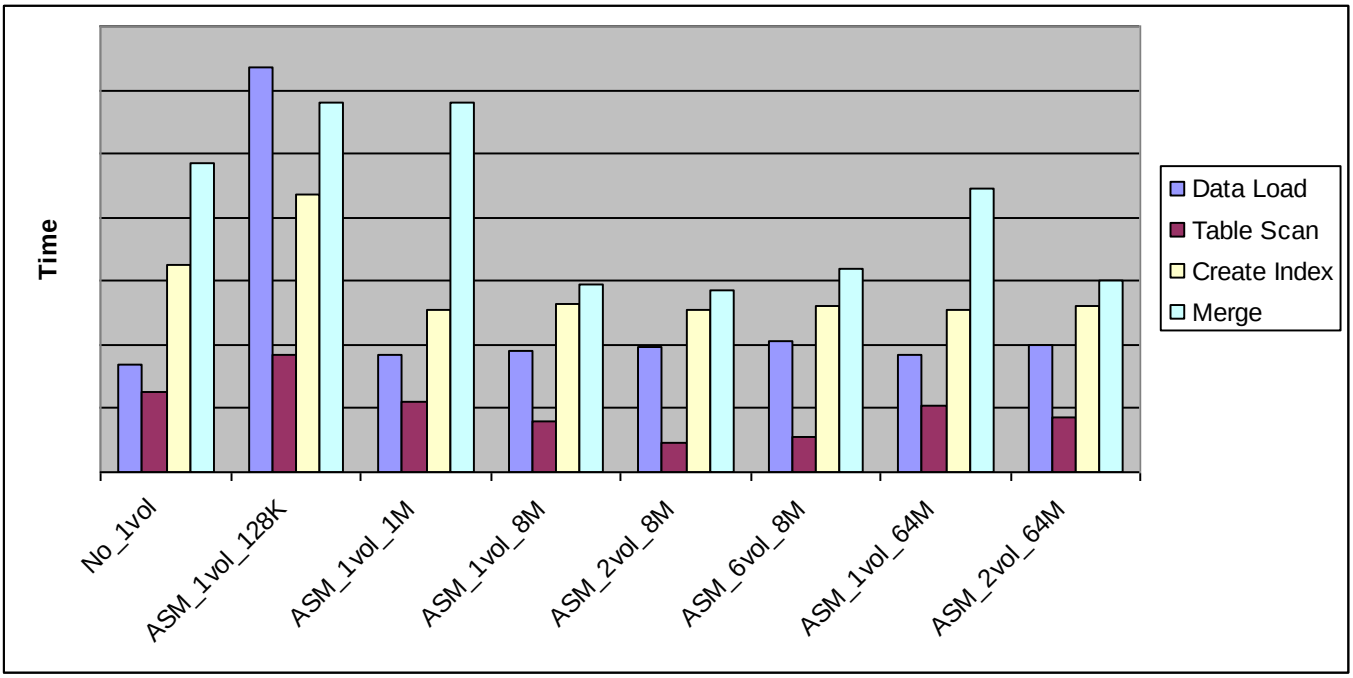

Figure1: Preferred AU size and number of volumes benchmark summary (the shorter bars represent better performance times)

Results summary and conclusion

The test results have shown the following:

- When Oracle ASM is used with AU sizes larger then 1M, Oracle database out-performs the configuration without ASM for most workload types.
- For more complex workloads (workloads that combine reads and writes) the 8M and 16M AU sizes have shown slightly better results then with 64M size.
- Diskgroups with 1 or 2 XIV volumes have shown better results than the one configure with large number of volumes (for example 6 volumes diskgroup in our test).

In conclusion, when configuring Oracle database using ASM on XIV, as a rule of thumb, to achieve better performance and create a configuration that is easy to mange use:

- 1 or 2 XIV volume to create an ASM diskgroup
- Create the diskgroup with 8M or 16M AU size

Migrating to XIV array from Legacy Storage Systems

This section introduces three methods to migrate Oracle database from legacy storage systems to XIV.

1. Using XIV Data Migration Service
2. Using ASM diskgroup mirroring
3. Using Oracle RMAN

Each method has its own advantages and limitations. For example, methods #1 and #3 will require a short downtime to redirect the database to the XIV storage; vs. with method #2 the migration can be done completely transparent to the application, but it requires at least double storage capacity.

Using XIV Data Migration Service - Diskgroup with EXTERNAL, NORMAL or HIGH Redundancy

The following migration procedure utilizes XIV Data Migration service without complex configurations, additional software or capital expenditure. A small outage window is required to redirect the host and the database from legacy storage to XIV. Once the database is redirected to the XIV array, the application can be started, while data migration is continuing in the background. This procedure requires you to preserve the original data layout and volume sizes.

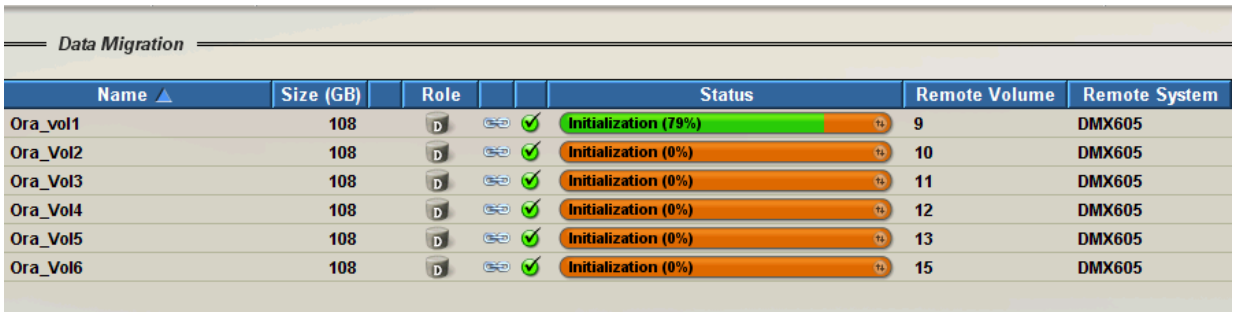

Figure 2: A background migration process from legacy storage to XIV

Step 1: Initial configuration and preparation of the arrays for migration

Following are the high level steps required to prepare the arrays for migration. For detailed process description, please refer to "***Data Migrations with XIV Using XIV Native Migration Utility***" manual.

