DATA GUARD ALS HA-LÖSUNG

Praktische Erfahrungen

Mila Friedman
Lufthansa Systems AG
Kelsterbach

Schlüsselworte

Physical Data Guard Configuration, Data Guard Broker, Observer, Active Data Guard , RMAN, Corruption

Einleitung

Vor vier Jahren haben wir, die Entscheidung getroffen, die RAC-Systeme - wenn möglich - auf Data Guard umzustellen. Es wurde ein extra Engineering Projekt aufgesetzt; anschließend wurden die ersten Data Guard-Systeme mit automatischem Failover aufgebaut. Zurzeit betreuen wir über 50 physikalische Data Guard-Konfigurationen auf unterschiedlichen Unix-Plattformen. Die meisten davon sind extrem kritisch. Einige Systeme haben mehrere Standby-Datenbanken und Active Data Guard. In diesem Vortrag diskutieren wir über unsere Erfahrungen und die Besonderheiten des Aufbaus und der Administration von physikalischen Data Guard -Systemen bei der Lufthansa Systems AG.

Die ersten Schritte

Bevor Data Guard aufgebaut wird, muss man die Kundenanforderungen und die technischen Möglichkeiten analysieren, um die Data Guard-Konfiguration (Protection Mode, Delay, Active Data Guard) richtig vorzubereiten.

Ich möchte hier nicht über den Standardaufbau der physikalischen Standby Datenbank diskutieren, der in vielen Metalink Notes und der Oracle-Dokumentation bereits beschrieben ist. In meinem Vortrag konzentriere ich mich auf die Parameter und die Schritte, die während des Aufbaus berücksichtigt werden sollten.

Nachdem die Standby Datenbank laut Dokumentation aufgebaut ist, würde ich unbedingt empfehlen, ein „Test“-Tablespace in der Primary Datenbank anzulegen. um das STANDBY_FILE_MANAGEMENT zu testen und um ein Switchover mit SQL Plus: Oracle 10g/11g (Metalink Note 232240.1) und Oracle 12C (Note 1578787.1) durchzuführen. Nur wenn die Tests erfolgreich sind, kann man mit der Data Guard Broker Konfiguration anfangen

.

DATA GUARD BROKER

Der Data Guard Broker ist ein Framework für die zentrale Administration und Monitoring der Data Guard Konfiguration, welcher ab 9i eingeführt wurde. Der Data Guard Broker bietet die vereinfachte Durchführung von Switchover und Failover. Die folgenden Schritte sind notwendig, um den Broker aufzusetzen:

Vor dem Aufsetzen des Data Guard Broker sind einige Vorbereitungen durchzuführen. Als erstes soll der Broker-Prozess **DMON** (Data Guard Monitor) auf der Primary und den Standby Datenbanken gestartet werden: `alter system set dg_broker_start=true`. Dieser Prozess verbindet den Data Guard Broker mit der beteiligten Datenbank. Die komplette Information zum Data Guard Broker kann man in drc<DB_NAME>.log unter diagnostic_dest finden. Als nächstes werden die Konfigurationsdateien festgelegt: `dg_broker_config_file1 und dg_broker_config_file2`. Es ist zu empfehlen, aus Sicherheitsgründen die Konfigurationsdateien auf zwei unterschiedlichen Lokationen zu konfigurieren. Als nächster Schritt muss einen Eintrag mit DB_UNIQUE_NAME _DGMGRL .domain in listener.ora auf dem Standby und dem Prod Server vorgenommen werden.

Jetzt kann man in DMGMRL (Data Guard Manager CLIs) eine Konfiguration anlegen:

```
CREATE CONFIGURATION <configuration name> AS
    PRIMARY DATABASE IS <database name>
    CONNECT IDENTIFIER IS <connect identifier>;
```

und standby Datenbank hinfügen

```
ADD DATABASE <database name> AS
    CONNECT IDENTIFIER IS <connect identifier>
    MAINTAINED AS {PHYSICAL};
```

Nun muss die Konfiguration aktiviert werden: **`enable configuration.`**
Mit **`show configuration verbose`** kann man prüfen, ob Konfiguration erfolgreich aktiviert wurde.

```
DGMGRL> show configuration verbose;
Configuration - t1201t
Protection Mode: MaxAvailability
Databases:
T1201T_4 - Primary database
T1201T_3 - Physical standby database
  Properties:
    FastStartFailoverThreshold      = '30'
    OperationTimeout                = '30'
    TraceLevel                      = 'USER'
    FastStartFailoverLagLimit       = '30'
    CommunicationTimeout            = '180'
    ObserverReconnect               = '0'
```

```
    FastStartFailoverAutoReinstate  = 'TRUE'
    FastStartFailoverPmyShutdown    = 'TRUE'
    BystandersFollowRoleChange      = 'ALL'
    ObserverOverride                = 'FALSE'
 Fast-Start Failover: DISABLED
 Configuration Status:
 SUCCESS
```

Die meisten Konfigurationseinstellungen vom Data Guard sollten nur mit DGMGRL angepasst werden. Dazu gehören auch die Änderung vom Protection Mode und das Aktivieren und Deaktivieren des Transportstatus. Die komplette Liste der Data Guard-Einstellungen kann man aus der Oracle Dokumentation (Oracle® Data Guard Broker) entnehmen.

