Use RMAN to relocate a 10TB RAC database with minimum downtime

Tao Zuo
tao_zuo@npd.com
NPD Inc.
9/2011
Methods of relocate a database with minimum down time
RMAN oracle suggested backup strategy
Case Study: Relocate a 10TB RAC database on ASM with RMAN
Methods of relocate a database with minimum down time

- Data Guard switchover
- Long-distance storage mirror
- RMAN oracle suggested backup strategy
Data Guard switchover

- Downtime dependency:
  - Last of the redo data transmission
  - Last of the redo data apply to standby database
- Force logging is the prerequisite
- Extra Network bandwidth for redo transmission
Long-distance storage mirror

- Storage volume level copy to remote location
- Down time can be within minutes. Dependency:
  - I/O freeze, volumes synchronize & split on Source site
  - ASM on target site startup.
  - Database on target site recovery
- Require extra network bandwidth
- Issue encountered: SAN reboot during the volume level synchronization
RMAN oracle suggested backup strategy

- Is based on creating an image copy of a database
- This copy is rolled forward by means of incrementally updated backups
RMAN oracle suggested backup strategy

Daily Script:
run {
recover copy of database with tag 'DBcopy';
backup incremental level 1
for recover of copy with tag 'DBcopy'
database;
}

RMAN oracle suggested backup strategy

- The key: incrementally updated backups
  - Fast with block change tracking enabled
  - Recovery time reduced with less redo to apply
  - Less bandwidth consumption on network transmission
  - Include blocks changed with nologging
RMAN oracle suggested backup strategy
Case Study: Relocate a 10TB RAC database on ASM with RMAN

- Specification of the database to be relocated:
  - RAC database runs on 4 nodes
  - Size: 10TB
  - 2 supported ASM disk groups: DATA, FLASH
  - Support no logging applications
  - Maximum archived log generated: 200G/hr
Case Study: Relocate a 10TB RAC database on ASM with RMAN

- Create ASM disk groups in the remote site
- Create a image copy on the remote site
- Roll forward the image copy on the remote site
- switch database to copy on remote site
- Startup the database as RAC database on the remote site
- Downtime Dependency:
  - Last incrementally updated backup & roll forward
  - Last of the redo data transmission
  - Last of the redo data apply to the database on the remote site
Case Study: Create ASM disk groups in the remote site

- **Init.ora:** `asm_diskstring='/dev/mapper/ora*'
- **CREATE DISKGROUP DATA External REDUNDANCY DISK**
  - `'/dev/mapper/oradisk01' size 512000M,
  - ...
  - `'/dev/mapper/oradisk22' size 512000M;
- **CREATE DISKGROUP FLASH External REDUNDANCY DISK**
  - `'/dev/mapper/oradisk24' size 512000M,
  - ...
  - `'/dev/mapper/oradisk44' size 512000M;`
Case Study: Create a image copy on the remote site

- Create a tape backup for the local database image copy
- Restore the database to the remote site from the tape backup in the FLASH ASM disk group
- Create image copy on the remote site in the DATA ASM disk group
Case Study: Create a image copy on the remote site

- Create a tape backup for the local database image copy

CONFIGURE CONTROLFILE AUTOBACKUP ON;
CONFIGURE DEFAULT DEVICE TYPE TO sbt;
CONFIGURE DEVICE TYPE 'SBT_TYPE' PARALLELISM 3 BACKUP TYPE TO BACKUPSET;
Case Study: Create a image copy on the remote site

- Create a tape backup for the local database image copy

```bash
run {
sql 'alter system archive log current';
backup
recovery area
tag 'Local_DB'
tag 'Local_DB_FRA_%s:%t:%p.bk'
force;
```
Case Study: Create a image copy on the remote site

- Restore the database to the remote site
  - Restore control file
  - Startup the database in mount state
  - Disable BLOCK CHANGE TRACKING
  - Restore database
  - Register the database to OCR as RAC database
Case Study: Create a image copy on the remote site

- Restore database to FLASH disk group
  - `.db_create_file_dest='+FLASH'`
Case Study: Create a image copy on the remote site

- Restore database to FLASH disk group
  - RMAN> startup nomount;
  - RMAN> restore controlfile from ...
  - RMAN> alter database mount;
  - RMAN> restore database;
Case Study: Create image copy on the remote site

