CLONING ORACLE DATABASES ON PURE

A STEP BY STEP GUIDE

PURESTORAGE®

TABLE OF CONTENTS

INTRODUCTION

Anywhere there is a database, there will eventually be a need to make a copy, or clone of it. A copy of the production database, for example, may be cloned for development testing, or reporting, or simply for backup. Or a QA environment may need to restore an environment to a specific point in time to perform testing. While the cloning process has traditionally been tedious, time consuming, and subject to errors, Pure Storage Snapshots make cloning databases quick and easy.

Pure Storage FlashArray comes with a snapshotting capability that users can leverage for a variety of purposes, including fast and simple cloning of databases. This document describes the database cloning procedure for Oracle databases running on Linux, using both Automatic Storage Management (ASM) and file system based data files. The servers used in the presentation were running Red Hat Enterprise Linux Server release 6.3 (Santiago), but the procedure is the same for any Linux version.

Our snapshots are crash consistent, even across multiple LUNs. In other words, the data in the snapshot is the same as the situation after a power outage. Every write I/O that has been acknowledged to the host is guaranteed to be present in the snapshot, and the data on all LUNs will be from the same point in time.

Since snapshots are crash consistent, there is no need to put an Oracle database into backup mode before performing the snapshot. Therefore, the database does not even need to have media recovery enabled; i.e., the database does not need to be in ARCHIVELOG mode. Furthermore, Oracle Corporation supports storage snapshots without backup mode enabled if the snapshot meets the following three criteria:

- The database is crash consistent during the snapshot.
- The snapshot preserves the write order for each file.
- The snapshot technology stores the time at which the snapshot was completed.

These conditions are described in the Oracle documentation: http://docs.oracle.com/cd/E16655_01/backup.121/e17630/osbackup.htm#BRADV90019.

Another distinguishing feature of Pure Storage snapshots is that the time required to perform a snapshot is the same regardless of how many LUNs are included, and regardless of the size of the LUNs. Snapshots always take a second or less to perform.

These features are possible because the Pure Storage implementation of snapshots is unlike that of any other storage vendor. We leverage our metadata to create snapshots that are nearly instantaneous and consume very little space.

ENVIRONMENT PREPARATION

The database cloning procedure assumes the source and destination database server are both connected to the same Pure Storage FlashArray. The source and destination databases are both local to the same array.

Linux User and Group Setup

Although it is not strictly required, the user and group IDs should be the same on both servers. Otherwise, it will be necessary to perform **chown** commands on files and directories on the destination server so that ownership is correct.

The values we used are as follows:

USERS

- oracle :   501
- grid :   502

GROUPS

- oinstall :   501
- dba :   502
- oper :   503 (ASM-specific)
- asmadmin :   505 (ASM-specific)
- asmdba :   506 (ASM-specific)
- asmoper :   507 (ASM-specific)

Oracle Software

The oracle software version and oracle software tree should be identical on both servers. The specific version of Oracle software is not significant. The configuration we used is as follows.

GRID INFRASTRUCTURE (FOR ASM)

ORACLE_BASE : /u01/app/oracle
ORACLE_HOME : /u01/app/oracle/product/12.1.0.1/grid

DATABASE SOFTWARE

ORACLE_BASE : /u01/app/oracle
ORACLE_HOME : /u01/app/oracle/product/12.1.0.1/dbhome_1

INIT.ORA File

The source database can use either an INIT.ORA file (pfile) or an spfile, but for simplicity we use an INIT.ORA file on the destination server. You will probably want to change the values for local_listener and you may want to change the ORACLE_SID (details below). For these reasons, an editable INIT.ORA file in $ORACLE_HOME/dbs is most convenient.

