Contents

Exercise: Implementing Maximum Availability Architecture 3

Starting point 3

MAA using file system storage 4

On iron1, prepare the primary database for Data Guard 4

Prepare the primary instances for Data Guard 4

Prepare the standby instances 5

Add the database to the cluster registry 6

Backup the primary on iron1 7

Create the standby database on gold1 7

Configure SQL\*Net: 7

Start redo propagation and apply 8

Enable and test active data guard 8

Configure the Data Guard Broker on file system storage 10

Set the dg broker parameters 10

Configure SQL\*Net 10

Remove the existing redo transport settings 11

Create the configuration 11

Test switchover 11

Enable fast start failover 11

Test fast start failover and re-instatement 12

MAA using disk group storage 13

On iron1, prepare the primary database for Data Guard 13

Prepare the primary instances for Data Guard 13

Prepare the standby instances 14

On gold1 add the database to the cluster registry 16

Backup the primary on iron1 16

Create the standby database on gold1 16

Configure SQL\*Net: 17

Start redo propagation and apply 17

Enable and test active data guard 18

Configure the Data Guard Broker on disk group storage 19

Set the dg broker parameters 19

Configure SQL\*Net 19

Remove the existing redo transport settings 20

Create the configuration 20

Test switchover 20

Enable fast start failover 21

Test fast start failover and re-instatement 21

Client connections 21

Create primary and standby services 21

Configure SQL\*Net 22

Trigger to ensure services are correct 23

Test connect-time connection fault tolerance 24

Backup and recovery 25

Create a recovery catalog database 25

Create a recovery catalog 25

Register and configure the databases 26

Backup database and archive logs on the standby 26

Repair a damaged primary with a backup from the standby 27

# Exercise: Implementing Maximum Availability Architecture

14 September 2017

This exercise implements MAA with RAC -> RAC physical standby, using databases on ACFS clustered file system storage and on ASM disk groups.

## Starting point

The host server is configured with five VMs:

ironnet provides support services (DHCP, DNS, NFS) needed by the clusters

iron1 and iron2 are clustered into the iron cluster

gold1 and gold2 are clustered into the gold cluster

GI 12.1.0.2 is installed on all four cluster nodes

Flex ASM is configured with standard (not flex) clusters

GNS is used but without a delegated subdomain

all VIPs are statically registered in the DNS (which is the most common usage)

the SCANs are gold-cluster-scan and iron-cluster-scan

Each cluster has five disk groups:

cwfiles store the clusterware files and management database

datadg and fradg are used for the orcldg database

datafs and frafs are used for ACFS volumes for the orclfs database

Two databases are configured on the iron cluster:

orcldg uses ASM diskgroups for storage (simulating older environments)

orclfs uses ACFS file system storage (as on newer ODAs)

the databases are two node RACs

the gold cluster has the storage and Oracle Homes configured, but no databases

All passwords (Linux and Oracle) are set to oracle, both GI and DB homes are owner by oracle

## MAA using file system storage

### On iron1, prepare the primary database for Data Guard

. oraenv orclfs

srvctl stop database -db orclfs

srvctl start instance -db orclfs -instance orclfs1 -startoption mount

sqlplus sys/oracle@iron1:1521/orclfs as sysdba

alter database archivelog;

alter database flashback on;

alter database force logging;

alter database add standby logfile thread 1 size 50m;

alter database add standby logfile thread 1 size 50m;

alter database add standby logfile thread 2 size 50m;

alter database add standby logfile thread 2 size 50m;

select log\_mode,flashback\_on,force\_logging,name,db\_unique\_name,database\_role,open\_mode from gv$database;

select \* from v$log;

select \* from v$standby\_log;

select \* from v$logfile;