Client Failover

Im Falle einer Störung auf der Primary Datenbank verliert die Applikation die Verbindung zur der Produktionsdatenbank. Damit nach dem Failover die Applikation eine neue Primary Datenbank finden kann, muss man in der Datenbank einen extra Service mit dbms_service konfigurieren und die Net Konfiguration anpassen.

Das ist ein Beispiel:

```
T121M_PROD.WORLD =
  (DESCRIPTION =
    (CONNECT_TIMEOUT=5)(TRANSPORT_CONNECT_TIMEOUT=3)(RETRY_COUNT=3)
   (ADDRESS_LIST=
    (ADDRESS = (PROTOCOL = TCP)(Host = <Horstname1>)(Port = <Listener_Port>))
    (ADDRESS = (PROTOCOL = TCP)(Host = <Horstname2>)(Port = <Listener_Port>))
   )
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = T121M_PROD.WORLD)
    )
  )
```

Die Applikation benutzt diesen Service, um die Verbindung auf die neue Primary Datenbank automatisch aufzubauen. Der Service wird über einen Startup Trigger auf der Primary Datenbank gestartet. Wenn die Applikation mehrere IP-Adressen nicht unterstützt, kann man eine Sonderlösung mit „geschwenkter“ VIP-Adresse benutzen, die in diesem Vortrag vorgestellt wird. Für Active Data Guard sollte auch ein extra Service angelegt werden, der nach dem Failover / Switchover auf der Standby Datenbank über den Startup Trigger auf der Standby Datenbank gestartet werden.

FAST START FAILOVER (FSFO)

Mit der Aktivierung des FSFO besteht die Möglichkeit, automatisch eine Standby-Datenbank zu einer Primärdatenbank zu aktivieren, wenn die aktuelle Primärdatenbank ausgefallen ist. Dafür wird ein Observer auf einem dritten Host gestartet, der bei Ausfall der Primärdatenbank durch den Parameter

FastStartFailoverTarget konfigurierte Standby-Datenbank aktiviert. Mit dem Parameter FastStartFailoverthreshold kann man ein Zeitintervall angeben, in dem der Observer wartet, bevor FSFO initialisiert wird. Dieser Parameter ist per Default auf 30 Sekunden eingestellt; ich würde empfehlen, diesen Parameter auf 40 Sekunden zu konfigurieren. Wenn Maximum Performance Protection Mode eingestellt ist, sollte der Parameter FastStartFailoverLagLimit konfiguriert werden (Default 30 sec, min 10 sec).

Für FSFO sollte der Protection Mode bei Oracle 10g auf Maximum Availability gesetzt werden, bei Oracle 11g ist auch Maximum Performance erlaubt. Ein Fast-Start Failover findet nur dann statt, wenn bestimmte Failover-Konditionen erfüllt sind: Das sind die per Default vordefinierten Failover-Konditionen (Database Offline, Corrupted Dictionary, Corrupted Controlfile). Neben den vordefinierten Konditionen können auch ORA-Meldungen das FSFO auslösen.

Wie ich bereits erwähnt habe, ist der Observer eine wichtige Komponente des FSFO. Um den Observer zu konfigurieren, muss man zuerst eine richtige Lokation für den Observer festlegen. Die ersten Observer für unsere Data Guard-Systeme wurden auf unserem Cloud Control Server konfiguriert. Leider haben wir festgestellt, dass auf einigen Systemen unerwartete und unerklärte Failovers auftraten. Um das FSFO-Risiko zu minimieren, haben wir die Entscheidung getroffen, die Observer auf die dazugehörigen Applikationsserver in separater Umgebung zu verschieben.

Auf dem Server, der die Observer Funktion übernehmen soll, muss ein Oracle Client Administrator installiert werden. Am besten ist es, wenn der Oracle Client und die Oracle Datenbank die gleiche Version haben. Die volle Kompatibilitätsmatrix zwischen dem Oracle Datenbank Server und dem Oracle Observer kann man aus Note 1625597.1 entnehmen. Erfolgt die Administration über das DGMGRL, reicht die Oracle Client Administrator Installation. Bei der der Nutzung von Grid / Cloud Control muss auch der Grid / Cloud Agent mit installiert werden. Unsere Test-Observer mit Grid Control zu administrieren war leider wegen diverser Bugs nicht erfolgreich.