- Create image copy in DATA ASM group
  - \*db_recovery_file_dest='+DATA'
Run {
    backup as copy
    incremental level 0
    tag ‘Remote_DB’
    Database;
}
Case Study: Roll forward the image copy on the remote site

- Create incrementally updated backups for local database
- Compress the backupsets & transmit to remote site
- Uncompress the backup sets on the remote site
- Roll forward the image copy on the remote site
Case Study: Roll forward the image copy on the remote site

- Create incrementally updated backups

CONFIGURE DEVICE TYPE DISK PARALLELISM 8 BACKUP TYPE TO BACKUPSET;
CONFIGURE DEFAULT DEVICE TYPE TO disk;
Case Study: Roll forward the image copy on the remote site

- Create incrementally updated backups

```sql
run {
  sql 'alter system archive log current';
  backup incremental level 1
database
tag 'Incr_DB'
format '/DBbackup/backupset/bs_%T_%U'
;}
```
Case Study: Roll forward the image copy on the remote site

- RMAN> catalog start with '/DBbackup/backups/bs' noprompt;
- RMAN> recover copy of database with tag 'Remote_DB';
Case Study: Roll forward the image copy on the remote site

- Schedule the remote copy roll forward
- Last roll forward occurred at switch over
Case Study: Switch database to copy

- Prepare switch over in the local site:
  - Backup controlfile
  - Backup last set of archivelogs

- Switch over in the remote site:
  - Update the controlfile(s)
  - Switch database to copy
  - Recover & Open database
Case Study:
Switch database to copy

- Backup controlfile

  RMAN> backup current controlfile format '/DBbackup/control/crtl.bk';
Case Study: Switch database to copy

- Backup last set of archivelogs
  - Identify the last image copy’s checkpoint#
  - Identify the archivelogs need by SCN for switch over
  - RMAN> sql ‘alter system archive log current’;
  - RMAN> backup as copy archivelog scn between <startSCN> and <endSCN>;
Case Study: Switch database to copy

- Backup last set of archivelogs

```sql
select 'backup as copy archivelog scn between ||min(lchk#)|| and
  ||max(lchk#)||chr(10)||
' format "/DBbackup/archivelog/al_lst_%U";
from (select max(CHECKPOINT_CHANGE#) lchk#,
    max(CHECKPOINT_TIME) lchkt,
    lstbk,file#
from (select max(COMPLETION_TIME) OVER (PARTITION BY file#) lstbk, file#,
    CHECKPOINT_CHANGE#, CHECKPOINT_TIME
from V$BACKUP_DATAFILE
where file#<>0
)
group by lstbk,file#)
);```
Case Study: Switch database to copy

- Update the controlfile in remote site
  - Reserve the datafile copy names to be re-cataloged
  - Restore the controlfile from the last backup made from the local database
  - Re-catalog datafile copy reserved
Case Study:
Switch database to copy

- Reserve the datafile copy names to be re-cataloged

```
select 'catalog datafilecopy "'||name||"";'
from v$backup_copy_details
where name like '+DATA/%'
and file#<>0 -- controlfile copy excluded
order by file#;
```
Case Study: Switch database to copy

- Restore the controlfile from the last backup made from the local database

  - RMAN> shutdown immediate;
  - RMAN> startup nomount;
  - RMAN> restore controlfile from '/DBbackup/control/crtl.bk';
Case Study: Switch database to copy

- Re-catalog

RMAN> catalog datafilecopy ...
RMAN> catalog start with '/DBbackup/archivelog/' noprompt;
Case Study: Switch database to copy

- Switch database to copy

    RMAN> switch database to copy;
Case Study:
Switch database to copy

- Recover & Open database

```
RMAN> recover database until scn <endSCN>;
SQL> ALTER DATABASE DISABLE BLOCK CHANGE TRACKING;
SQL> alter database open resetlogs;
```
Case Study: Startup the database as RAC database

- Edit initDB.ora
  - with current controlfile
  - *.db_create_file_dest='+DATA'
  - *.db_recovery_file_dest='+FLASH'
- Create SPFILE=‘+DATA/DB/spfileDB.ora’ from pfile=‘initDB.ora’
- Restart the database
- SQL> ALTER DATABASE enable BLOCK CHANGE TRACKING;
Use RMAN oracle suggested backup strategy to relocate database:

- No extra charge on software
- Zero impact on the database before switch over
- No extra network cost
- Database ‘down’ time is minimum
Thanks You!

Tao Zuo

tao_zuo@npd.com

NPD Inc.

9/2011