### Prepare the primary instances for Data Guard

sqlplus sys/oracle@iron1:1521/orclfs as sysdba

#alter system set db\_unique\_name=orclfs scope=spfile;

alter system set db\_file\_name\_convert='goldfs','orclfs','GOLDFS','ORCLFS' scope=spfile;

alter system set log\_file\_name\_convert='goldfs','orclfs','goldfs','ORCLFS' scope=spfile;

alter system set fal\_server=goldfs;

alter system set fal\_client=orclfs;

alter system set log\_archive\_config='dg\_config=(orclfs,goldfs)';

alter system set standby\_file\_management=auto;

alter system set log\_archive\_dest\_1='location=USE\_DB\_RECOVERY\_FILE\_DEST valid\_for=(all\_logfiles,all\_roles) db\_unique\_name=orclfs';

alter system set log\_archive\_dest\_2='service=goldfs lgwr async noaffirm valid\_for=(online\_logfiles,primary\_role) db\_unique\_name=goldfs';

alter system set log\_archive\_dest\_state\_2=defer;

srvctl stop database -db orclfs

srvctl start database -db orclfs

### Prepare the standby instances

Create a staging directory, as root on iron1:

mkdir /u01/app/backup

chown oracle:oinstall /u01/app /u01/app/backup

ssh gold1 mkdir /u01/app/backup

ssh gold1 chown oracle:oinstall /u01/\* /u01/app/backup

ssh gold2 chown oracle:oinstall /u01/\*

ssh iron2 chown oracle:oinstall /u01/\*

As oracle on iron1:

sqlplus sys/oracle@iron1:1521/orclfs as sysdba

create pfile='/u01/app/backup/initorclfs1.ora' from spfile;

scp /u01/app/backup/initorclfs1.ora gold1:/u01/app/backup/initorclfs1.ora

ssh gold1 mkdir /u01/data/orclfs

scp /u01/data/orclfs/orapworclfs gold1:/u01/data/orclfs/orapworclfs

ssh gold1 mkdir -p /u01/app/oracle/admin/orclfs/adump

ssh gold2 mkdir -p /u01/app/oracle/admin/orclfs/adump

ssh gold1 chown oracle:oinstall /u01/data/orclfs

ssh gold1 chown oracle:oinstall /u01/app/oracle/admin/orclfs/adump

ssh gold2 chown oracle:oinstall /u01/app/oracle/admin/orclfs/adump

On gold1, edit parameters in /u01/app/backup/initorclfs1.ora as follows:

\*.db\_unique\_name='goldfs'

\*.remote\_listener='gold-cluster-scan:1521'

\*.fal\_client='GOLDFS'

\*.fal\_server='ORCLFS'

db\_file\_name\_convert='orclfs','goldfs','ORCLFS','GOLDFS'

log\_file\_name\_convert='orclfs','goldfs','ORCLFS','GOLDFS'

\*.log\_archive\_dest\_1='location=USE\_DB\_RECOVERY\_FILE\_DEST valid\_for=(all\_logfiles,all\_roles) db\_unique\_name=goldfs'

\*.log\_archive\_dest\_2='service=orclfs lgwr async noaffirm valid\_for=(online\_logfiles,primary\_role) db\_unique\_name=orclfs'

\*.log\_archive\_dest\_state\_2='ENABLE'

Create the spfile:

export ORACLE\_HOME=/u01/app/oracle/product/12.1.0/dbhome\_1

export PATH=$ORACLE\_HOME/bin:$PATH

export ORACLE\_SID=orclfs1

sqlplus / as sysdba

create spfile='/u01/data/orclfs/spfileorclfs.ora' from pfile='/u01/app/backup/initorclfs1.ora';

### Add the database to the cluster registry

On gold1,

. oraenv +ASM1

srvctl add database -db goldfs -oraclehome /u01/app/oracle/product/12.1.0/dbhome\_1 -dbtype rac -role physical\_standby -dbname orclfs -spfile /u01/data/orclfs/spfileorclfs.ora -pwfile /u01/data/orclfs/orapworclfs

srvctl add instance -db goldfs -instance orclfs1 -node gold1

srvctl add instance -db goldfs -instance orclfs2 -node gold2

### Backup the primary on iron1

rman target sys/oracle@iron1:1521/orclfs

backup as compressed backupset database format '/u01/app/backup/db\_%U.bset';

backup as compressed backupset archivelog all format '/u01/app/backup/arch\_%U.bset';

scp /u01/app/backup/\*.bset gold1:/u01/app/backup

### Create the standby database on gold1

export ORACLE\_HOME=/u01/app/oracle/product/12.1.0/dbhome\_1

export PATH=$ORACLE\_HOME/bin:$PATH

export ORACLE\_SID=orclfs1

srvctl start instance -db goldfs -instance orclfs1 -startoption nomount

rman target sys/oracle@iron1:1521/orclfs auxiliary /

duplicate target database for standby dorecover backup location '/u01/app/backup';