1. Establish physical connectivity between XIV array and the legacy system either via FC fabric or iSCSI connection.

2. Define XIV to legacy array as Linux host.

3. Define legacy array to XIV

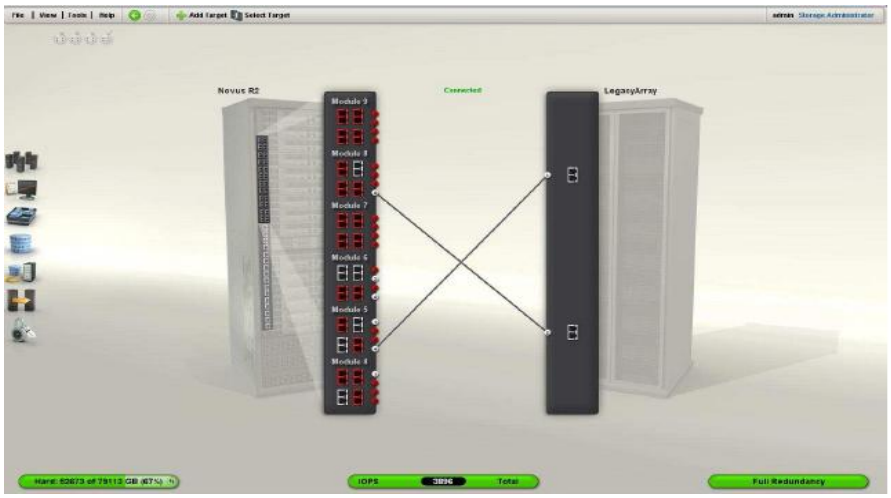

Figure 3: Establishing connectivity between XIV and legacy array

4. Defining the database server to XIV array

Step 2: Preparing the application and database server for migration

```
State    Type    Rebal  Sector  Block       AU  Total_MB  Free_MB  Req_mir_free_MB  Usable_file_MB  Offline_disks  Name
MOUNTED  EXTERN  N         512   4096  1048576    621354   200311                0          200311              0  XIV_TEST/
ASMCMD> lsdsk -k
Total_MB  Free_MB   OS_MB  Name          Failgroup     Library  Label  UDID  Product  Redund   Path
  103559    33384  103559  XIV_TEST_0000  XIV_TEST_0000  System                        UNKNOWN  /dev/emcpowera1
  103559    33386  103559  XIV_TEST_0001  XIV_TEST_0001  System                        UNKNOWN  /dev/emcpowerb1
  103559    33386  103559  XIV_TEST_0002  XIV_TEST_0002  System                        UNKNOWN  /dev/emcpowerc1
  103559    33391  103559  XIV_TEST_0003  XIV_TEST_0003  System                        UNKNOWN  /dev/emcpowerd1
  103559    33382  103559  XIV_TEST_0004  XIV_TEST_0004  System                        UNKNOWN  /dev/emcpowere1
  103559    33382  103559  XIV_TEST_0005  XIV_TEST_0005  System                        UNKNOWN  /dev/emcpowerf1
```

Figure 4: ASM diskgroup configuration on legacy array before the migration

Note: For detailed description of the process bellow, please refer to "***Data Migrations with XIV Using XIV Native Migration Utility***".

1. It is highly recommended to backup a migrated application before actual migration is started.

2. Shutdown Oracle database and the ASM instance on the server

3. Redirect the database server to XIV array (fabric zoning)

4. Remove legacy multipath software

5. Install multipath driver supported by XIV (see SSIC for supported multipath drivers)

6. Install XIV Host Attachment Kit
7. Allocate legacy volumes to XIV (the target XIV volumes can be manually pre-created or automatically allocated.
8. Activate data migration
9. Map XIV volumes to the database server

   As root, using XIV's host connectivity package, rescan SCSI bus to discover XIV LUNs on the database server. For example:

   ```
   /opt/xiv/host_attach/bin/qlogic_lun_scan -s
   ```

   Display the discovered volumes by using the vol_list command (see Figure 5).

```
[root@host244 ~]# /opt/xiv/host_attach/bin/vol_list
Multipathed IBM-XIV 2810 devices
------------------------------------------------------------------------------------------
Device file                                Volume name          Size      Machine   Vol id
------------------------------------------------------------------------------------------
/dev/mapper/mpath19                        asm_6                 34 GB    000010    00533
/dev/mapper/mpath21                        asm_8                 34 GB    000010    00535
/dev/mapper/mpath18                        asm_5                 34 GB    000010    00532
/dev/mapper/mpath20                        asm_7                 34 GB    000010    00534
/dev/mapper/mpath17                        asm_4                 34 GB    000010    00531
/dev/mapper/mpath16                        asm_3                 34 GB    000010    00530
/dev/mapper/mpath15                        asm_2                 34 GB    000010    00529
/dev/mapper/mpath14                        asm_1                 34 GB    000010    00528
/dev/mapper/mpath13                        asm_dump             412 GB    000010    00454

No Non-multipathed IBM-XIV 2810 devices found.
```

Figure 5: `vol_list` command output

Step 3: Verify that ASM can discover the new XIV LUNs as ASM volumes

As root verify that ASMLib identifies all required LUNs run:

```
/etc/init.d/oracleasm scandisks
/etc/init.d/oracleasm listdisks
```

```
[root@host244 ~]# /etc/init.d/oracleasm listdisks
VOL1
VOL2
VOL3
VOL4
VOL5
VOL6
VOL7
VOL8
[root@host244 ~]#
```

Figure 6: `oracleasm listdisks` command output

Change the ownership of the XIV device path to oracle:dba. For example:

```
chown oracle:dba /dev/mapper/mapth18
```

Note: This is one of the most important steps to execute which is usually forgotten. If it is not executed, Oracle ASM won't be able to mount diskgroups on the volumes.

Step 4: Redirect ASM diskgroup to XIV LUNs and startup the database

1. Startup ASM – it will complain about the fact that diskgroup/s cannot be mounted because of missing disks. Ignore it for now.

2. Alter ASM disk discovery string to point to XIV LUNs

   ```
   SQL> ALTER SYSTEM SET ASM_DISKSTRING='/dev/mapper/mpath*'
   ```

3. Mount the diskgroups that used to reside on the legacy system , for example:

   ```
   SQL> ALTER DISKGROUP ora_data MOUNT;
   ```

Step 5: Complete data migration

Once all data is migrated to XIV array (see figure 1), finalize the migration using process described in "***Data Migrations with XIV Using XIV Native Migration Utility*".**

Using ASM to Migrate to XIV Array – Diskgroup with NORMAL or HIGH Redundancy

This example introduces a process of migrating Oracle database installed on ASM with diskgroups defined with NORMAL or HIGH redundancy. It will not work if original volumes are defined with EXTERNAL redundancy or if you wish to migrate to EXTERNAL redundancy configuration.

```
ASMCMD> lsdsk -k -d ASM_TEST
Total_MB  Free_MB  OS_MB  Name           Failgroup   Library  Label  UDID  Product  Redund   Path
   32768    26832  32768  ASM_TEST_0000  FAILGROUP1  System                        UNKNOWN  /dev/mapper/mpath14
   32768    26837  32768  ASM_TEST_0001  FAILGROUP1  System                        UNKNOWN  /dev/mapper/mpath15
   32768    26836  32768  ASM_TEST_0002  FAILGROUP2  System                        UNKNOWN  /dev/mapper/mpath16
   32768    26833  32768  ASM_TEST_0003  FAILGROUP2  System                        UNKNOWN  /dev/mapper/mpath17
ASMCMD>
```

Figure 7: Configuration of ASM_TEST diskgroup with 2 failgroups, 2 disks of similar size in each group. The original volumes can be on any disk array, including XIV.

ASM diskgroup with NORMAL redundancy has to have at least two failgroups, like in the example above. To migrate to XIV array you need to preallocate storage and LUNs enough for at least 2 failgroups.

