Hardware migration using Data Guard

Jon Westgaard
NDGF / University of Oslo

November 2009

NDGF
Nordic DataGrid Facility

- The ATLAS conditions database at NDGF was migrated to new hardware in October

- We used the migration procedure with Data Guard from the CERN Twiki pages: https://twiki.cern.ch/twiki/bin/viewauth/PDBService/MigrationWithDataGuard32to64

- The database was migrated from:
  - Single instance
  - 32 bit, RHEL4
  - Located in Helsinki, Finland (CSC)
- migrated to:
  - 3 node RAC
  - 64 bit, RHEL5
  - Located in Oslo, Norway (University of Oslo)
- Size of database during migration: ~350GB

NDGF
Nordic DataGrid Facility

Step 1 – Backup the old DB

- Full backup of the old database to local disk (outside of ASM) including archivelog and current controlfile for standby
- We had to use RMAN compressed backupset due to lack of disk space
- RMAN compression algorithm in 10gR2 is really slow - CPU intensive
- Elapsed time for the backup: 1.5 hour

NDGF
Nordic DataGrid Facility

Step 2 – Copy backup files

- Copied the RMAN backup files from the old server to the new cluster
- This step was required since the old and new servers did not have any shared disk or shared RMAN tape device
- The path and filename of the backup files on the new server and the old server must be identical
- Elapsed time for copying the backup files from Helsinki to Oslo: 10 minutes

NDGF
Nordic DataGrid Facility

Step 3 – create stdby database

- Created standby database with
  DUPLICATE TARGET DATABASE FOR STANDBY DORECOVER;
- I did an initial duplicate with dorecover to get a consistent database that could be activated and opened for testing without starting redo apply
- Elapsed time for “duplicate … for standby dorecover”: 2.5 hours (using compressed backupsets from local disk)

Step 4 – start redo apply

- Started redo apply and prepared for switchover
- Read note 751600.1 *Data Guard Physical Standby Switchover*:
  - Pre-switchover checks: no apply delay, no large gaps
  - The note has some useful information about fallback options if switchover fails

Step 5 - switchover

- Turned on redo apply tracing:
  alter system set log_archive_trace=8191
  (optional, might be useful if something fails)
- Stopped streams propagation from Tier 0
- Switched the old database from primary to standby (and verified that the new database had received end-of-redo from the old database)
- Switched the new database from standby to primary

Step 6 – post switchover tasks

- Additional step due to switchover from 32 to 64 bit:
  - Recompile PL/SQL code (see note: 414043.1 *Role Transitions for Data Guard Configurations Using Mixed Oracle Binaries)*
- Switched to cluster database - added RAC node 2 and 3
- Restarted streams propagation from Tier 0 using the new connect string

Conclusion

- The migration procedure using Data Guard worked fine giving:
  - Minimal service downtime
  - Downtime independant of the size of the database

- One comment:

  If the spfile is created from pfile when the database is down (as in this procedure) it is created in the following directory:
  +*dg*/DB_UNKNOWN/PARAMETERFILE/
  and not in:
  +*dg*/<db_name>/PARAMETERFILE/

  We should possibly create the spfile again with database in mount so that it is moved to the correct directory