(warning regarding no backup of recent archives is to be expected)

srvctl start instance -db goldfs -instance orclfs2 -startoption mount

sqlplus sys/oracle@gold-cluster-scan:1521/goldfs as sysdba

select host\_name,instance\_name from gv$instance;

select log\_mode,force\_logging,name,db\_unique\_name,database\_role,open\_mode from gv$database;

### Configure SQL\*Net:

Add these entries to /u01/app/oracle/product/12.1.0/dbhome\_1/network/admin/tnsnames.ora on all four nodes:

orclfs=(description=(address=(protocol=tcp)(host=iron-cluster-scan)(port=1521))(connect\_data=(service\_name=orclfs)))

orclfs1=(description=(address=(protocol=tcp)(host=iron1)(port=1521))(connect\_data=(sid=orclfs1)))

orclfs2=(description=(address=(protocol=tcp)(host=iron2)(port=1521))(connect\_data=(sid=orclfs2)))

goldfs=(description=(address=(protocol=tcp)(host=gold-cluster-scan)(port=1521))(connect\_data=(service\_name=goldfs)))

goldfs1=(description=(address=(protocol=tcp)(host=gold1)(port=1521))(connect\_data=(sid=orclfs1)))

goldfs2=(description=(address=(protocol=tcp)(host=gold2)(port=1521))(connect\_data=(sid=orclfs2)))

### Start redo propagation and apply

On gold1,

. oraenv goldfs

sqlplus sys/oracle@goldfs as sysdba

alter database recover managed standby database using current logfile disconnect from session;

select THREAD#, GROUP# , SEQUENCE#, STATUS,BYTES from v$standby\_log order by 1,2;

select inst\_id,process,status,sequence#,block# from gv$managed\_standby;

On iron1,

. oraenv orclfs

sqlplus sys/oracle@orclfs as sysdba

alter system set log\_archive\_dest\_state\_2=enable sid='\*';

select inst\_id,process,status,sequence#,block# from gv$managed\_standby;

select host\_name,instance\_name from gv$instance;

select log\_mode,force\_logging,name,db\_unique\_name,database\_role,open\_mode from gv$database;

### Enable and test active data guard

On gold1:

srvctl stop database -db goldfs

srvctl start database -db goldfs

sqlplus sys/oracle@goldfs as sysdba

alter database recover managed standby database using current logfile disconnect from session;

select host\_name,instance\_name from gv$instance;

select log\_mode,force\_logging,name,db\_unique\_name,database\_role,open\_mode from gv$database;

sqlplus system/oracle@orclfs

create table t1 as select \* from all\_users;

sqlplus system/oracle@goldfs

select \* from t1;

## Configure the Data Guard Broker on file system storage

### Set the dg broker parameters

sqlplus sys/oracle@orclfs as sysdba

alter system set dg\_broker\_config\_file1 = '/u01/data/orclfs/dr1orclfs.dat' scope=both sid='\*';

alter system set dg\_broker\_config\_file2 = '/u01/data/orclfs/dr2orclfs.dat' scope=both sid='\*';

alter system set dg\_broker\_start = true scope=both sid='\*';

sqlplus sys/oracle@goldfs as sysdba

alter system set dg\_broker\_config\_file1 = '/u01/data/orclfs/dr1orclfs.dat' scope=both sid='\*';

alter system set dg\_broker\_config\_file2 = '/u01/data/orclfs/dr2orclfs.dat' scope=both sid='\*';

alter system set dg\_broker\_start = true scope=both sid='\*';