As a general guideline, Oracle provides the fallowing recommendations to guarantee equal load balancing across all disks in the diskgroup and prevent hot spots:

- All disks in the diskgroup should have the same size
- All failgroups in the diskgroup should have the same number of disks
- All failgroups should have the same total size

During the migration the flowing guidelines should be used:

- The total capacity of target failgroup should be equal to or larger then original failgroup
- All destination failgroups should follow the guidelines above

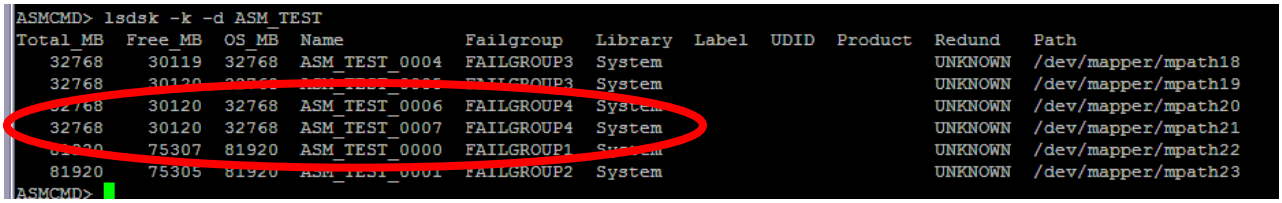

Figure 8: Migrating to failgroups with total storage size larger then the original, with smaller number of disks

In example described in Figure 8, database is migrated from failgroups with total storage size of 64GB and 2 disks in a failgroup, to failgroups with 80GB and 1 disk per failgroup.

Step 1: Allocating LUNs on XIV array

Create LUNs of desired size on XIV array and map the to the database server using XIVGUI or xcli. For example:

```
pool_create pool="asmtest" size=1013 snapshot_size=200
vol_create vol="asm_5" size=34 pool="OraASM"
vol_create vol="asm_6" size=34 pool="OraASM"
vol_create vol="asm_7" size=34 pool="OraASM"
vol_create vol="asm_8" size=34 pool="OraASM"
map_vol host="HOST_244" vol="asm_5" lun=10
map_vol host="HOST_244" vol="asm_6" lun=12
map_vol host="HOST_244" vol="asm_7" lun=13
map_vol host="HOST_244" vol="asm_8" lun=14
```

Step 2: Mapping XIV LUNs to the database server

As root, using XIV's host connectivity package, rescan SCSI bus to discover XIV LUNs on the database server. For example:

```
/opt/xiv/host_attach/bin/qlogic_lun_scan -s
```

Display the discovered volumes by using the vol_list command:

```
[root@host244 ~]# /opt/xiv/host_attach/bin/vol_list
Multipathed IBM-XIV 2810 devices
------------------------------------------------------------------------------------------
Device file                                Volume name          Size      Machine   Vol id
------------------------------------------------------------------------------------------
/dev/mapper/mpath19                        asm_6                 34 GB    000010    00533
/dev/mapper/mpath21                        asm_8                 34 GB    000010    00535
/dev/mapper/mpath18                        asm_5                 34 GB    000010    00532
/dev/mapper/mpath20                        asm_7                 34 GB    000010    00534
/dev/mapper/mpath17                        asm_4                 34 GB    000010    00531
/dev/mapper/mpath16                        asm_3                 34 GB    000010    00530
/dev/mapper/mpath15                        asm_2                 34 GB    000010    00529
/dev/mapper/mpath14                        asm_1                 34 GB    000010    00528
/dev/mapper/mpath13                        asm_dump             412 GB    000010    00454

No Non-multipathed IBM-XIV 2810 devices found.
```

Figure 9: `vol_list` command output

Step 3: Marking LUNs for use by ASM

As root, mark XIV LUNs for use by ASMLib:

```
/etc/init.d/oracleasm createdisk VOL5 /dev/mapper/mpath18
```

To verify that ASMLib marked all required LUNs run:

```
/etc/init.d/oracleasm scandisks
/etc/init.d/oracleasm listdisks
```

```
[root@host244 ~]# /etc/init.d/oracleasm listdisks
VOL1
VOL2
VOL3
VOL4
VOL5
VOL6
VOL7
VOL8
[root@host244 ~]#
```

Figure 10: `oracleasm listdisks` command output

Change the ownership of the device path to `oracle:dba`. For example:

```
chown oracle:dba /dev/mapper/mapth18
```

Step 8: Adding XIV LUNs to a diskgroup

As oracle user, under ASM instance, add at least two new failgroups to the database diskgroup from sqlplus command line. For example:

```
SQL> alter diskgroup ASM_TEST add
SQL>  failgroup falgroup3 disk
SQL>    '/dev/mapper/mpath18',
SQL>    '/dev/mapper/mpath19'
SQL>  failgroup falgroup4 disk
```

```
SQL>     '/dev/mapper/mpath20',
SQL>     '/dev/mapper/mpath21';
```

To enable faster synchronization between the original disks with the newly added, you can "add power" to the synchronization process:

```
SQL> alter diskgroup ASM_TEST REBALANCE POWER 11;
```

The power value ranges from 1 to 11, with default of 1. The appropriate value is dependent on your performance and availability requirements.

From ASMCMD command line, check the updated definition of the diskgroup:

```
ASMCMD> lsdsk -k -d ASM_TEST
```

```
ASMCMD> lsdsk -k -d ASM_TEST
Total_MB  Free_MB  OS_MB  Name           Failgroup   Library  Label  UDID  Product  Redund   Path
   32768    32493  32768  ASM_TEST_0000  FAILGROUP1  System                        UNKNOWN  /dev/mapper/mpath14
   32768    32493  32768  ASM_TEST_0001  FAILGROUP1  System                        UNKNOWN  /dev/mapper/mpath15
   32768    32494  32768  ASM_TEST_0002  FAILGROUP2  System                        UNKNOWN  /dev/mapper/mpath16
   32768    32494  32768  ASM_TEST_0003  FAILGROUP2  System                        UNKNOWN  /dev/mapper/mpath17
   32768    27087  32768  ASM_TEST_0004  FAILGROUP3  System                        UNKNOWN  /dev/mapper/mpath18
   32768    27088  32768  ASM_TEST_0005  FAILGROUP3  System                        UNKNOWN  /dev/mapper/mpath19
   32768    27088  32768  ASM_TEST_0006  FAILGROUP4  System                        UNKNOWN  /dev/mapper/mpath20
   32768    27088  32768  ASM_TEST_0007  FAILGROUP4  System                        UNKNOWN  /dev/mapper/mpath21
```

Figure 11: Disk group configuration information

To check on the synchronization status, use the following command:

```
ASMCMD> lsdg -g -d ASM_TEST
```

```
ASMCMD> lsdg -c -g
Inst_ID  State    Type    Rebal  Sector  Block       AU  Total_MB  Free_MB  Req_mir_free_MB  Usable_file_MB  Offline_disks  Name
     1  MOUNTED  NORMAL  Y         512   4096  1048576    262144   238325            65536           86394              0  ASM_TEST/
```

