**Utilizing NAS Flex-Clone Capabilities for Partial or Full Database Recoveries**

Business Case

With the proliferation of the Shared LION architecture and also as our existing application-dedicated LION databases continue to grow in size, it is becoming increasingly important to be able to restore logical parts of a particular database, whether it be a particular application table or schema or some variant of those.

The traditional Oracle backup and recovery techniques do not lend themselves to partial database recoveries where the data of interest is defined at the logical application, not physical, level and there is a requirement to be able to go back quite a bit in time for meeting the RPO objectives.

Premise of the Technique

For this reason outlined above, there is now a new method that allows to restore application data comprising only part of a database, without affecting the rest of the existing production environment. The basic premise of this method involves setting up a Restore VM with storage allocated on a different (non-production) filer and flex-cloning a production snap off of the production backup filer to the Restore VM.

Restore VM Considerations

For typical LION configurations, as of the time of this writing, it is recommended that the Restore VM be a virtual machine LION stack. The restore environment does not have to be RAC, whether or not the source production database is RAC. A smaller memory footprint, such as 8GB should be sufficient. The Restore VM should have the matching version of the OS as the source production environment and have Oracle database s/w installed on it, also of the matching version. The Restore VM may or may not be an environment dedicated to its purpose and a particular business application or sets of applications. This is largely an operational decision that hinges on application requirements for RTO, security and other SLA obligations. Having said that, it would be highly recommended to dedicate the VM to the flex-cloning purpose. The flex-clone license needs to cover the backup (secondary) filer and does not need to cover the primary or the Restore VM filers. Here are some additional considerations for allocating the space on the Restore VM:

|  |  |  |
| --- | --- | --- |
| **Mount Point** | **Sizing Considerations** | **Mounting Considerations** |
| /n0[n]/oradata[n] | The size should account for the online redo logs and the controlfile of the largest potential target database. 20-50GB or larger, depending on the database needs. | The underlying NAS volume should be provisioned at the time the VM is built, but should not be mounted until a flex-clone is done. At the time of flex-cloning, the directory with the proper path should be created on the VM and the underlying volume mounted on it. |
| /s0[n]/oradata[n] | The directory should be pre-created on the Restore VM. **NOTE:** ensure the primary server’s naming convention is used to avoid having to move database files with the DB\_FILE\_NAME\_CONVERT or ALTER DATABASE RENAME FILE techniques. | The underlying NAS volume is unknown until the time of flex-cloning and therefore no mounting can occur. No separate NAS volume needs to be provisioned at the time of the VM build – all snap volumes will be the flex-cloned ones. At the time of flex-cloning, the directory with the proper path should be created on the VM. When the flex-clone is subsequently attempted, it will automatically use this path to mount the flex-cloned volume on it. The process is facilitated by the NetApp NFSClone utility discussed later in the document. |
| /n0[n]/oraadmin[n] | The size should account for all admin type content of the target database. 10GB or larger, depending on the database needs. | The underlying NAS volume should be provisioned at the time the VM is built, but should not be mounted until a flex-clone is done. At the time of flex-cloning, the directory with the proper path should be created on the VM and the underlying volume mounted on it. |
| /n0[n]/oraarch[n] | The directory should be pre-created on the Restore VM. **NOTE:** ensure the primary server’s naming convention is used to avoid having to specify each archived log’s path during the point-in-time recovery. | <UNDER CONSIDERATION> |

The NFSClone Utility

NetApp provides a useful **nfsclone** utility to automate and obfuscate the nuanced aspects of flex-cloning. This utility is a perl script compiled into an exe binary for x86 systems. It should be run on the restore VM where the flex-cloned volume needs to be presented, as root. The utility can be currently obtained at the following address:

<http://nerstrand.int.westgroup.net/netapp/nfsclone/>

Its usage is:

Usage: nfsclone.pl -o connect -h storagehost -u storageuser

-m mountpoint -p [/vol/volname | /vol/volname/path]

-s snapshot [-i staticIP] [-v]