ABOUT THE DEMONSTRATION ENVIRONMENT

We used two database servers and a Pure Storage FlashArray for this demonstration, as follows:

SOURCE DATABASE SERVER

- SuperMicro 6016XT-TF
- Red Hat Enterprise Linux Server Release 6.3 (Santiago)
- 2 sockets / 12 cores
- Intel Xeon E5645 @ 2.40GHz
- 96GB Memory
- Oracle Database 12c Enterprise Edition Release 12.1.0.1.0

TARGET DATABASE SERVER

- HP DL580 G7
- Red Hat Enterprise Linux Server Release 6.3 (Santiago)
- 4 sockets / 40 cores
- Intel Xeon E7-4870 @ 2.40GHz
- 512GB Memory
- Oracle Database 12c Enterprise Edition Release 12.1.0.1

PURE STORAGE FLASHARRAY

- FlashArray
- 11 TB Raw Storage
- Purity Operating Environment

INITIAL SNAPSHOT AND VOLUME COPY: FILE SYSTEM

The source Oracle database was deployed on two LUNS as follows:

- **falcondata** : 250GB volume holding controlfiles and datafiles for all tablespaces
- **falconredo** : 50GB volume holding all online redo logs for the database

Before we can clone the database to the target database server, we need to create volumes of the same sizes and present them to the target database server. In our example we created:

- **fsoradata** : 250GB volume to which we copy the snapshot of falcondata
- **fsoraredo** : 50GB volume to which we copy the snapshot of falconredo

The detailed steps for creating volumes and presenting them to a host are available in our document **Best Practices for Oracle on Pure Storage**.

These volumes should use the same mount points as their counterparts on the source database (in our case **/oradata** and **/oraredo**). There is a one-time need to create entries for these volumes in /etc/fstab. In our example the entries appear as follows:

```
#
# /etc/fstab
# Created by anaconda on Tue Sep 25 16:08:44 2012
#
# Accessible filesystems, by reference, are maintained under '/dev/disk'
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info
#
UUID=149604f7-a6f1-4cf8-b7f8-ca90d7eec668 /                   ext4    defaults        1 1
UUID=f039ece7-701c-4a05-ad6b-7d9d7825cee8 /boot               ext4    defaults        1 2
UUID=c08a6ce8-2460-4307-b9a1-13d74e2db9c5 swap                swap    defaults        0 0
UUID=42ddf292-c500-4c22-85e3-dbd83a74c50c /u01                ext4    discard,noatime 1 2
UUID=b433acc9-3014-44ba-b790-bc8ce78337f0 /oradata            ext4    discard,noatime 1 2
UUID=f3c7cf02-b839-4942-ac82-3109c029b76c /oraredo            ext4    discard,noatime 1 2
UUID=e418f79b-9bf3-48ea-bea5-a1328adcd400 /pgdata             ext4    discard,noatime 1 2
tmpfs                   /dev/shm              tmpfs   defaults        0 0
devpts                  /dev/pts              devpts  gid=5,mode=620  0 0
sysfs                   /sys                  sysfs   defaults        0 0
proc                    /proc                 proc    defaults        0 0
```

After the initial volume and file system creation, the **UUIDs** will not change even after you copy a snapshot on to the volume. There is no need to modify the **/etc/fstab** file after the initial setup.

DATABASE CLONING PROCEDURE: FILE SYSTEM

Once the initial setup described above is complete, you are ready to clone your database. The essential steps are:

1. **Verify** that target volume(s) on target database server are not mounted
2. **Create** snapshot of source volume(s)
3. **Copy** snapshot(s) to corresponding target volume(s)
4. **Mount** target volume(s) on target database server
5. **Start** database instance on target database server

Steps in Detail

1. Verify that target volume(s) on target database server are not mounted

We check to see if **/oradata** and/or **/oraredo** are mounted using the **df** command and unmount them if they are:

```
# df -h | egrep "oradata|oraredo"
/dev/mapper/mpathbn    247G  141G    94G  61% /oradata
/dev/mapper/mpathbm     50G   41G  6.6G  87% /oraredo
#
# umount /oradata /oraredo
#
# df -h | egrep "oradata|oraredo"
#
```

2. Create snapshot(s) of source volume(s)

In our example, we will create snapshots of LUNs **fsoradata** and **fsoraredo**. Check both of these LUNs in the GUI and click the **Create Snapshots** icon:

At this point you will be prompted to provide a suffix for the snapshot. The suffix will allow you to identify which snapshots were taken together. Provide a suffix and click **Create**.