Figure 12: Synchronization in process

Step 5: Removing original (non XIV) disks from diskgroup

Make sure that data is completely synchronized across all failgroups (see Figure 13).

```
ASMCMD> lsdg -g
Inst_ID  State    Type    Rebal  Sector  Block       AU  Total_MB  Free_MB  Req_mir_free_MB  Usable_file_MB  Offline_disks  Name
     1  MOUNTED  NORMAL  N         512   4096  1048576    262144   238323            65536           86393              0  ASM_TEST/
```

Figure 13: No active synchronization processes

Remove the original disks from the diskgroup by running the following command from sqlplus:

```
SQL> alter diskgroup ASM_TEST
SQL> drop disks in failgroup failgroup1, failgroup2;
```

At this point ASM will rebalance the updated diskgroup. Once the rebalance operation is done, you can safely remove the original disks from the database server.

```
SMCMD> lsdsk -k -d ASM_TEST
otal_MB  Free_MB  OS_MB  Name          Failgroup   Library  Label  UDID  Product  Redund   Path
  32768    26832  32768  ASM_TEST_0004  FAILGROUP3  System                        UNKNOWN  /dev/mapper/mpath18
  32768    26834  32768  ASM_TEST_0005  FAILGROUP3  System                        UNKNOWN  /dev/mapper/mpath19
  32768    26833  32768  ASM_TEST_0006  FAILGROUP4  System                        UNKNOWN  /dev/mapper/mpath20
  32768    26833  32768  ASM_TEST_0007  FAILGROUP4  System                        UNKNOWN  /dev/mapper/mpath21
```

Figure 14: Diskgroup configuration on XIV disks

Using Oracle Recovery Manager (RMAN) to Migrate to XIV Array – Diskgroup with EXTERNAL, NORMAL or HIGH Redundancy

The following migration procedure utilizes Oracle RMAN to migrate the database to XIV storage. A small outage window is required to redirect the database from legacy storage to XIV. The advantage of this procedure is that it can be used to migrate diskgroups with EXTERNAL redundancy and it does not require preserving original volume layout. Most of the migration process is done while the database is online and applications can access it. But, it will have a performance impact on the application since the copy process consumes server and storage resources to copy the data between the arrays.

Note: This migration option might require XIV volumes to be presented to a server with legacy array volumes managed by a multipath driver that doesn't support XIV, for example PowerPath. Please submit a RPQ request to enable temporary support of this configuration (only during the migration period, when XIV and legacy LUNs have to be presented to the database server simultaneously).

There are three phases to this migration process:

1. Pre Migration
2. The Switch
3. Post Migration

Pre Migration

Step 1: Allocating LUNs on XIV array

1. Install and configure XIV host attachment kit. For details please review XIV's "***Host System Attachment Guide".***
2. Create LUNs of desired size on XIV array and map them to the database server using XIVGUI or xcli. Make sure the total size of allocated LUNs is equal or larger then the original storage size. On the other hand, number of LUNs allocated on XIV does not have to match number of LUNs on the legacy array. For example:

```
pool_create pool="asmnew" size=1013 snapshot_size=200
vol_create vol="asm_new_1" size=500 pool="OraASM"
map_vol host="HOST_244" vol="asm_new_1" lun=10
```

3. As root, using XIV's host connectivity package, rescan SCSI bus to discover XIV LUNs on the database server.

```
/opt/xiv/host_attach/bin/qlogic_lun_scan –s
```

4. Display the discovered volumes by using the vol_list command.

```
/opt/xiv/host_attach/bin/vol_list
```

Step 2: Marking XIV LUNs for use by ASM

5. As root, mark XIV LUNs for use by ASMLib:

```
/etc/init.d/oracleasm createdisk VOL5 /dev/mapper/mpath18
```

6. To verify that ASMLib marked all required LUNs run:

```
/etc/init.d/oracleasm scandisks
/etc/init.d/oracleasm listdisks
```

7. Change the ownership of the device path to `oracle:dba`. For example:

```
chown oracle:dba /dev/mapper/mapth18
```

8. Create a new diskgroup on XIV LUNs

```
SQL> create diskgroup XIV_ASM external redundancy
SQL> disk '/dev/mapper/mpath18';
```

Step 3: Prepare the Production Database for Migration

The next step is to advise the production database that XIV volumes should be used for all new data files as well as all recovery area usage.

1. Change the db_create_file_dest init.ora parameter to point to the XIV_ASM diskgroup.

```
SQL> alter system set db_create_file_dest='+XIV_ASM' scope=both;
```

2. Change the db_recovery_file_dest init.ora parameter to point to the XIV_ASM diskgroup.

```
SQL> alter system set db_recovery_file_dest='+XIV_ASM'
SQL> scope=both;
```

Step 4: Migrate the current RMAN backups to the Recovery Area

If RMAN is used to backup the database, this step will migrate all the current RMAN backups in the recovery area to the XIV diskgroup.

1. Move current backup sets to the XIV diskgroup

```
RMAN> backup backupset all delete input;
```

2. Move current data file copies to the XIV diskgroup

```
RMAN> backup as copy datafilecopy all delete input;
```

3. Move current archive log files

```
RMAN> backup as copy archivelog all delete input;
```

4. If Database Block Change Tracking has been enabled previously, the file must be recreated in the XIV diskgroup:

```
SQL> alter database disable block change tracking;
SQL> alter database enable block change tracking;
```

If Database Block Change Tracking has not been enabled previously on the database, then it must be enabled for the duration of the migration.

```
SQL> alter database enable block change tracking;
```

Step 5: Make the initial copy of the Oracle datafiles on XIV

```
RMAN> backup device type disk incremental level 0 as copy tag
RMAN> 'XIV_Migration' database format '+XIV_ASM';
```

Step 6: Migrate the Oracle Redo Log and Standby Redo Log files to XIV diskgroup

This phase will move the Oracle redo log files and Oracle standby redo log files to XIV diskgroup. How the new redo log files are added, depends upon the number of redo log members that control file can support. The following example assumes 1 redo log members per group.

1. Identify the current redo log members

```
SQL> select member from v$logfile;
MEMBER
--------------------------------------------------------------------------------
 +OLD_ASM/ora/onlinelog/group_1.456.689706377
 +OLD_ASM/ora/onlinelog/group_2.457.689706441
 +OLD_ASM/ora/onlinelog/group_3.458.689706447
```

2. For each redo log group add the new redo log members:

```
SQL> alter database add logfile member '+XIV_ASM' to group 1;
SQL> alter database add logfile member '+XIV_ASM' to group 2;
SQL> alter database add logfile member '+XIV_ASM' to group 3;
```

3. Archive all online logfiles by:

```
SQL> alter system switch logfile;
SQL> /
SQL> /
```

4. Drop the other original redo log member

```
 SQL> alter database drop logfile member
 SQL> '+OLD_ASM/ora/onlinelog/group_1.456.689706377';
 SQL> alter database drop logfile member
 SQL> '+OLD_ASM/ora/onlinelog/group_2.457.689706441';
 SQL> alter database drop logfile member
 SQL> '+OLD_ASM/ora/onlinelog/group_3.458.689706447';
```

Step 7: Migrate the tempfiles to XIV diskgroup

RMAN does not migrate the tempfiles as part of the `BACKUP AS COPY` and `SWITCH` command because the tempfile is not listed in controlfile. It has to be manually migrated.