Usage: nfsclone.pl -o disconnect -h storagehost -u storageuser

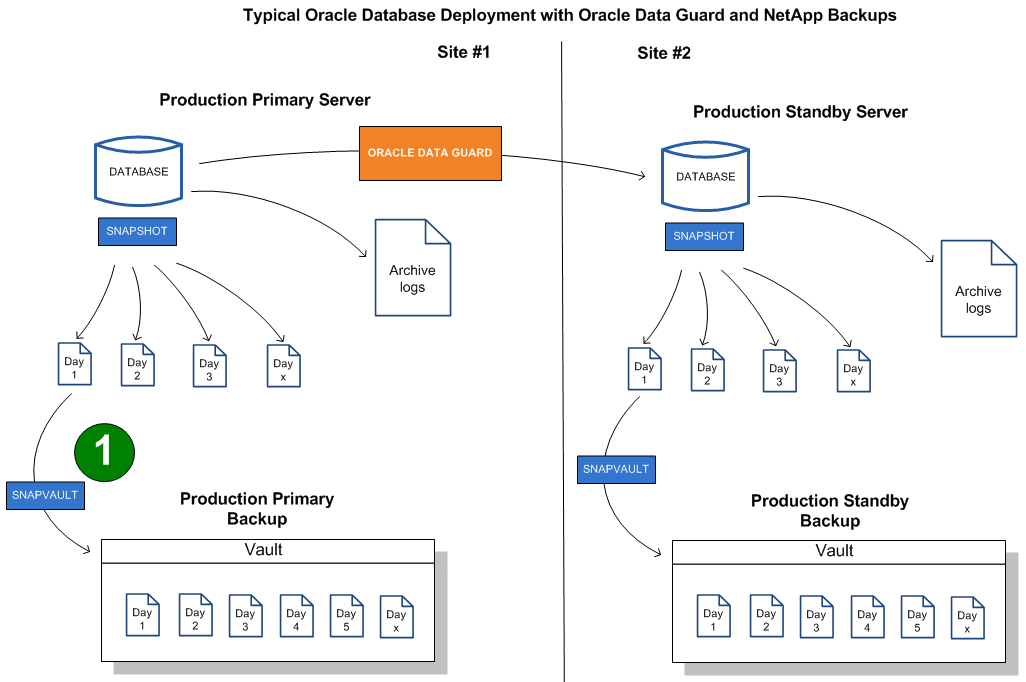
-m mountpoint [-v]

The –o option specifies whether to mount or unmount a flex-cloned volume from the Restore VM. The –h option specifies the secondary (backup) filer where the primary’s snapshots are being vaulted. The –u option specifies the user that the utility will authenticate with, connecting to the secondary (backup) filer from the Restore VM. The –m option specifies the directory name pre-created on the Restore VM that the flex-cloned volume will be mounted on. The –p option specifies the snapvaulted backup destination on the secondary (backup) filer. The –s option specifies the snapshot name, as it is known to the backup (snapvault) filer.