Command line equivalent:

```
pureuser@pure-demo> purevol snap --suffix DEMOSNAP-01 falcondata falconredo
Name                    Size   Source      Created               Serial
falcondata.DEMOSNAP-01  250G   falcondata  2014-01-31 14:00:50 PST  BB22E766DD857943000130F7
falconredo.DEMOSNAP-01  50G    falconredo  2014-01-31 14:00:50 PST  BB22E766DD857943000130F8
```

At this point you will see the snapshots **falcondata.DEMOSNAP-01** and **falconredo.DEMOSNAP-01** in the GUI.

PURESTORAGE®

3. Copy snapshot(s) to corresponding target volume(s)

Next we copy the snapshots to the corresponding volumes that are connected to the target server.

It's necessary to use the CLI to copy a snapshot to an existing volume:

```
pureuser@pure-demo> purevol copy --force falcondata.DEMOSNAP-01 fsoradata
Name        Size  Source      Created              Serial
fsoradata   250G  falcondata  2014-01-31 14:00:50 PST  BB22E766DD8579430001001C
pureuser@pure-demo> purevol copy --force falconredo.DEMOSNAP-01 fsoraredo
Name        Size  Source      Created              Serial
fsoraredo   50G   falconredo  2014-01-31 14:00:50 PST  BB22E766DD8579430001001D
```

4. Mount target volume(s) on target database server

Now that the contents of the volumes have been replaced with the data in the snapshots, the volumes can be mounted.

```
[root@warthog ~]# df -h
Filesystem            Size  Used Avail Use% Mounted on
/dev/sda2             185G  158G   19G  90% /
tmpfs                 253G  727M  252G   1% /dev/shm
/dev/sda1             485M   39M  421M   9% /boot
/dev/mapper/mpathaa  1008G  171G  786G  18% /u01
/dev/mapper/mpathdo   493G  155G  313G  34% /pgdata
[root@warthog ~]# mount -a
[root@warthog ~]# df -h
Filesystem            Size  Used Avail Use% Mounted on
/dev/sda2             185G  158G   19G  90% /
tmpfs                 253G  727M  252G   1% /dev/shm
/dev/sda1             485M   39M  421M   9% /boot
/dev/mapper/mpathaa  1008G  171G  786G  18% /u01
/dev/mapper/mpathdo   493G  155G  313G  34% /pgdata
/dev/mapper/mpathbn   247G   66G  169G  28% /oradata
/dev/mapper/mpathbm    50G   27G   21G  56% /oraredo
```

5. Start database instance on target database server

```
14:21:52 [demo] oracle@warthog 1063$ sqlplus / as sysdba

SQL*Plus: Release 12.1.0.1.0 Production on Fri Jan 31 14:21:57 2014

Copyright (c) 1982, 2013, Oracle.  All rights reserved.

Connected to an idle instance.

SQL> startup pfile=/u01/app/oracle/product/12.1.0.1/dbhome_1/dbs/initDEMOFS.ora
ORACLE instance started.

Total System Global Area 2137886720 bytes
Fixed Size                  2683632 bytes
Variable Size            1660947728 bytes
Database Buffers          419430400 bytes
Redo Buffers               54824960 bytes
Database mounted.
Database opened.
SQL> exit
Disconnected from Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 - 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real Application Testing options
```