### Configure SQL\*Net

Add dgmgrl static registrations to the four /u01/app/12.1.0/grid/network/admin/listener.ora files:

iron1,

sid\_list\_listener=(sid\_list=(sid\_desc=(global\_dbname=orclfs\_dgmgrl)(oracle\_home=/u01/app/oracle/product/12.1.0/dbhome\_1)(sid\_name=orclfs1)))

iron2,

sid\_list\_listener=(sid\_list=(sid\_desc=(global\_dbname=orclfs\_dgmgrl)(oracle\_home=/u01/app/oracle/product/12.1.0/dbhome\_1)(sid\_name=orclfs2)))

gold1,

sid\_list\_listener=(sid\_list=(sid\_desc=(global\_dbname=goldfs\_dgmgrl)(oracle\_home=/u01/app/oracle/product/12.1.0/dbhome\_1)(sid\_name=orclfs1)))

gold2,

sid\_list\_listener=(sid\_list=(sid\_desc=(global\_dbname=goldfs\_dgmgrl)(oracle\_home=/u01/app/oracle/product/12.1.0/dbhome\_1)(sid\_name=orclfs2)))

Re-start the node listeners on both clusters:

srvctl stop listener

srvctl start listener

### Remove the existing redo transport settings

sqlplus sys/oracle@orclfs as sysdba

alter system set log\_archive\_dest\_2='' sid='\*';

sqlplus sys/oracle@goldfs as sysdba

alter system set log\_archive\_dest\_2='' sid='\*';

Create the configuration

On iron1,

dgmgrl sys/oracle@orclfs

create configuration conf as primary database is orclfs connect identifier is orclfs;

add database goldfs as connect identifier is goldfs;

show configuration verbose

show database verbose orclfs

show database verbose goldfs

show instance verbose orclfs1 on database orclfs

show instance verbose orclfs2 on database orclfs

show instance verbose orclfs1 on database goldfs

show instance verbose orclfs2 on database goldfs

enable configuration;

show configuration verbose

### Test switchover

switchover to goldfs

show configuration verbose

Enable fast start failover

Start flashback logging at both databases:

select flashback\_on from v$database;

alter database flashback on;

dgmgrl sys/oracle@goldfs

edit database orclfs set property faststartfailovertarget=goldfs;

edit database goldfs set property faststartfailovertarget=orclfs;

show database orclfs faststartfailovertarget

show database goldfs faststartfailovertarget

enable fast\_start failover

show configuration verbose

start observer

Or, to start the observer in a production mode,

nohup dgmgrl sys/oracle@orclfs "start observer" &

### Test fast start failover and re-instatement

On gold1,

srvctl stop database -db goldfs -stopoption abort

srvctl start database -db goldfs -startoption mount

## MAA using disk group storage

### On iron1, prepare the primary database for Data Guard

. oraenv orcldg

srvctl stop database -db orcldg

srvctl start instance -db orcldg -instance orcldg1 -startoption mount

sqlplus sys/oracle@iron1:1521/orcldg as sysdba

alter database archivelog;

alter database flashback on;

alter database force logging;

alter database add standby logfile thread 1 size 50m;

alter database add standby logfile thread 1 size 50m;

alter database add standby logfile thread 2 size 50m;

alter database add standby logfile thread 2 size 50m;

select log\_mode,flashback\_on,force\_logging,name,db\_unique\_name,database\_role,open\_mode from gv$database;

select \* from v$log;

select \* from v$standby\_log;

select \* from v$logfile;

