Learning ASM Using a Single Disk Drive On the Apple OS X and Linux Platforms

An ATS HOWTO Paper
by Kent Stroker
July 2005

Advanced Technology Services, Inc.
Oracle Technology Delivered

Learning ASM Using a Single Disk Drive On the Apple OS X and Linux Platforms

TECHNICAL OVERVIEW

Automatic Storage Management (ASM) is a new feature of Oracle Database 10g and is thoroughly discussed in any number of Oracle publications. Visit otn.oracle.com and find a wealth of detailed information regarding ASM. The essential idea is that ASM distributes files across multiple disk drives allowing the DBA to manage these dynamically without the need to shutdown the database. This brings many benefits to the DBA and to overall database operations. The use of ASM is highly recommended.

For the DBA who wishes to learn this new technology, there is the very real dilemma in that most DBAs do not have access to multiple disk drives in order to gain experience with this new feature and develop best practices for their own environment. It is critical for the success of a deployed database project, not to mention the long-term career of the DBA, to have experience on how ASM works, how to deploy and tune ASM, how to grow with ASM and what to do when something goes astray (not that Oracle technology ever suffers from techno wanderlust).

There are methods by which a single disk drive can be used to simulate multiple raw disk drives, which in turn can be managed by ASM on a single disk drive, on a single server. This paper demonstrates how to create such pseudo disk drive devices using techniques for both the Apple OS X and Redhat Linux operating systems.

The basic idea is simple. Create a 2GB (or any size desired) disk file, then "persuade" the operating systems to "see" and "mount" the file as a raw device. Once "transmogrified" (a classic "Calvin and Hobbs" term) into a raw device, the Oracle ASM feature is able to use that device as a disk group(s).

Take note and dire warnings, these techniques are not suitable for any production environment and are meant only for educational purposes. The disk I/O performance will be slow, but the Oracle ASM feature will be fully functional. Repeat; do not use this for any production system, as the performance would be unsatisfactory and Oracle would not support it.

Apple OS X Raw Pseudo Disk Drives

In the Apple OS X operating system, the underlying operating system is a Unix derivative of FreeBSD called "Darwin". Apple does not use disk drives, partitions and file systems like other variants of Unix and the ability to create addressable raw devices from filesystem level files requires a slightly different approach. This approach is a combination of both command line and GUI steps.

The following steps may be used to create six (6) raw pseudo disk drives.

Access to the admin and root accounts is required to use this technique.

Step One – Create the Image Files

The first step is to create disk files, which will mimic disk drives. Using the `hdiutil` command create six (6) 2GB disk image files as follows:

```
$ su -
# hdiutil create /Volumes/asmDisk1 -size 2g -partitionType Apple_Label
# hdiutil create /Volumes/asmDisk2 -size 2g -partitionType Apple_Label
# hdiutil create /Volumes/asmDisk3 -size 2g -partitionType Apple_Label
# hdiutil create /Volumes/asmDisk4 -size 2g -partitionType Apple_Label
# hdiutil create /Volumes/asmDisk5 -size 2g -partitionType Apple_Label
# hdiutil create /Volumes/asmDisk6 -size 2g -partitionType Apple_Label
```

The `hdiutil` command is a BSD general command, which manipulates disk images. The above commands create a 2GB disk image with 2 partitions contained therein. The first (e.g., disk4s1) is the standard Apple partition map, the second (e.g., disk4s2) is marked as partition type Apple_Label. This is an important detail since only this type of partition can have a label later associated to it using the `disklabel` command.

The end result of the above commands are six (6) 2GB disk images files in the `/Volumes` directory named `asmDisk<`*n*`>.dmg`.

Step Two – Force Disk Utility to Recognize Image Devices

Once the disk images are created, it is necessary to get OS X to recognize the existence of these pseudo disk drive devices. Recall that these are to be used as raw devices, so these pseudo disk drives will not be formatted with a filesystem.