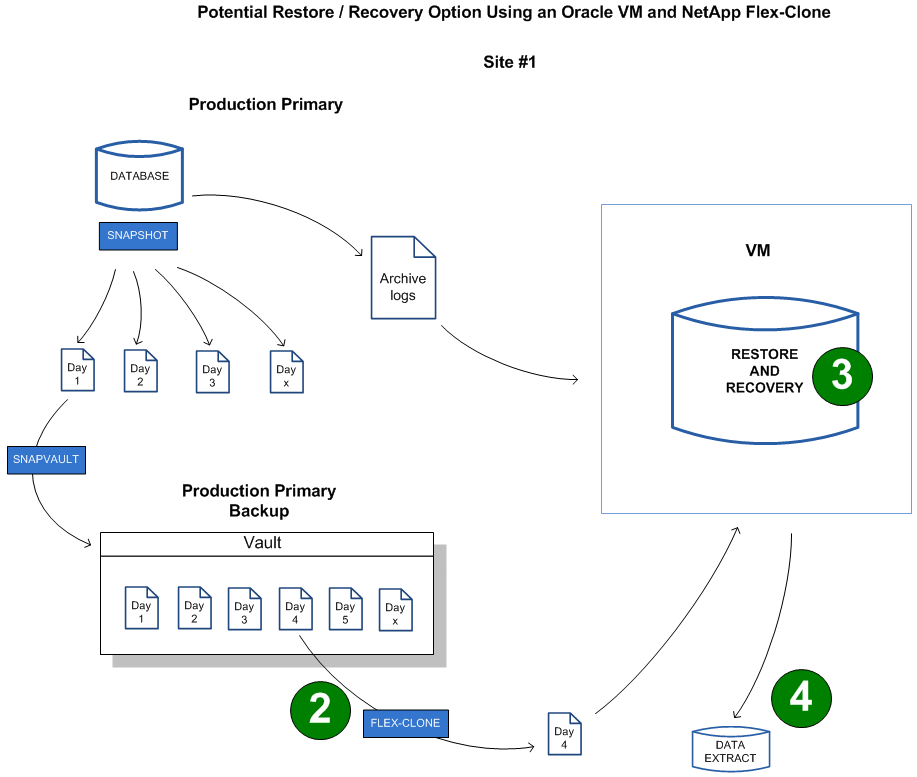
Process Overview

Below are the diagrams depicting the cloning and data restoration steps.

Regular Operation:



Restore Operation:



The process can be broken down into the following steps:

Step

Take the snapshot backup of the primary database. Ensure the snapshots are being properly taken and properly vaulted on a defined schedule. The latter can be checked with the following command, run on the primary site as oracle user:

**ssh lab-g7tst-s001 -i /home/oracle/backup/nssdba.snapvault "snapvault status"**

Snapvault is ON.

Source Destination State Lag Status

lab-g7tst-s001:/vol/fnr\_egrc\_s01ora1\_snap lab-nasbkp-001:/vol/sv\_7\_fnr\_egrc\_s01ora1\_snap/1 Source 21:24:26 Idle

lab-g7tst-s001:/vol/fnr\_egrc\_s01oraadm1\_snap lab-nasbkp-001:/vol/sv\_7\_fnr\_egrc\_s01oraadm1\_snap/1 Source 08:27:12 Idle

lab-g7tst-s001:/vol/fnr\_egrc\_s02ora1\_snap lab-nasbkp-001:/vol/sv\_7\_fnr\_egrc\_s02ora1\_snap/1 Source 594:36:04 Idle

lab-g7tst-s001:/vol/fnr\_egrc\_s02oraadm1\_snap lab-nasbkp-001:/vol/sv\_7\_fnr\_egrc\_s02oraadm1\_snap/1 Source 08:27:12 Idle

**Note**: the volume being snapped has to be vaulted in order to be “clonable”. It is recommended to coordinate vaulting schedules with the storage team and strive to shorten the time-window between the backup time and the time the volume is vaulted. The flex-cloning technique should be very useful for satisfying RPOs dealing with having to go back very far and moderately far back in time. For recoveries to less than 24 hours back in time, other techniques may need to be employed, such as a standby snapshot, flashback table/database, etc…

**Note**: you may need to change the path of the nssdba.snapvault ssh key location.

Also, ssh access to the secondary (backup) filer should work and can be checked with the following command, run on the primary site as oracle user:

**ssh lab-nasbkp-001 -i /home/oracle/backup/nssdba.snapvault "snap list"**

Look for the output section concerning the volume of interest. In our case it is:

**Volume sv\_7\_fnr\_egrc\_s01ora1\_snap**

working...

%/used %/total date name

---------- ---------- ------------ --------

0% ( 0%) 0% ( 0%) Oct 25 12:07 sv\_fnr\_egrc\_s01ora1\_snap\_s01oradata1.0

1% ( 1%) 0% ( 0%) Oct 25 12:06 lab-nasbkp-001(1873802807)\_sv\_7\_fnr\_egrc\_s01ora1\_snap-base.0 (busy,snapvault)