INITIAL SNAPSHOT AND VOLUME COPY: ASM

Our ASM demonstration database consists of two ASM disk groups: **DGDEMODATA** and **DGDEMOREDO**, as follows:

```
14:40:53 system@demo.falcon SQL> l
  1  SELECT      g.name groupname,
  2      d.path,
  3      d.name diskname,
  4      g.sector_size,
  5      g.block_size,
  6      g.allocation_unit_size au,
  7      g.state,
  8      g.total_mb/1024 total_gig,
  9      d.total_mb/1024 dtotal_gig,
 10      g.free_mb/1024 free_gig,
 11      d.free_mb/1024 dfree_gig
 12  FROM       v$asm_diskgroup g,
 13      v$asm_disk d
 14  WHERE      d.group_number = g.group_number
 15* ORDER BY g.name, d.disk_number
14:41:03 system@demo.falcon SQL> /
                                                          Group    Disk   Group    Disk
ASM                                     Sector Block      Total   Total    Free    Free
Disk Group  LUN              Disk Name   Size  Size AU Size STATE   GB      GB      GB      GB
----------- ---------------- ----------- ----- ----- -------- --------- ------- ------- ------- -------
DGDEMODATA  /dev/dm-0        DGDEMODATA_0000   512  4096  1048576 CONNECTED   750     250     512     171
            /dev/dm-3        DGDEMODATA_0001   512  4096  1048576 CONNECTED           250             171
            /dev/dm-4        DGDEMODATA_0002   512  4096  1048576 CONNECTED           250             171

DGDEMOREDO  /dev/dm-7        DGDEMOREDO_0000   512  4096  1048576 CONNECTED   200     100     100      50
            /dev/dm-5        DGDEMOREDO_0001   512  4096  1048576 CONNECTED           100              50

5 rows selected.
```

DGDEMODATA consists of three 250GB ASM disks, and **DGDEMOREDO** has two 100GB ASM disks. Each ASM disk corresponds to a LUN on the Pure Storage FlashArray.

The one-time setup for the ASM disk groups on the target database server entails creating a target LUN for each source LUN, and configuring these LUNs for use with ASM with **udev** or **multipath.conf**. Please refer to our document **Best Practices for Oracle on Pure Storage** for details. The **udev** rules for the LUNs in this demonstration are as follows:

```
# ASM snap demo.
ENV{DM_UUID}=="mpath-3624a9370bb22e766dd85794300012eb9", OWNER:="grid", GROUP:="asmadmin", MOD
ENV{DM_UUID}=="mpath-3624a9370bb22e766dd85794300012eba", OWNER:="grid", GROUP:="asmadmin", MOD
ENV{DM_UUID}=="mpath-3624a9370bb22e766dd85794300012ebb", OWNER:="grid", GROUP:="asmadmin", MOD
ENV{DM_UUID}=="mpath-3624a9370bb22e766dd85794300012ebc", OWNER:="grid", GROUP:="asmadmin", MOD
ENV{DM_UUID}=="mpath-3624a9370bb22e766dd85794300012ebd", OWNER:="grid", GROUP:="asmadmin", MOD
```

The UUIDs of the LUNs will not change when snapshots are copied to these volumes, so there is no need to modify the **udev** rules after the initial configuration.

DATABASE CLONING PROCEDURE: ASM

Once the initial setup described above is complete, you are ready to clone your database. The essential steps are:

1. **Dismount** the ASM disk groups on the target database server
2. **Create** snapshot(s) of source volume(s)
3. **Copy** snapshot(s) to corresponding target volume(s)
4. **Mount** the ASM disk groups on target database server
5. **Start** database instance on target database server

As you can see, the only difference between the ASM scenario and the file system scenario is step 1 (unmount the ASM disk groups instead of the file system) and step 4 (mount the ASM disk groups instead of the file systems). The other steps are the same.