Auch nach der Migration vom Grid Control auf Cloud Control haben wir die Entscheidung getroffen, den Observer über die Kommandozeile DGMGRL zu steuern. Dafür wird ein Unix / Linux-Script angelegt und der Observer als Background-Prozess gestartet. Ab Oracle 11.2 kann man den Observer direkt im Script als Background Prozess starten. Damit das Passwort im Script nicht im Klartext steht, würde ich empfehlen, einen Secure External Password Store konfigurieren (keine Lizenzierung notwendig). Natürlich ist auch Oracle Net auf dem Observer vollständig zu konfigurieren, damit der Observer Kontakt zu allen beteiligten Datenbanken aufnehmen kann.

Das ist ein Beispiel für Observer **start script** für Oracle 12C:

```
export ORACLE_BASE=/ora_u01/app/oraagt
export TNS_ADMIN=/ora_u01/app/oraagt/admin/network
export ORACLE_HOME=/ora_u01/app/oraagt/product/12.1.0.1/client64
export
PATH=$PATH:/bin:/usr/bin:/usr/sbin:/usr/local/bin:/usr/openwin/bin/:.
export BASE_PATH=$PATH
```

```
export PATH=$ORACLE_HOME/bin:$BASE_PATH
dgmgrl -logfile $ORACLE_BASE/Observer/log/ob<DB_NAME>.log
/@<db_connect_string> "start observer
file='/ora_u01/app/oraagt/product/12.1.0.1/client64/<DB_NAME>.dat'" &
```

Das ist ein Beispiel für Observer **stop script** fur Oracle 12C:

```
export ORACLE_BASE=/ora_u01/app/oraagt
export TNS_ADMIN=/ora_u01/app/oraagt/admin/network
export ORACLE_HOME=/ora_u01/app/oraagt/product/12.1.0.1/client64
export
PATH=$PATH:/bin:/usr/bin:/usr/sbin:/usr/local/bin:/usr/openwin/bin/:.
export BASE_PATH=$PATH
export PATH=$ORACLE_HOME/bin:$BASE_PATH
dgmgrl -logfile $ORACLE_BASE/Observer/log/ob<DB_NAME>.log << eof
connect  /@<db_connect_strin>
stop observer;
```

Es ist auch wichtig, start / stop Skripte in die Server runlevel zu integrieren, damit der Observer beim Reboot automatisch gestartet und gestoppt wird. Nach dem Starten vom Observer kann man im Data Guard Manager (dgmgrl) FSFO mit `enable fast_start failover` aktivieren. Anschließend muss man den Status der Konfiguration mit `show configuration verbose` prüfen. Bei `ERROR` oder `WARNING` sollte drc<DB_NAME> .log analysiert werden.

Jetzt, wenn die Data Guard Broker Konfiguration fertig ist, es ist wichtig, mehrere Switchover und Failover Tests incl. Standby Datenbank „Reinstate“ mit DGMGRL auszuführen, bevor die Datenbank produktiv geht. Es ist auch zu empfehlen, die Datenbank nur noch mit DGMRL zu stoppen und zu starten. Sollte ein Switchover mit SQL Plus ausgeführt werden, kann der Broker durcheinander kommen, sodass sogar der Broker anschließend komplett neu konfiguriert werden soll.

Backup Konzept

Unsere Datenbanken werden standardmäßig immer mit RMAN gesichert. Bei der Lufthansa Systems AG wird das RMAN Backup auf der Primary und den Standby Datenbanken gestartet. Anschließend wird in den Skripten DATABASE_ROLE in `v$database` geprüft. Nur wenn DATABASE_ROLE gleich PRIMARY ist, läuft das Backup weiter. An der RMAN Katalog Datenbank muss lediglich die Primary Datenbank registriert werden. Alle physikalischen Standby Datenbanken haben die gleiche DBID wie die Primary Datenbank und werden dadurch automatisch mit registriert. Die Zuordnung der Backups erfolgt über den DB_UNIQUE_NAME. Es muss noch die Archive Log Deletion Policy konfiguriert werden. Diese Policy steuert, wann RMAN Archive löschen darf.