6% ( 5%) 0% ( 0%) Oct 25 12:06 sv\_7\_fnr\_egrc\_s01ora1\_snap\_s01oradata1.0

7% ( 1%) 0% ( 0%) Oct 24 12:07 sv\_fnr\_egrc\_s01ora1\_snap\_s01oradata1.1

20% (15%) 0% ( 0%) Oct 24 12:06 sv\_7\_fnr\_egrc\_s01ora1\_snap\_s01oradata1.1

20% ( 0%) 0% ( 0%) Oct 23 12:07 sv\_fnr\_egrc\_s01ora1\_snap\_s01oradata1.2

21% ( 0%) 0% ( 0%) Oct 23 12:06 sv\_7\_fnr\_egrc\_s01ora1\_snap\_s01oradata1.2

21% ( 0%) 0% ( 0%) Oct 22 12:07 sv\_fnr\_egrc\_s01ora1\_snap\_s01oradata1.3

21% ( 0%) 0% ( 0%) Oct 22 12:06 sv\_7\_fnr\_egrc\_s01ora1\_snap\_s01oradata1.3

21% ( 0%) 0% ( 0%) Oct 21 12:07 sv\_fnr\_egrc\_s01ora1\_snap\_s01oradata1.4

22% ( 0%) 0% ( 0%) Oct 21 12:06 sv\_7\_fnr\_egrc\_s01ora1\_snap\_s01oradata1.4

22% ( 0%) 0% ( 0%) Oct 20 12:07 sv\_fnr\_egrc\_s01ora1\_snap\_s01oradata1.5

22% ( 0%) 0% ( 0%) Oct 20 12:06 sv\_7\_fnr\_egrc\_s01ora1\_snap\_s01oradata1.5

22% ( 0%) 0% ( 0%) Oct 19 12:07 sv\_fnr\_egrc\_s01ora1\_snap\_s01oradata1.6

23% ( 0%) 0% ( 0%) Oct 19 12:06 sv\_7\_fnr\_egrc\_s01ora1\_snap\_s01oradata1.6

Step

Initiate a flex-clone on the Storage VM using the following command, run as root:

**./nfsclone -o connect -h lab-nasbkp-001 -u nfsclone -m /s01/oradata1 -p /vol/sv\_7\_fnr\_egrc\_s01ora1\_snap/1 -s "sv\_7\_fnr\_egrc\_s01ora1\_snap\_s01oradata1.0"**

Enter password for nfsclone@lab-nasbkp-001:

Thu Oct 25 15:19:18 2012: Checking connectivity for nfsclone@lab-nasbkp-001.

Thu Oct 25 15:19:18 2012: Connectivity OK - NetApp Release 8.1.1 7-Mode: Mon Jul 30 12:49:46 PDT 2012 Multistore.

Thu Oct 25 15:19:18 2012: Checking storage system configuration.

You will be asked for a password to authenticate access to the secondary (backup)filer. Storage support should be able to provide the password to DBAs executing the restore. Note that the snap volume name provided with the –s argument should be enclosed in double-quotes and should match the Name column’s data of the “snap list” command run on the primary site in the previous step.

Ensure that the directories under /s0[n] are properly owned. Execute as root on the Restore VM:

cd /s01; chown oracle:oinstall oraadmin1 oradata1

It is possible that the nfsclone utility will restore the snapshot in another subdirectory carrying the q-tree name:

/s01/oradata1/s01oradata1> ls

poc012

In this case, just move the poc012 datafile directory under its original location. Execute the following on the Storage VM as oracle:

cd /s01/oradata1/s01oradata1; mv poc012 ../.

/s01/oradata1> ls

poc012 s01oradata1

Step

Restore and recover the database on the Restore VM. To accomplish this step, you will need the pfile and the backup controlfile binary copy, both of which should have been backed up during the primary’s snapshot backup. It is important that the backup controlfile binary copy was taken after the snap itself had already been taken, to ensure its SCN is newer than those in the headers of the restored snapshot datafiles, allowing for a point-in-time roll-forward.