Change to the `/Volumes` directory and change the permissions on the newly create image device files. Then use `hdiutil attach` command to attach the drives to the operating system. The command will throw an error, this is expected and may be safely ignored.

```
# cd /Volumes
# chmod 0666 disk*

# hdiutil attach asmDisk1.dmg
# hdiutil attach asmDisk2.dmg
# hdiutil attach asmDisk3.dmg
# hdiutil attach asmDisk4.dmg
# hdiutil attach asmDisk5.dmg
# hdiutil attach asmDisk6.dmg
```

The end result of this step is that OS X and more importantly, the `Applications>Utilties>Disk Utility` tool shall "see" these disk images as disk drives. It is important to note that this step does not mount these disk drives, just gets OS X to "take notice of them" for the time being.

Step Three – Use Disk Utility to Open Image Devices

The next step is to force OS X to attach these pseudo disk drive devices to the operating system so that disk labels can be associated with the partitions and endpoint raw devices.

`Using the Applications>Utilities>Disk Utility`, tool double-click each pseudo disk drive, each drive should then show a dynamically assigned disk drive number and partition slice.

Step Four – Associate a Disk Label to each Drive Partition

Oracle directly addresses raw devices by using the /dev/r<filename> convention (e.g., /dev/rraw_asmDisk1). The `disklabel` command is used to create a persistently stored raw device label, which can be dynamically assigned to a dynamic disk drive device.

Start by using the `diskutil` command to list out the currently known list of disk drives and their associated partition maps. Make note of those drive numbers that are associated with the pseudo disk drives.

```
# diskutil list | more
```

Use the following commands, making sure to substitute the correct disk drive device number and partition numbers to create a disk label for each pseudo disk drive raw partition. Please take note that the following commands are displayed in line-wrapped style.

```
# disklabel –create /dev/rdisk<n>s2 owner-uid=oracle
> group-gid=dba owner-mode=0640 dev-name=raw_asm1

# disklabel –create /dev/rdisk<n>s2 owner-uid=oracle
> group-gid=dba owner-mode=0640 dev-name=raw_asm2

# disklabel –create /dev/rdisk<n>s2 owner-uid=oracle
> group-gid=dba owner-mode=0640 dev-name=raw_asm3

# disklabel –create /dev/rdisk<n>s2 owner-uid=oracle
> group-gid=dba owner-mode=0640 dev-name=raw_asm4

# disklabel –create /dev/rdisk<n>s2 owner-uid=oracle
> group-gid=dba owner-mode=0640 dev-name=raw_asm5

# disklabel –create /dev/rdisk<n>s2 owner-uid=oracle
> group-gid=dba owner-mode=0640 dev-name=raw_asm6

# ls /dev/raw*
# ls /dev/rraw*
```

Step Five – Wipe Out Partition Header Blocks

It always a best practice to zero out the first few megabytes of any raw partition to be used for ASM. In the interest of following such best practices and ensuring a "clean slate each time", please perform the following commands.

```
# dd if=/dev/zero of=/dev/raw_asm1 bs=1024 count=10240
# dd if=/dev/zero of=/dev/raw_asm2 bs=1024 count=10240
# dd if=/dev/zero of=/dev/raw_asm3 bs=1024 count=10240
# dd if=/dev/zero of=/dev/raw_asm4 bs=1024 count=10240
# dd if=/dev/zero of=/dev/raw_asm5 bs=1024 count=10240
# dd if=/dev/zero of=/dev/raw_asm6 bs=1024 count=10240
```

At this point, these are six (6) raw devices in the `/dev` directory, ready to be used by Oracle ASM.

What to do after a reboot?

The "pseudo disk drive-to-raw disk drive" device mappings are lost during a subsequent reboot. In order to "reattach" and "access" your disk image raw pseudo disk drives, perform the following commands:

```
$ su –
# cd /Volumes
# hdiutil attach asmDisk1.dmg
# hdiutil attach asmDisk2.dmg
# hdiutil attach asmDisk3.dmg
# hdiutil attach asmDisk4.dmg
# hdiutil attach asmDisk5.dmg
# hdiutil attach asmDisk6.dmg
```

This forces the operating system to "see" the disk images, next attach them to the operating system using the `Disk Utility` tool as previously documented.

And before anyone asks, no, there is no easy way to automate this task – perhaps a savvy Apple Script coder could come up with a solution.