1. Identify the current temporary files and their sizes

```
SQL> select bytes, name from v$tempfile;
BYTES       NAME
---------- ------------------------------------------------------------
20971520   +OLD_ASM/ora/tempfile/temp.460.689706925
```

2. Add the new temporary file

```
SQL> alter tablespace temp add tempfile size 20M;
```

3. Remove the original temporary file

```
SQL> alter database tempfile
SQL> '+OLD_ASM/ora/tempfile/temp.460.689706925' drop;
```

Step 8: Refresh the previous copy of the Oracle Datafiles

This phase will refresh the copies of all the Oracle datafiles in the XIV_ASM diskgroup.

Using RMAN create an incremental level 1 backup of the database, and restore the backup into the database copy.

```
RMAN> run {
RMAN> backup incremental level 1 for recover of copy with tag
RMAN> 'XIV_Migration' database;
RMAN> recover copy of database with tag 'XIV_Migration';
RMAN> }
```

The Switch

This is the start of the outage phase, which should be kept to a minimum of steps:

1. Prepare the control_file from ASM Disk Groups
2. Switching the database data files to XIV diskgroup

Step 1: Prepare the control_file from XIV diskgroup

1. Ensure that the directory structure exists in the new XIV diskgroup for the control files.

```
SQL> alter database backup controlfile to '+XIV_ASM';
```

2. Determine the value of the db_unique_name init.ora parameter.

```
SQL> show parameter db_name
NAME                                 TYPE        VALUE
------------------------------------ ----------- ----------------------
db_name                              string      ORCL
SQL> show parameter db_unique_name
```

```
NAME                                 TYPE        VALUE
------------------------------------ ----------- ------------------------
db_unique_name                        string     ORCL
```

3. Update the control_file parameter in the spfile

```
SQL> alter system set control_files =
SQL> '+XIV_ASM/{DB_UNIQUE_NAME}/CONTROLFILE/control01.ctl',
SQL> '+XIV_ASM/{DB_UNIQUE_NAME}/CONTROLFILE/control02.ctl'
SQL> scope=spfile;
```

4. Capture the current control file name

```
SQL> select name from v$controlfile;
NAME
--------------------------------------------------------------------------------
+OLD_ASM/ORCL/CONTROLFILE/control01.ctl
+OLD_ASM/ORCL/CONTROLFILE/control02.ctl
```

Step 2: Switching the database data files to XIV diskgroup

1. Shutdown the database cleanly:

```
RMAN> shutdown immediate;
```

2. Startup nomount the instance in preparation for restoring the control files.

```
RMAN> startup nomount
```

3. Restore one of the original control files into the new control file locations.

```
RMAN> restore controlfile from
'+OLD_ASM/ORCL/CONTROLFILE/control01.ctl;
Starting restore at 25-JUN-09
allocated channel: ORA_DISK_1
channel ORA_DISK_1: sid=270 devtype=DISK
channel ORA_DISK_1: copied controlfile copy
output filename=+XIV_ASM/ORCL/CONTROLFILE/control01.ctl'
output filename=+XIV_ASM/ORCL/CONTROLFILE/control02.ctl'
Finished restore at 25-JUN-09
```

4. Mount the database

```
RMAN> alter database mount;
```

5. Switch the database to use the backup datafiles copies created previously

```
RMAN> switch database to copy;
```

6. Since there is a degree of change between the time the last incremental backup was taken and the database being shutdown, we must recover the datafile copies.

```
RMAN> recover database;
```

7. Open the database

```
RMAN> alter database open;
```

Post Migration

1. If Block Change Tracking was enabled for the purpose of the migration, then this should now be disabled

```
SQL> alter database disable block change tracking;
```

2. Query the database and ensure that all files reside in either the XIV_ASM diskgroup

```
SQL> select name from v$controlfile
union
select name from v$datafile
union
select name from v$tempfile
union
select member from v$logfile
union
select filename from v$block_change_tracking;
```

3. Delete the legacy ASM diskgroup

```
SQL> drop diskgroup OLD_ASM including contents;
```

4. Deallocate the legacy storage LUNs

5. Deinstall the original multipath driver

Provisioning Storage to Oracle Database

In order to enable oracle database files to extend automatically, independent of a method in which additional storage is added, you can create datafiles or alter existing datafiles with AUTOEXTEND option. The file size increases in specified increments up to a specified maximum or unlimited. For example:

```
SQL> CREATE BIGFILE TABLESPACE "test" LOGGING
SQL> DATAFILE '+ASM_TEST'
SQL> SIZE 300G AUTOEXTEND ON NEXT 30G MAXSIZE UNLIMITED;
```

Extending size of an existing ASM disk on Linux

The following example will demonstrate how to add additional storage to an Oracle database by growing the size of already allocated storage. This procedure requires to shutdown all instances on the database server and ASM for a short time to restart the multipath driver.

```
[root@host244 ~]# /opt/xiv/host_attach/bin/vol_list
Multipathed IBM-XIV 2810 devices
-------------------------------------------------------------------------------------------
Device file                                Volume name          Size      Machine   Vol id
-------------------------------------------------------------------------------------------
/dev/mapper/mpath23                        asm_9                 85 GB    000010    00539
/dev/mapper/mpath22                        asm_10                85 GB    000010    00540
```

Figure 15: Displays 2 XIV LUNs, 85 GB each, allocated to an ASM diskgroup ASM_TEST

```
ASMCMD> lsdsk -k -d ASM_TEST
Total_MB  Free_MB  OS_MB  Name           Failgroup   Library  Label  UDID  Product  Redund   Path
   81920     70037  81920  ASM_TEST_0000  FAILGROUP1  System                        UNKNOWN  /dev/mapper/mpath22
   81920     70037  81920  ASM_TEST_0001  FAILGROUP2  System                        UNKNOWN  /dev/mapper/mpath23
ASMCMD>
```

Figure 16: ASM diskgroup with 2 85 GB LUNs

Step 1: Grow XIV LUN sizes

To increase the XIV LUN size use XIVGUI or xcli (Figure 17).