Steps in Detail

1. Dismount the ASM disk groups on the target database server

```
$ sqlplus / as sysasm

SQL*Plus: Release 12.1.0.1.0 Production on Mon Mar 10 16:57:47 2014

Copyright (c) 1982, 2013, Oracle.  All rights reserved.


Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 - 64bit Production
With the Automatic Storage Management option

SQL> ALTER DISKGROUP dgdemodata DISMOUNT;

Diskgroup altered.

SQL> ALTER DISKGROUP dgdemoredo DISMOUNT;

Diskgroup altered.
```

2. Create snapshot(s) of source volume(s)

```
pureuser@pure-demo> purevol snap --suffix DEMO-ASM dg_demodata01 dg_demodata02 dg_demodata03 dg_demoredo01 dg_demoredo02
Name                    Size  Source         Created               Serial
dg_demodata01.DEMO-ASM  250G  dg_demodata01  2014-03-10 17:05:50 PDT  BB22E766DD857943000131EF
dg_demodata02.DEMO-ASM  250G  dg_demodata02  2014-03-10 17:05:50 PDT  BB22E766DD857943000131EB
dg_demodata03.DEMO-ASM  250G  dg_demodata03  2014-03-10 17:05:50 PDT  BB22E766DD857943000131ED
dg_demoredo01.DEMO-ASM  100G  dg_demoredo01  2014-03-10 17:05:50 PDT  BB22E766DD857943000131EC
dg_demoredo02.DEMO-ASM  100G  dg_demoredo02  2014-03-10 17:05:50 PDT  BB22E766DD857943000131EE
```

3. Copy snapshot(s) to corresponding target volume(s)

```
pureuser@pure-demo> purevol copy --force dg_demodata01.DEMO-ASM dg_demodata01w
Name            Size  Source        Created                Serial
dg_demodata01w  250G  dg_demodata01 2014-03-10 17:05:50 PDT  BB22E766DD85794300012EB9
pureuser@pure-demo> purevol copy --force dg_demodata02.DEMO-ASM dg_demodata02w
Name            Size  Source        Created                Serial
dg_demodata02w  250G  dg_demodata02 2014-03-10 17:05:50 PDT  BB22E766DD85794300012EBA
pureuser@pure-demo> purevol copy --force dg_demodata03.DEMO-ASM dg_demodata03w
Name            Size  Source        Created                Serial
dg_demodata03w  250G  dg_demodata03 2014-03-10 17:05:50 PDT  BB22E766DD85794300012EBB
pureuser@pure-demo> purevol copy --force dg_demoredo01.DEMO-ASM dg_demoredo01w
Name            Size  Source        Created                Serial
dg_demoredo01w  100G  dg_demoredo01 2014-03-10 17:05:50 PDT  BB22E766DD85794300012EBC
pureuser@pure-demo> purevol copy --force dg_demoredo02.DEMO-ASM dg_demoredo02w
Name            Size  Source        Created                Serial
dg_demoredo02w  100G  dg_demoredo02 2014-03-10 17:05:50 PDT  BB22E766DD85794300012EBD
```

4. Mount the ASM disk group(s) on the target database server

```
$ sqlplus / as sysasm

SQL*Plus: Release 12.1.0.1.0 Production on Mon Mar 10 17:16:24 2014

Copyright (c) 1982, 2013, Oracle.  All rights reserved.


Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 - 64bit Production
With the Automatic Storage Management option

SQL>  ALTER DISKGROUP dgdemodata MOUNT FORCE;

Diskgroup altered.

SQL> ALTER DISKGROUP dgdemoredo MOUNT FORCE;

Diskgroup altered.
```

5. Start the database on the target database server

```
$ sqlplus / as sysdba

SQL*Plus: Release 12.1.0.1.0 Production on Mon Mar 10 17:20:21 2014

Copyright (c) 1982, 2013, Oracle.  All rights reserved.

Connected to an idle instance.

SQL> startup
ORACLE instance started.

Total System Global Area 2137886720 bytes
Fixed Size                  2683632 bytes
Variable Size            1744833808 bytes
Database Buffers          335544320 bytes
Redo Buffers               54824960 bytes
Database mounted.
Database opened.
```