Redhat Linux Raw Pseudo Disk Drive

The Linux world does not have the elegant and very useful Apple OS X Aqua interface, thus the steps to create pseudo disk drives is a bit more straight forward. Sadly, it does not offer the dynamic abilities of Apple OS X and any persistency of raw devices is done by writing scripts, which need to be run at each reboot.

Step One – Create Files

The `dd` command is used to create 2GB files "full of zeros", which will be used as pseudo disk drives by Oracle ASM.

```
$ su –
# mkdir /virtual
# dd if=/dev/zero of=/virtual/disk1 bs=1024 count=2097152
# dd if=/dev/zero of=/virtual/disk2 bs=1024 count=2097152
# dd if=/dev/zero of=/virtual/disk3 bs=1024 count=2097152
# dd if=/dev/zero of=/virtual/disk4 bs=1024 count=2097152
# dd if=/dev/zero of=/virtual/disk5 bs=1024 count=2097152
# dd if=/dev/zero of=/virtual/disk6 bs=1024 count=2097152
```

Step Two – Associate Files to Loop Back Devices

Redhat Linux allows for the association of filesystem files, like those created in the previous step, to be associated with a character block device by the use of the `losetup` command.

```
# chown <oracle_uid>:<oracle_gid> /virtual/disk*

# losetup /dev/loop1 /virtual/disk1
# losetup /dev/loop2 /virtual/disk2
# losetup /dev/loop3 /virtual/disk3
# losetup /dev/loop4 /virtual/disk4
# losetup /dev/loop5 /virtual/disk5
# losetup /dev/loop6 /virtual/disk6
```

Step Three – Associate Loop Back Devices to Raw Devices

The final step is to associate the character block device with a raw block device. Use the `raw` command to make the association.

```
# raw /dev/raw/raw1 /dev/loop1
# raw /dev/raw/raw2 /dev/loop2
# raw /dev/raw/raw3 /dev/loop3
# raw /dev/raw/raw4 /dev/loop4
# raw /dev/raw/raw5 /dev/loop5
# raw /dev/raw/raw6 /dev/loop6
# chown oracle:dba /dev/raw/raw[1-6]
```

Step Four – Wipe Out Partition Header Blocks

It always a best practice to zero out the first few megabytes of any raw partition to be used for Oracle ASM. In the interest of following such best practices and ensuring a "clean slate each time", please perform the following commands.

```
# dd if=/dev/zero of=/dev/loop1 bs=1024 count=10240
# dd if=/dev/zero of=/dev/loop2 bs=1024 count=10240
# dd if=/dev/zero of=/dev/loop3 bs=1024 count=10240
# dd if=/dev/zero of=/dev/loop4 bs=1024 count=10240
# dd if=/dev/zero of=/dev/loop5 bs=1024 count=10240
# dd if=/dev/zero of=/dev/loop6 bs=1024 count=10240
```

At this point, these are six (6) raw devices in the `/dev` directory, ready to be used by Oracle ASM.

What to do after a reboot?

It should be noted that while the drives files are preserved during a reboot, the assignments to loopback devices and associations to raw devices are lost. If it is desired to have these assignments survive reboots persistently, the place all of the losetup and raw commands from the above example in a script and place into /etc/rc.local.

Closing Comments

The ability to play with Oracle 10g ASM is vital for any DBA who wishes to improve their skill set. By using the techniques outlined in this paper, the DBA can safely play and learn within the confines of their own sandbox environment. Oracle ASM offers many subtle nuances and it is only through experience gained can today's Oracle 10g DBA be able to deploy ASM and have the skill set to maintain, tune and recovery from problems.

So, the next step is to fire up the Database Configuration Assistant (DBCA), build a database and be sure to build it using ASM for the storage. Once a viable database is built and running, using the Enterprise Manager dbconsole, or Grid Control if so inclined, to "play" with the Oracle 10g ASM features.

Advanced Technology Services, Inc.
Oracle Technology Delivered

“Learning ASM Using a Single Disk Drive
on the Apple OS X and Linux Platforms”
July 2005 (Rev. 2)
by Kent Stroker

Advanced Technology Services, Inc.
Seattle, WA

www.advtechsvcs.com