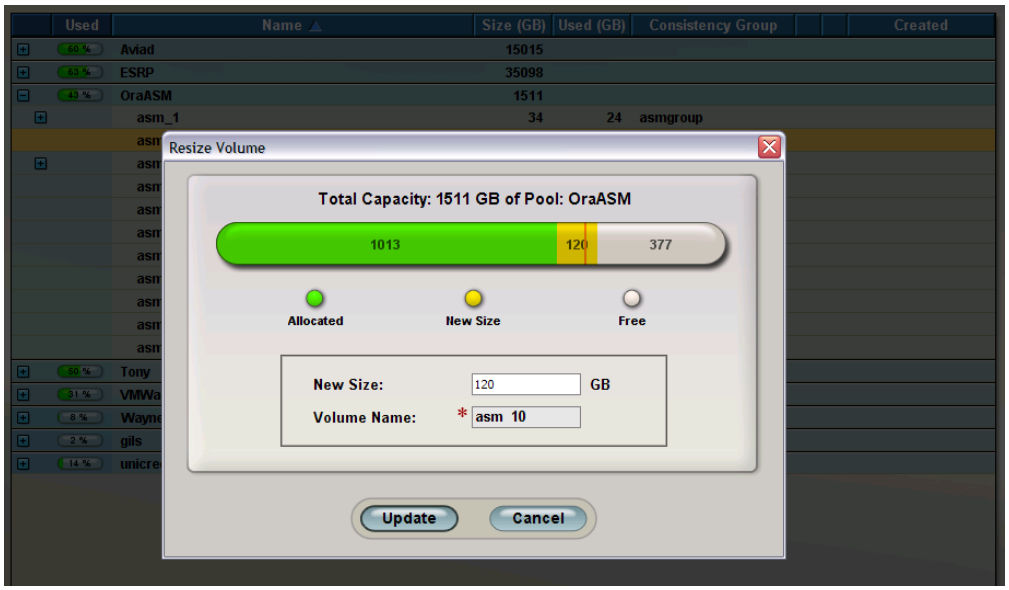

Figure 17: Increasing capacity of XIV LUN via XIVGUI

Step 2: Get list of all block device names on the mpath

As root, get the list of all block devices in the mpath by running the multipath command. For example

```
multipath -l mpath22
multipath -l mpath23
```

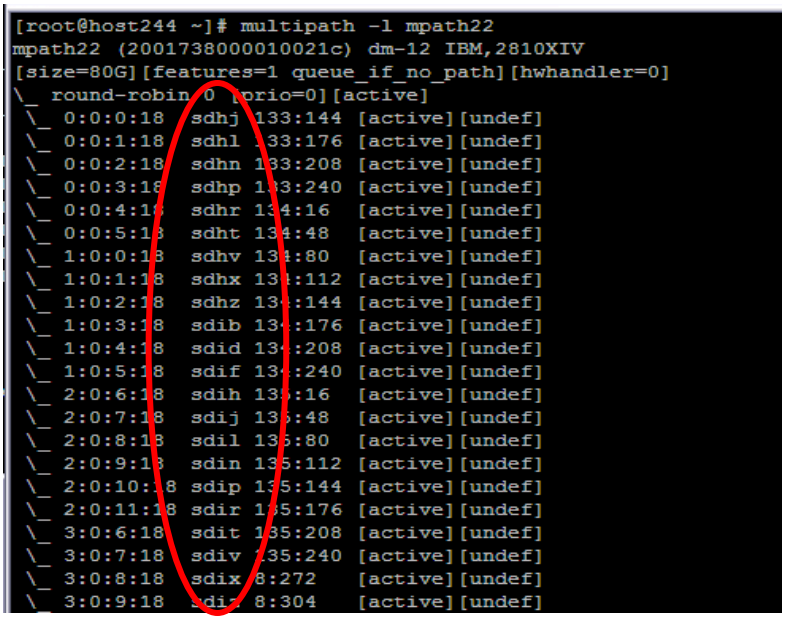

Figure 18: List all the block devices associated with /dev/mapper/mpath22

Step 3: Reread the partition table for each block device in the multipath

For each block device on the list from the *Step 2* reread the partition table with `blockdev` command. For example:

```
blockdev --rereadpt /dev/sdhj
blockdev --rereadpt /dev/sdhl
.
.
.

blockdev --rereadpt /dev/sdjd
```

Step 4: Shutdown database and ASM

As Oracle administrator shutdown:

1. All database instances on the database server
2. ASM instance

Step 5: Restart the multipath driver and reset LUN ownership

As root:

1. Restart the multipath service
   `/etc/init.d/multipathd restart`
2. Rebuild the multipath devices
   `multipath`
3. Change back the ownership of the device to `oracle:dba`
   `chown oracle:dba /dev/mapper/multipath22`
   `chown oracle:dba /dev/mapper/multipath23`

Step 6: Restart ASM and database instances

As Oracle administrators:

1. Restart ASM instance
2. Restart all database instances on the server

```
[oracle@host244 ~]$ asmcmd
ASMCMD> lsdsk -k -d ASM_TEST
Total_MB  Free_MB  OS_MB  Name          Failgroup   Library  Label  UDID  Product  Redund   Path
   81920    70035  114688  ASM_TEST_0000  FAILGROUP1  System                        UNKNOWN  /dev/mapper/mpath22
   81920    70035  114688  ASM_TEST_0001  FAILGROUP2  System                        UNKNOWN  /dev/mapper/mpath23
ASMCMD>
```

Figure 19: ASM diskgroup with 2 120 GB LUNs

If the database or tablespace is defined with `AUTOEXTEND` option, it will automatically claim the allocated space as it grows.

Extending storage space by adding disks to ASM diskgroups

The following example will demonstrate how to add additional storage to an Oracle database by adding disks to ASM diskgroup. This procedure does not require shutting down ASM or database instances, and does not affect at all the database and application availability.

Step 1: Creating new LUNs and mapping them to the database server

Using XIVGUI create new LUNs. The size of those LUNs preferably should be equal to LUNs already allocated to the database to guarantee even data distribution across all disks in the disk group.