Here is a typical backed up pfile that we will use to bring up our new database on the Restore VM and the parameters that we will eliminate (crossed out) or change (highlighted) because they are not appropriate for the new environment. Copy from the backup location to the Storage VM:

scp initpoc012a1.ora.bkp xtrm-lionvm-01:/u01/app/oracle/product/11.2.0.3/db/dbs/initpoc012a.ora

Edit:

~~poc012a1.\_\_db\_cache\_size=27044872192~~

~~poc012a2.\_\_db\_cache\_size=27044872192~~

~~poc012a1.\_\_java\_pool\_size=67108864~~

~~poc012a2.\_\_java\_pool\_size=67108864~~

~~poc012a1.\_\_large\_pool\_size=67108864~~

~~poc012a2.\_\_large\_pool\_size=67108864~~

~~poc012a1.\_\_pga\_aggregate\_target=10200547328~~

~~poc012a2.\_\_pga\_aggregate\_target=10200547328~~

~~poc012a1.\_\_sga\_target=30467424256~~

~~poc012a2.\_\_sga\_target=30467424256~~

~~poc012a1.\_\_shared\_io\_pool\_size=0~~

~~poc012a2.\_\_shared\_io\_pool\_size=0~~

~~poc012a1.\_\_shared\_pool\_size=3087007744~~

~~poc012a2.\_\_shared\_pool\_size=3087007744~~

~~poc012a1.\_\_streams\_pool\_size=0~~

~~poc012a2.\_\_streams\_pool\_size=0~~

\*.audit\_file\_dest='/u01/app/oracle/admin/poc012a/adump'

\*.audit\_trail='NONE'

~~\*.cluster\_database=true~~

\*.compatible='11.2.0.0.0'

\*.control\_files='/s01/oradata1/poc012/POC012A/controlfile/o1\_mf\_86sbg5qb1\_.ctl','/n01/oradata1/poc012/POC012A/controlfile/o1\_mf\_86sbg5qb\_.ctl'

\*.db\_block\_size=8192

\*.db\_create\_file\_dest='/s01/oradata1/poc012'

\*.db\_create\_online\_log\_dest\_1='/n01/oradata1/poc012'

\*.db\_domain='int.westgroup.com'

\*.db\_name='poc012a'

\*.diagnostic\_dest='/u01/app/oracle'

\*.dispatchers='(PROTOCOL=TCP) (SERVICE=poc012aXDB)'

~~poc012a2.instance\_number=2~~

~~poc012a1.instance\_number=1~~

\*.log\_archive\_dest\_1='LOCATION=/n01/oraarch1/poc012a'

\*.log\_archive\_format='%t\_%s\_%r.dbf'

\*.open\_cursors=300

~~\*.pga\_aggregate\_target=10142875648~~

**\*.pga\_aggregate\_target=500M**

\*.processes=150

~~\*.remote\_listener='labv2-scan-zl1:1521'~~

\*.remote\_login\_passwordfile='exclusive'

~~\*.sga\_target=30429675520~~

\*.sga\_target=2G

~~poc012a2.thread=2~~

~~poc012a1.thread=1~~

~~poc012a2.undo\_tablespace='UNDOTBS2'~~

~~poc012a1.undo\_tablespace='UNDOTBS1'~~

undo\_tablespace='UNDOTBS1'

This init file can be stored in any standard location on the Storage VM, such as $ORACLE\_HOME/dbs.

The backup controlfile gotten from the backup location can be copied over right into the expected locations on the Storage VM (per the control\_files parameter of the init file).

scp cntrl\_poc012a1.bkp xtrm-lionvm-01:/s01/oradata1/poc012/POC012A/controlfile/o1\_mf\_86sbg5qb1\_.ctl

scp cntrl\_poc012a1.bkp xtrm-lionvm-01:/n01/oradata1/poc012/POC012A/controlfile/o1\_mf\_86sbg5qb\_.ctl

Ensure the Thomson Reuters’ standard OFA directories and soft links are appropriately created under /01/app/oracle/diag and admin directories for the proper functioning of the new database.

Startup the database nomount on the Storage VM:

sqlplus /'as sysdba'

SQL\*Plus: Release 11.2.0.3.0 Production on Thu Oct 25 16:05:59 2012

Copyright (c) 1982, 2011, Oracle. All rights reserved.

Connected to an idle instance.