### Prepare the primary instances for Data Guard

sqlplus sys/oracle@iron1:1521/orcldg as sysdba

#alter system set db\_unique\_name=orcldg scope=spfile;

alter system set db\_file\_name\_convert='golddg','orcldg','GOLDDG','ORCLDG' scope=spfile;

alter system set log\_file\_name\_convert='golddg','orcldg','golddg','ORCLDG' scope=spfile;

alter system set fal\_server=golddg;

alter system set fal\_client=orcldg;

alter system set log\_archive\_config='dg\_config=(orcldg,golddg)';

alter system set standby\_file\_management=auto;

alter system set log\_archive\_dest\_1='location=USE\_DB\_RECOVERY\_FILE\_DEST valid\_for=(all\_logfiles,all\_roles) db\_unique\_name=orcldg';

alter system set log\_archive\_dest\_2='service=golddg lgwr async noaffirm valid\_for=(online\_logfiles,primary\_role) db\_unique\_name=golddg';

alter system set log\_archive\_dest\_state\_2=defer;

srvctl stop database -db orcldg

srvctl start database -db orcldg

### Prepare the standby instances

Create a staging directory, as root on iron1:

mkdir /u01/app/backup

chown oracle:oinstall /u01/app /u01/app/backup

ssh gold1 mkdir /u01/app/backup

ssh gold1 chown oracle:oinstall /u01/\* /u01/app/backup

ssh gold2 chown oracle:oinstall /u01/\*

ssh iron2 chown oracle:oinstall /u01/\*

sqlplus sys/oracle@iron1:1521/orcldg as sysdba

create pfile='/u01/app/backup/initorcldg1.ora' from spfile;

scp /u01/app/backup/initorcldg1.ora gold1:/u01/app/backup/initorcldg1.ora

. oraenv +ASM1

asmcmd pwget --dbuniquename orcldg

asmcmd pwcopy +DATADG/ORCLDG/PASSWORD/pwdorcldg.256.944258139 /u01/app/backup/orapworcldg

scp /u01/app/backup/orapworcldg gold1:/u01/app/backup/orapworcldg

ssh oracle@gold1

. oraenv +ASM1

asmcmd pwcopy /u01/app/backup/orapworcldg +datadg/orapworcldg

mkdir -p /u01/app/oracle/admin/orcldg/adump

ssh gold2 mkdir -p /u01/app/oracle/admin/orcldg/adump

chown oracle:oinstall /u01/app/oracle/admin/orcldg/adump

ssh gold2 chown oracle:oinstall /u01/app/oracle/admin/orcldg/adump

Edit parameters in /u01/app/backup/initorcldg1.ora as follows:

\*.db\_unique\_name='golddg'

\*.remote\_listener='gold-cluster-scan:1521'

\*.fal\_client='GOLDDG'

\*.fal\_server='ORCLDG'

\*.log\_archive\_dest\_1='location=USE\_DB\_RECOVERY\_FILE\_DEST valid\_for=(all\_logfiles,all\_roles) db\_unique\_name=golddg'

\*.log\_archive\_dest\_2='service=orcldg lgwr async noaffirm valid\_for=(online\_logfiles,primary\_role) db\_unique\_name=orcldg'

\*.db\_file\_name\_convert='orcldg','golddg','ORCLDG','GOLDDG'

\*.log\_file\_name\_convert='orcldg','golddg','ORCLDG','GOLDDG'

\*.log\_archive\_dest\_state\_2='ENABLE'

Create the spfile:

export ORACLE\_HOME=/u01/app/oracle/product/12.1.0/dbhome\_1

export PATH=$ORACLE\_HOME/bin:$PATH

export ORACLE\_SID=orcldg1

sqlplus / as sysdba

create spfile='+datadg/spfileorcldg.ora' from pfile='/u01/app/backup/initorcldg1.ora';

### On gold1 add the database to the cluster registry

srvctl add database -db golddg -oraclehome /u01/app/oracle/product/12.1.0/dbhome\_1 -dbtype rac -spfile +datadg/spfileorcldg.ora -pwfile +datadg/orapworcldg -role physical\_standby -dbname orcldg -diskgroup datadg,fradg

srvctl add instance -db golddg -instance orcldg1 -node gold1

srvctl add instance -db golddg -instance orcldg2 -node gold2

### Backup the primary on iron1

rman target sys/oracle@iron1:1521/orcldg

backup as compressed backupset database format '/u01/app/backup/db\_%U.bset';

backup as compressed backupset archivelog all format '/u01/app/backup/arch\_%U.bset';

scp /u01/app/backup/\*.bset gold1:/u01/app/backup

### Create the standby database on gold1

export ORACLE\_HOME=/u01/app/oracle/product/12.1.0/dbhome\_1

export PATH=$ORACLE\_HOME/bin:$PATH

export ORACLE\_SID=orcldg1

srvctl start instance -db golddg -instance orcldg1 -startoption nomount

rman target sys/oracle@iron1:1521/orcldg auxiliary /

duplicate target database for standby dorecover backup location '/u01/app/backup';