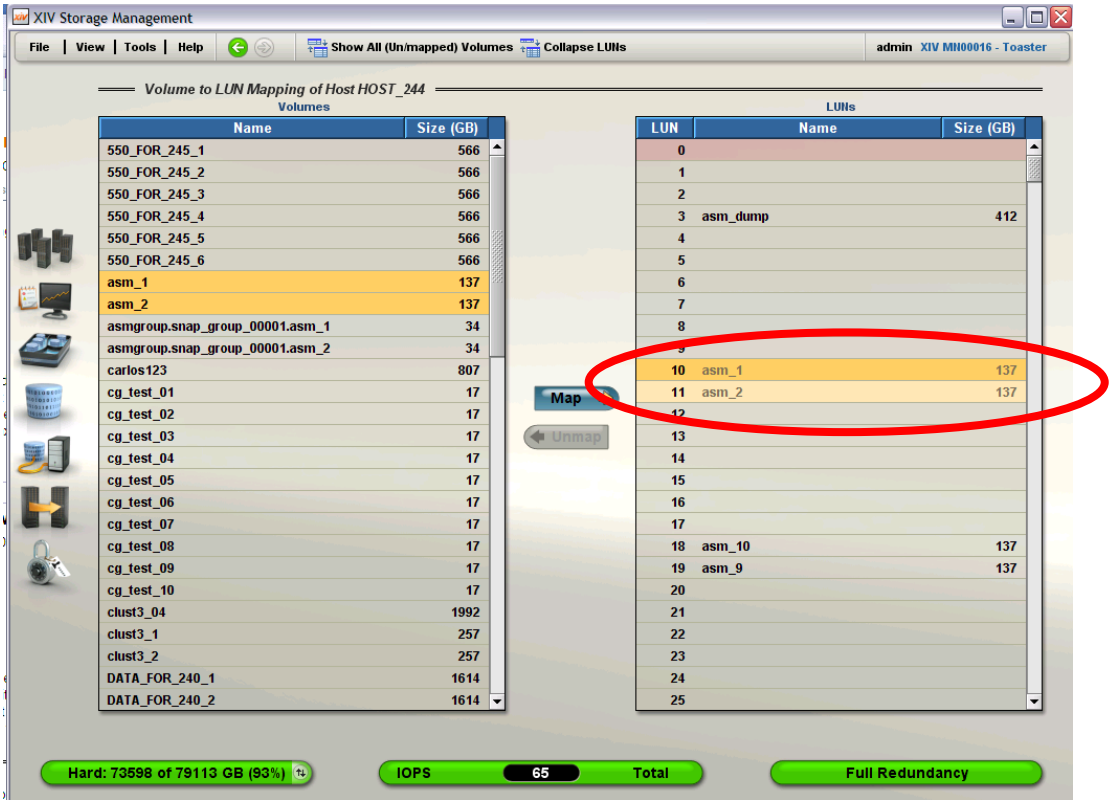

Figure 20: Mapping additional storage to the database server

Step 2: Discover the new storage on the server

Discover the new LUNs by scanning the SCSI bus with the following command:

```
/opt/xiv/host_attach/bin/qlogic_lun_scan -s
```

To view the new LUNs run the `vol_list` command.

```
[root@host244 ~]# /opt/xiv/host_attach/bin/vol_list
Multipathed IBM-XIV 2810 devices
------------------------------------------------------------------------------------------------
Device file                                Volume name          Size      Machine   Vol id
------------------------------------------------------------------------------------------------
/dev/mapper/mpath23                        asm_9                137 GB    000010    00539
/dev/mapper/mpath22                        asm_10               137 GB    000010    00540
/dev/mapper/mpath15                        asm_2                137 GB    000010    00529
/dev/mapper/mpath14                        asm_1                137 GB    000010    00528
/dev/mapper/mpath13                        asm_dump             412 GB    000010    00454
```

Figure 21: List all mpath XIV LUNs with `vol_list` command

Step 3: Adding disks to ASM diskgroup

As root mark the new disks for use by ASMLib:

```
/etc/init.d/oracleasm VOL3 /dev/mapper/mpath14
/etc/init.d/oracleasm VOL4 /dev/mapper/mpath15
/etc/init.d/oracleasm listdisks
```

Change the ownership of the XIV devices to oracle:dba. For example:

```
chown oracle:dba /dev/mapper/mapth1*
```

As ASM administrator add the new LUNs above to ASM diskgroup:

```
SQL> alter diskgroup ASM_TEST add
SQL> failgroup FAILGROUP1 disk '/dev/mapper/mpath14'
SQL> failgroup FAILGROUP2 disk '/dev/mapper/mpath15';
```

If the database or tablespace is defined with `AUTOEXTEND` option, it will automatically claim the allocated space as it grows.

Cloning Oracle Database with XIV Snapshots and Consistency Groups

Many database administrative activities, like database backup and cloning for create development/test environments have significant impact on database availability and performance.

XIV's snapshot functionally allows performing those operations from the snapped copy with no performance or availability impact on production environment, enabling creation of practically unlimited number of database clones. Snapshot creation is a simple and easy task to accomplish. It can be done we XIVGUI or scripted and automated with xcli.

If a database spans more then one XIV volume, a XIV Consistency Group should be used to pool all the database volumes together so that a snapshot can be taken of all the volumes at the same time.

When backing up a database, it is important to establish consistent image of the data across all LUNs, so that it is recoverable on the database level. If the data is inconsistent, a database restore will not be possible, because the inconsistency between the logs and the data. Therefore, part of the data may be lost.

When XIV's consistency groups and snapshots are used to create a database copy, database consistency can be established without shutting it down or even putting it into a backup mode.

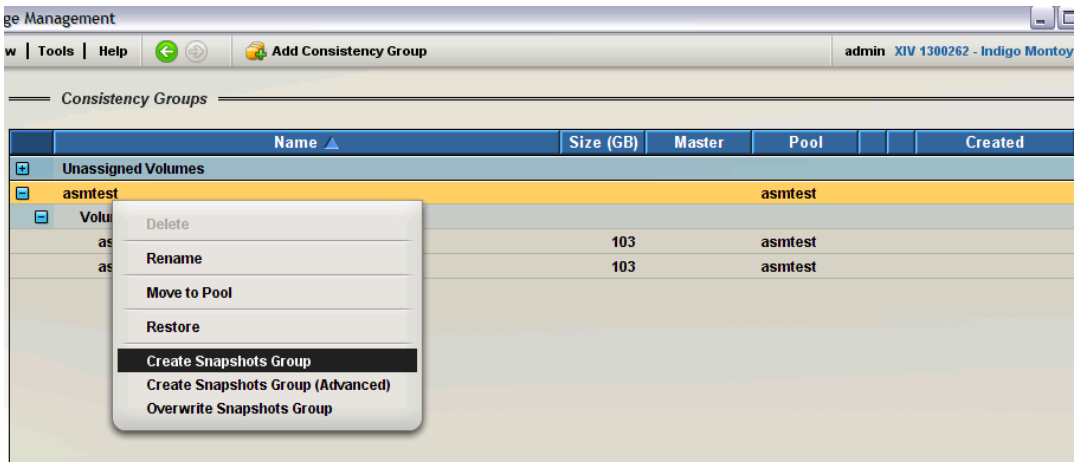

Figure 21: Creating Consistency Group snapshots with XIVGUI

The following procedure will describe in detail how to prepare the database for cloning, cloning ASM and database instances on the backup/development host, mounting the clone on the backup host and restoring production database from the snap.

Step 1: Preparing Oracle database for cloning

1. To make sure that you can create consistent and recoverable database snapshots u need to make sure that the database was created with ARCHIVELOG mode. To check if this mode is on execute the following command as Oracle database administrator from sqlplus command line:

```
SQL> archive logs list
```

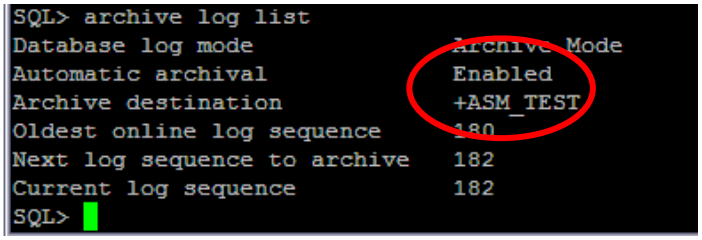

Figure 22: Validating that the ARCHIVELOG mode is on and the log destination

In addition, it is important to verify that the archive destination resides on XIV volumes that you intend to snap and not on a host internal drive.