Für Oracle 10G, wenn das Backup auf der Primary Datenbank läuft, auf der Standby Datenbank
`RMAN> CONFIGURE ARCHIVELOG DELETION POLICY TO APPLIED ON STANDBY;`
Ab Oracle 11g, wenn das Backup auf der Primary Datenbank läuft, auf der Standby Datenbank,:
`RMAN> CONFIGURE ARCHIVELOG DELETION POLICY TO APPLIED ON` **`ALL`** `STANDBY;`

Auf der Primary Datenbank:

```
RMAN> CONFIGURE ARCHIVELOG DELETION POLICY TO NONE;
```

Eine Beispielkonfiguration ist:

```
RMAN> show all for db_unique_name all;
RMAN configuration parameters for database with db_unique_name PTEST_1 are:
CONFIGURE RETENTION POLICY TO REDUNDANCY 1; # default
CONFIGURE BACKUP OPTIMIZATION OFF; # default
CONFIGURE DEFAULT DEVICE TYPE TO DISK; # default
CONFIGURE CONTROLFILE AUTOBACKUP OFF; # default
CONFIGURE CONTROLFILE AUTOBACKUP FORMAT FOR DEVICE TYPE DISK TO '%F'; # default
CONFIGURE DEVICE TYPE DISK PARALLELISM 1 BACKUP TYPE TO BACKUPSET; # default
CONFIGURE DATAFILE BACKUP COPIES FOR DEVICE TYPE DISK TO 1; # default
CONFIGURE ARCHIVELOG BACKUP COPIES FOR DEVICE TYPE DISK TO 1; # default
CONFIGURE MAXSETSIZE TO UNLIMITED; # default
CONFIGURE ENCRYPTION FOR DATABASE OFF; # default
CONFIGURE ENCRYPTION ALGORITHM 'AES128'; # default
CONFIGURE COMPRESSION ALGORITHM 'BASIC' AS OF RELEASE 'DEFAULT' OPTIMIZE FOR LOAD TRUE ; # default
CONFIGURE ARCHIVELOG DELETION POLICY TO NONE;
CONFIGURE SNAPSHOT CONTROLFILE NAME TO
'/ora_u01/app/orapstor/product/11.2.0.3/PSTORM/dbs/snapcf_PTEST_1.f';



RMAN configuration parameters for database with db_unique_name PTEST_2 are:
CONFIGURE RETENTION POLICY TO REDUNDANCY 1; # default
CONFIGURE BACKUP OPTIMIZATION OFF; # default
CONFIGURE DEFAULT DEVICE TYPE TO DISK; # default
CONFIGURE CONTROLFILE AUTOBACKUP OFF; # default
CONFIGURE CONTROLFILE AUTOBACKUP FORMAT FOR DEVICE TYPE DISK TO '%F'; # default
CONFIGURE DEVICE TYPE DISK PARALLELISM 1 BACKUP TYPE TO BACKUPSET; # default
CONFIGURE DATAFILE BACKUP COPIES FOR DEVICE TYPE DISK TO 1; # default
CONFIGURE ARCHIVELOG BACKUP COPIES FOR DEVICE TYPE DISK TO 1; # default
CONFIGURE MAXSETSIZE TO UNLIMITED; # default
CONFIGURE ENCRYPTION FOR DATABASE OFF; # default
CONFIGURE ENCRYPTION ALGORITHM 'AES128'; # default
CONFIGURE COMPRESSION ALGORITHM 'BASIC' AS OF RELEASE 'DEFAULT' OPTIMIZE FOR LOAD TRUE ; # default
CONFIGURE ARCHIVELOG DELETION POLICY TO APPLIED ON ALL STANDBY;
CONFIGURE SNAPSHOT CONTROLFILE NAME TO '/ora_u01/app/orapstor/product/11.2.0.3/PSTORM/dbs/snapcf_PTEST_2f';
```

Für das Disaster Recovery Konzept sollte das Backup auf den Primary und Standby Datenbanken aktiviert werden. In diesem Fall muss man dementsprechend auch zwei unterschiedliche Kataloge anlegen.

FAZIT

Durch den Wechsel von RAC zu Data Guard haben wir mehr Stabilität und eine Kostenersparnis erreicht. Basierend auf der Fast Start Failover Option können wir Data Guard als HA-Lösung erfolgreich einsetzen. Die Verwendungsmöglichkeiten von Data Guard sind vielfältig
Für die Oracle Migration von 11g nach 12c planen wir mehr Rolling Upgrades mittels der Umwandlung von physikalischer Standby in Transient logische Standby Datenbank durchzuführen.

Kontaktadresse:
Mila Friedman
Lufthansa Systems AG
Am Weiher 24
D-65451 Kelsterbach
Telefon: +49 (0) 69-696 6277
E-Mail mila.friedman@lhsystems.com
Internet: www.lufthansa-systems.com