(warning regarding no backup of recent archives is to be expected)

srvctl start instance -db golddg -instance orcldg2 -startoption mount

sqlplus sys/oracle@gold-cluster-scan:1521/golddg as sysdba

select host\_name,instance\_name from gv$instance;

select log\_mode,force\_logging,name,db\_unique\_name,database\_role,open\_mode from gv$database;

### Configure SQL\*Net:

Add these entries to /u01/app/oracle/product/12.1.0/dbhome\_1/network/admin/tnsnames.ora on all four nodes:

orcldg=(description=(address=(protocol=tcp)(host=iron-cluster-scan)(port=1521))(connect\_data=(service\_name=orcldg)))

orcldg1=(description=(address=(protocol=tcp)(host=iron1)(port=1521))(connect\_data=(sid=orcldg1)))

orcldg2=(description=(address=(protocol=tcp)(host=iron2)(port=1521))(connect\_data=(sid=orcldg2)))

golddg=(description=(address=(protocol=tcp)(host=gold-cluster-scan)(port=1521))(connect\_data=(service\_name=golddg)))

golddg1=(description=(address=(protocol=tcp)(host=gold1)(port=1521))(connect\_data=(sid=orcldg1)))

golddg2=(description=(address=(protocol=tcp)(host=gold2)(port=1521))(connect\_data=(sid=orcldg2)))

### Start redo propagation and apply

On gold1,

. oraenv golddg

sqlplus sys/oracle@golddg as sysdba

alter database recover managed standby database using current logfile disconnect from session;

select THREAD#, GROUP# , SEQUENCE#, STATUS,BYTES from v$standby\_log order by 1,2;

select inst\_id,process,status,sequence#,block# from gv$managed\_standby;

sqlplus sys/oracle@orcldg as sysdba

alter system set log\_archive\_dest\_state\_2=enable sid='\*';

select inst\_id,process,status,sequence#,block# from gv$managed\_standby;

### Enable and test active data guard

On gold1:

srvctl stop database -db golddg

srvctl start database -db golddg

sqlplus sys/oracle@golddg as sysdba

alter database recover managed standby database using current logfile disconnect from session;

select host\_name,instance\_name from gv$instance;

select log\_mode,force\_logging,name,db\_unique\_name,database\_role,open\_mode from gv$database;

sqlplus system/oracle@orcldg

create table t1 as select \* from all\_users;

sqlplus system/oracle@golddg

select \* from t1;

## Configure the Data Guard Broker on disk group storage

### Set the dg broker parameters

sqlplus sys/oracle@orcldg as sysdba

alter system set dg\_broker\_config\_file1 = '+datadg/orcldg/dr1orcldg.dat' scope=both sid='\*';

alter system set dg\_broker\_config\_file2 = '+datadg/orcldg/dr2orcldg.dat' scope=both sid='\*';

alter system set dg\_broker\_start = true scope=both sid='\*';

sqlplus sys/oracle@golddg as sysdba

alter system set dg\_broker\_config\_file1 = '+datadg/golddg/dr1orcldg.dat' scope=both sid='\*';

alter system set dg\_broker\_config\_file2 = '+datadg/golddg/dr2orcldg.dat' scope=both sid='\*';

alter system set dg\_broker\_start = true scope=both sid='\*';