If the ARCHIVELOG is disabled execute the following steps from sqlplus command line to turn it on:

```
SQL> shutdown immediate
SQL> startup mount
SQL> alter system set log_archive_dest_1='LOCATION=+ASM_TEST/';
SQL> alter system set log_archive_format=%t_%s_%r.dbf;
SQL> alter database archivelog;
SQL> alter database open;
```

2. To prepare the source ASM and Oracle home to be cloned, archive the home directories for those instances. Make sure to exclude *.log, *.dbf, listener.ora, sqlnet.ora and tnsnames.ora files. For example:

```
zip -r asm asm -x \*.log \*.dbf \*listener.ora \*sqlnet.ora \*tnsnames.ora
zip -r ora ora -x \*.log \*.dbf \*listener.ora \*sqlnet.ora \*tnsnames.ora
```

Step 2: Preparing the backup server

In this procedure steps 1-4, 7 and 10 should be executed only 1st time, during backup host configuration process. The rest of the steps are required every time you want to refresh a snapshot mounted to the backup host.

1. Install and configure XIV host attachment kit. For details please review XIV's "***Host System Attachment Guide".***

2. Install and configure Oracle and ASM software. For details please review revilement "***Installing Oracle Database***" guide.

3. Copy the zip files to the backup host and extract the content of those files.

4. As root run:

```
$ASM_HOME/root.sh
$ORACLE_HOME/root.sh
```

5. Create a XIV snapshot and map the snapshot volumes to the backup hosts

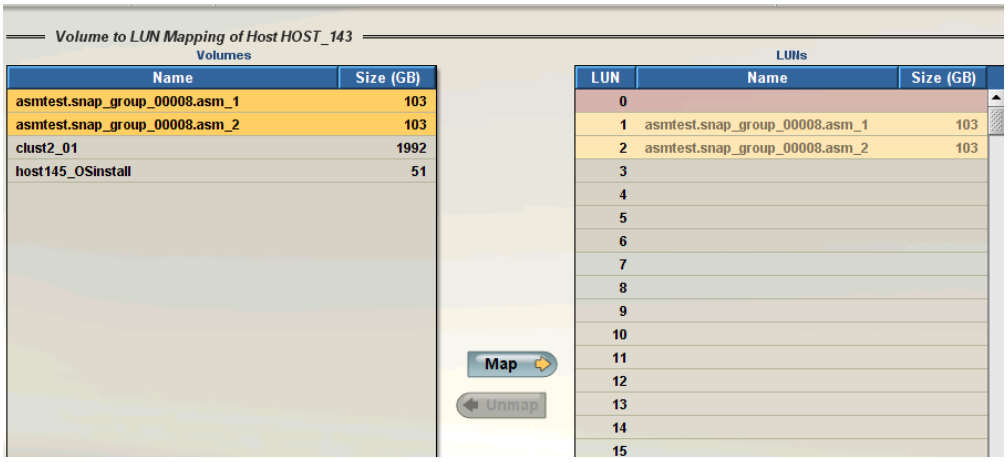

Figure 22: Mapping snapshot devices to backup host

6. Discover the new LUNs by scanning the SCSI bus with the following command:

```
/opt/xiv/host_attach/bin/qlogic_lun_scan -s
```

7. Configure the database connection information by running:

```
$ASM_HOME/bin/netca
```

8. Verify that ASMLib identifies all snapshot LUNS:

```
/etc/init.d/oracleasm scandisks
/etc/init.d/oracleasm listdisks
```

Note: There is no need to create new ASMLib volumes, since the snap volumes have ASM labels from the original volumes.

9. Change the ownership of the XIV device path to oracle:dba. For example:

```
chown oracle:dba /dev/mapper/mapth18
```

Note: This is one of the most important steps to execute which is usually forgotten. If it is not executed, Oracle ASM won't be able to mount diskgroups on the volumes.

10. As Oracle Configure ASM instance using DBCA utility. For example:

```
$ORACLE_HOME/bin/dbca –silent –configureASM –gdbName NO –sid NO \
-emConfiguration NONE \
-diskSring '/dev/mapper/mpath*' \
-obfuscatedPasswords false \
-oratabLocation /etc/oraatab \
-asmSysPasword password
```

11. As ASM admin using sqlplus command line mount the diskgroup:

```
SQL> alter diskgroup ASM_TEST mount;
```

12. As Oracle database administrator startup database clone from sqlplus command line:

```
SQL> startup
```

From this point the clone database is ready for any administrative activities otherwise would have been implemented on the production database, for example:

- Database backup using RMAN
- Application development and testing
- Decision support analytics
- Etc…

Restoring database from a XIV snapshot

Restoring XIV snapshots to the original volumes is as simple and as easy task to perform as creating one. You can use the snapshots in addition to or instead tape backup, generating the snapshot as frequently as business SLAs dictate, without impacting database availability or performance.

The procedure below guarantees quick recovery of the production database, with minimum down time.

Step 1: Unmap the snapshot from backup host

In case when a snapshot required for restore is mounted on a backup host follow the process below:

1. Shutdown the Oracle database on backup host
2. Shutdown the ASM instance on backup host
3. Using XIVGUI or xcli unmap snap volumes from the backup host

Step 2: Prepare production host for restore

On production host:

1. Shutdown the database

2. Unmount ASM diskgroup being restored. For example:

```
SQL> alter diskgroup ASM_TEST dismount;
```

3. Using XIVGUI or xcli lock snap volumes (if unlocked) and restore it to the production volume. For example:

```
xcli snap_group_lock snap_group="asmtest.snap_group_00008"
xcli snap_group_restore snap_group="asmtest.snap_group_00008"
```

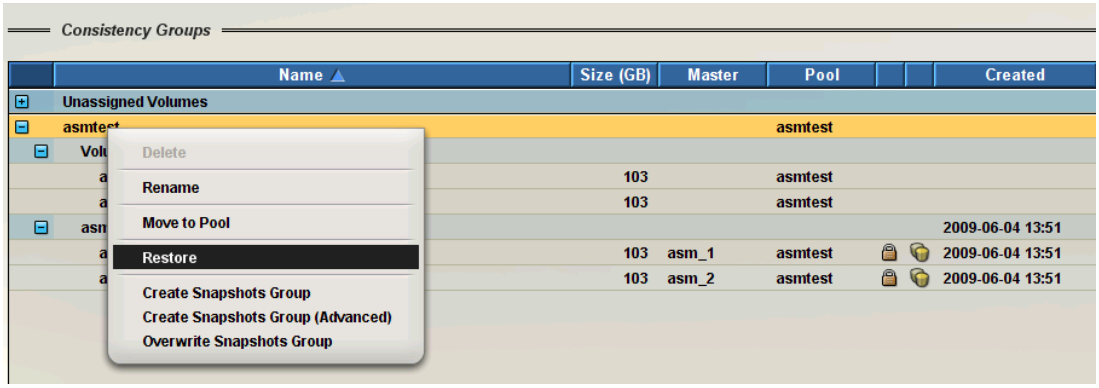

Figure 23: Restoring Snapshot Group using XIVGUI

The restore operation makes the data on the snapshot instantly available on production volumes.

4. Mount restored ASM diskgroup.

```
SQL> alter diskgroup ASM_TEST mount;
```

5. Startup the production database