SQL> startup nomount pfile='/u01/app/oracle/product/11.2.0.3/db/dbs/initpoc012a.ora'

ORACLE instance started.

Total System Global Area 2137886720 bytes

Fixed Size 2230072 bytes

Variable Size 503318728 bytes

Database Buffers 1627389952 bytes

Redo Buffers 4947968 bytes

Mount the database using the backup controlfiles:

SQL> alter database mount;

Database altered.

Recover the database to a known point in time. This will be the most recent time that the data of interest (the data that will be subsequently extracted from the Restore VM) was known to be present in the primary database:

SQL> recover database using backup controlfile until cancel;

ORA-00279: change 3761933 generated at 10/24/2012 22:03:21 needed for thread 1

ORA-00289: suggestion : /n01/oraarch1/poc012a/1\_33\_795715832.dbf

ORA-00280: change 3761933 for thread 1 is in sequence #33

Specify log: {<RET>=suggested | filename | AUTO | CANCEL}

ORA-00279: change 3761933 generated at 10/24/2012 22:03:04 needed for thread 2

ORA-00289: suggestion : /n01/oraarch1/poc012a/2\_34\_795715832.dbf

ORA-00280: change 3761933 for thread 2 is in sequence #34

Specify log: {<RET>=suggested | filename | AUTO | CANCEL}

ORA-00279: change 3762785 generated at 10/24/2012 22:03:25 needed for thread 2

ORA-00289: suggestion : /n01/oraarch1/poc012a/2\_35\_795715832.dbf

ORA-00280: change 3762785 for thread 2 is in sequence #35

ORA-00278: log file '/n01/oraarch1/poc012a/2\_34\_795715832.dbf' no longer needed

for this recovery

Specify log: {<RET>=suggested | filename | AUTO | CANCEL}

ORA-00279: change 3762788 generated at 10/24/2012 22:03:26 needed for thread 1

ORA-00289: suggestion : /n01/oraarch1/poc012a/1\_34\_795715832.dbf

ORA-00280: change 3762788 for thread 1 is in sequence #34

ORA-00278: log file '/n01/oraarch1/poc012a/1\_33\_795715832.dbf' no longer needed

for this recovery

Specify log: {<RET>=suggested | filename | AUTO | CANCEL}

ORA-00279: change 3762822 generated at 10/24/2012 22:03:45 needed for thread 1

ORA-00289: suggestion : /n01/oraarch1/poc012a/1\_35\_795715832.dbf

ORA-00280: change 3762822 for thread 1 is in sequence #35

ORA-00278: log file '/n01/oraarch1/poc012a/1\_34\_795715832.dbf' no longer needed

for this recovery

Specify log: {<RET>=suggested | filename | AUTO | CANCEL}

CANCEL

Media recovery cancelled.

SQL> alter database open resetlogs;

Database altered.

Ensure the database is open for read/write:

SQL> select open\_mode from v$database;

OPEN\_MODE

--------------------

READ WRITE

Step

Extract the data of interest from the recovered database via datapump export or another appropriate method and deliver it to the primary via an import or another appropriate method.

Considerations and Recommendations

It is recommended that the process of flex-clone –based recovery be practiced regularly for all databases that may need to resort to using this method for partial database point-in-time recoveries. Practicing this technique would uncover any misconfiguration errors (wrong or missing vaulting schedules, ssh key misconfigurations, nfsclone authentication misconfigurations, etc…) and capacity under-provisioning errors (not having enough archived logs on HNAS to support the RPO, undersizing the auxiliary volumes on the Restore VM, etc…)

The initial intent is to use this method for databases having unique partial recovery point-in-time RPOs and generic Shared LION implementations. The technique may be expanded to other cases, as its use becomes more widespread.

The technique is meant to complement, not replace or diminish the relevance of, other recovery technologies, such as TSPITR, flashback table or flashback database, snapshot standby, etc... Testing the flex-clone –based partial database recovery technique for a specific database environment and understanding its capabilities and limitations (particularly limitations) for the environment in question, may allow to adjust the procedures for making the other recovery techniques more successful, such as resizing one’s UNDO or the flashback log storage area.