### Configure SQL\*Net

Add dgmgrl static registrations to the four /u01/app/12.1.0/grid/network/admin/listener.ora files:

iron1,

sid\_list\_listener=(sid\_list=(sid\_desc=(global\_dbname=orcldg\_dgmgrl)(oracle\_home=/u01/app/oracle/product/12.1.0/dbhome\_1)(sid\_name=orcldg1)))

iron2,

sid\_list\_listener=(sid\_list=(sid\_desc=(global\_dbname=orcldg\_dgmgrl)(oracle\_home=/u01/app/oracle/product/12.1.0/dbhome\_1)(sid\_name=orcldg2)))

gold1,

sid\_list\_listener=(sid\_list=(sid\_desc=(global\_dbname=golddg\_dgmgrl)(oracle\_home=/u01/app/oracle/product/12.1.0/dbhome\_1)(sid\_name=orcldg1)))

gold2,

sid\_list\_listener=(sid\_list=(sid\_desc=(global\_dbname=golddg\_dgmgrl)(oracle\_home=/u01/app/oracle/product/12.1.0/dbhome\_1)(sid\_name=orcldg2)))

Re-start the node listeners on both clusters:

srvctl stop listener

srvctl start listener

### Remove the existing redo transport settings

sqlplus sys/oracle@orcldg as sysdba

alter system set log\_archive\_dest\_2='' sid='\*';

sqlplus sys/oracle@golddg as sysdba

alter system set log\_archive\_dest\_2='' sid='\*';

### Create the configuration

On iron1,

dgmgrl sys/oracle@orcldg

create configuration conf as primary database is orcldg connect identifier is orcldg;

add database golddg as connect identifier is golddg;

show configuration verbose

show database verbose orcldg

show database verbose golddg

show instance verbose orclfs1 on database orcldg

show instance verbose orclfs2 on database orcldg

show instance verbose orclfs1 on database golddg

show instance verbose orclfs2 on database golddg

enable configuration;

show configuration verbose

### Test switchover

switchover to golddg

show configuration verbose

### Enable fast start failover

Start flashback logging at both databases:

select flashback\_on from v$database;

alter database flashback on;

dgmgrl sys/oracle@golddg

edit database orcldg set property faststartfailovertarget=golddg;

edit database golddg set property faststartfailovertarget=orcldg;

show database orcldg faststartfailovertarget

show database golddg faststartfailovertarget

enable fast\_start failover

show configuration verbose

start observer

### Test fast start failover and re-instatement

On gold1,

srvctl stop database -db golddg -stopoption abort

srvctl start database -db golddg -startoption mount

## Client connections

Note that the examples below are for orclfs/goldfs, substitute orcldg/golddg if necessary.

### Create primary and standby services

On iron1, create and start a primary service and a real time query standby service:

srvctl add service -db orclfs -service prim -role primary -preferred orclfs1,orclfs2

srvctl add service -db orclfs -service rtq -role physical\_standby -preferred orclfs1,orclfs2

srvctl start service -db orclfs -service prim

srvctl status service -db orclfs -service prim

srvctl start service -db orclfs -service rtq

srvctl stop service -db orclfs -service rtq

Create the services on gold1:

srvctl add service -db goldfs -service prim -role primary -preferred orclfs1,orclfs2

srvctl add service -db goldfs -service rtq -role physical\_standby -preferred orclfs1,orclfs2

srvctl start service -db orclfs -service rtq

srvctl status service -db orclfs -service rtq

### Configure SQL\*Net

And entries to the /u01/app/oracle/product/12.1.0/dbhome\_1/network/admin/tnsnames.ora file on iron1 for the primary service and the real time query service, enabling connect-time failover:

prim = (description =

(address\_list =

(failover=on)(load\_balance=on)

(address = (protocol = tcp)(host=iron-cluster-scan)(port=1521))

(address = (protocol = tcp)(host=gold-cluster-scan)(port=1521))

)

(connect\_data=(service\_name=prim))

)

rtq = (description =

(address\_list =

(failover=on)(load\_balance=on)

(address = (protocol = tcp)(host=iron-cluster-scan)(port=1521))

(address = (protocol = tcp)(host=gold-cluster-scan)(port=1521))

)

(connect\_data=(service\_name=rtq))

)

### Trigger to ensure services are correct

If necessary, this trigger is an alternative (or addition) to registering services within the OCR. Crezting it in the primary will propagate it to the standby:

CREATE OR replace TRIGGER start\_service

AFTER db\_role\_change ON DATABASE

DECLARE

drole VARCHAR2(20);

omode VARCHAR2(20);

BEGIN

SELECT database\_role

INTO drole

FROM v$database;

SELECT open\_mode

INTO omode

FROM v$database;

IF drole = 'PRIMARY' THEN

dbms\_service.start\_service('PRIM');

dbms\_service.stop\_service('RTQ');

ELSIF drole = 'PHYSICAL STANDBY' THEN

dbms\_service.stop\_service('PRIM');

dbms\_service.start\_service('RTQ');

END IF;

END;

/

### Test connect-time connection fault tolerance

Test that the connections do go to the correct databases, and leave the sessions connected:

sqlplus system/oracle@prim

sqlplus system/oracle@rtq

Initiate a fast start failover and reinstatement. Return to the SQL\*Plus sessions. Any SQL will return an error. Reconnect and confirm that the sessions do go to the appropriate database following the failover.

## Backup and recovery

Note that the examples below are for orclfs/goldfs, substitute orcldg/golddg if necessary. It is also assumed that orclfs is the primary.

### Create a recovery catalog database

On iron1, create a database to be used for the recovery catalog:

export ORACLE\_BASE=/u01/app/oracle

export ORACLE\_HOME=$ORACLE\_BASE/product/12.1.0/dbhome\_1

export PATH=$ORACLE\_HOME/bin:$PATH:.

export ORACLE\_SID=rman

Launch the dbca, and in the GUI:

Select Advanced Mode, and then choose to create a database from the general purpose template. Accept defaults except for:

Set the global database name 'rman'

De-select enterprise manager

Set the password for all accounts to 'oracle'

De-select ‘create as a container database’

Select ‘use common location for all database files’

Select 'use oracle managed files', in ORACLE\_BASE/oradata

Set the memory limit to 800MB, disable automatic memory management

Add this entry to the tnsnames.ora file:

rman=(description=(address=(protocol=tcp)(host=iron1)(port=1521))(connect\_data=(sid=rman)))

### Create a recovery catalog

sqlplus sys/oracle@rman as sysdba

create user rman identified by rman default tablespace sysaux quota unlimited on sysaux;

grant recovery\_catalog\_owner to rman;

exit;

rman catalog rman/rman@rman

create catalog;

exit;

### Register and configure the databases

rman target sys/oracle@orclfs catalog rman/rman@rman

register database;

list db\_unique\_name of database;

show all for db\_unique\_name orclfs;

configure backup optimization on;

configure controlfile autobackup on;

configure archivelog deletion policy to applied on all standby;

configure db\_unique\_name goldfs connect identifier 'goldfs';

configure db\_unique\_name orclfs connect identifier 'orclfs';

configure snapshot controlfile name to '/u01/data/orclfs/snapshot\_cf';

list db\_unique\_name of database;

report schema for db\_unique\_name orclfs;

report schema for db\_unique\_name goldfs;

exit;

### Backup database and archive logs on the standby

rman target sys/oracle@goldfs catalog rman/rman@rman

resync catalog from db\_unique\_name all;

backup as compressed backupset database;

list archivelog all for db\_unique\_name goldfs;

backup as compressed backupset archivelog all delete all input;

list archivelog all for db\_unique\_name goldfs;

list backup of tablespace users; --note the name of the backup piece!

exit;

### Repair a damaged primary with a backup from the standby

Simulate damage to a primary datafile:

sqlplus sys/oracle@orclfs as sysdba

alter tablespace users offline immediate;

select 'host rm '||file\_name from dba\_data\_files where tablespace\_name='USERS';

alter tablespace users online;

Copy the backup the standby to the primary:

scp gold1:/<path\_and\_file\_name\_of\_the\_backup\_piece> /u01/data

Restore and recover the file:

rman target sys/oracle@goldfs1 catalog rman/rman@rman

catalog backuppiece '/u01/data/< name\_of\_the\_backup\_piece>';

restore tablespace users;

recover tablespace users;

alter tablespace users online;