Oracle Data Guard: Mit oder Ohne Broker?

Dierk Lenz
Herrmann & Lenz Services GmbH
Burscheid

Schlüsselworte
Data Guard, Data Guard Broker, Fast-Start Failover

Einleitung

Oracle Data Guard eine der zentralen Komponenten der Oracle Datenbank für Hochverfügbarkeit. Bei der Konfiguration von Data Guard hat man die Wahl, Data Guard manuell und explizit selbst zu administrieren oder den Data Guard Broker einzusetzen, der einige Abläufe automatisiert und Konfigurationsschritte vereinfacht. Zudem ist ein automatisiertes Failover (Fast-Start Failover) mit dem Observer möglich, wenn der Broker eingesetzt wird.
In diesem Vortrag werden Vor- und Nachteile der Varianten beleuchtet und praxisrelevante Aspekte diskutiert.

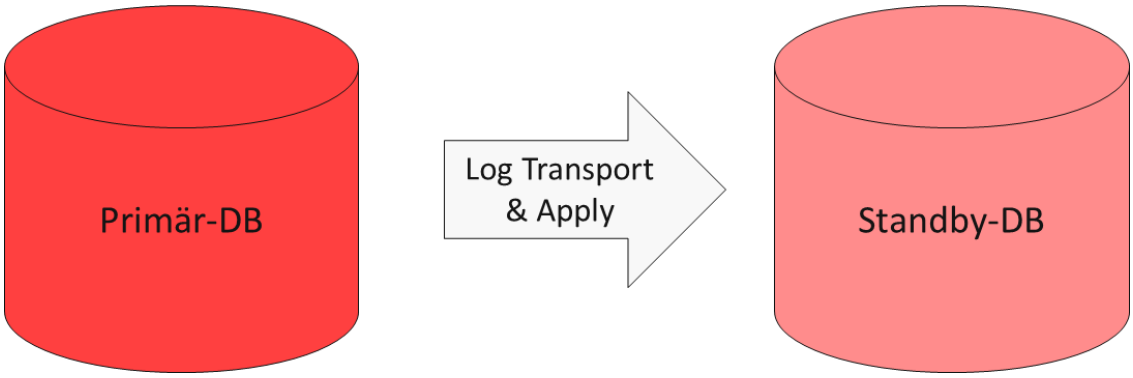
Data Guard Basics

Data Guard wird häufig in Kombination mit RAC (Real Application Clusters) eingesetzt. Dabei sichert RAC die Verfügbarkeit auf Hardware-Ebene durch Redundanz von Datenbank-Servern, Data Guard die Verfügbarkeit der Datenbank selbst. Dies geschieht dadurch, dass Änderungen des Primärsystems auf ein oder mehrere Standby-Systeme übertragen werden, so dass auch ein Verlust der Datenbankdateien des Primärsystems abgesichert wird. Primär- und Standby-Datenbank können dann z.B. kontrolliert einen Rollenwechsel durchführen (Switchover), um z.B. auf dem Primär-Server Wartungsarbeiten zu ermöglichen. Bei Verlust der Primärdatenbank kann die Standby-Datenbank durch einen Failover zur Primärdatenbank werden.

Für die Übertragung der Änderungen gibt es zwei Verfahren:

- Übertragung von SQLs (Logische Standby-Datenbank bzw. SQL Apply)
- Übertragung von Redo-Informationen (Physische Standby-Datenbank bzw. Redo Apply)

Dieser Vortrag beschäftigt sich mit der Variante Redo Apply. In vielen Projekten hat sich diese Variante als sehr robust erwiesen, Zusätzlich stellt sie keinerlei Bedingungen an Datenstrukturen (Datentypen, Primärschlüssel usw.).



Auch die Lizenzanforderungen sollen nicht unerwähnt bleiben: Data Guard ist eine Komponente, die ausschließlich in der Enterprise Edition zur Verfügung steht. Die Option Active Data Guard ist nur dann erforderlich, wenn man die zusätzlichen Funktion benötigt.

Konfiguration von Data Guard

Vor der Konfiguration einer Data Guard-Umgebung müssen die Anforderungen geklärt werden. Zentral ist die Frage, wie synchron die Standby-Datenbank zur Primärdatenbank sein soll. Ein Extremfall ist z.B., dass eine Transaktion erst dann in der Primärdatenbank commited werden darf, wenn diese gesichert bei der Standby-Datenbank „angekommen" ist. Dass dieser Modus u.a. Auswirkungen auf die Performance hat, liegt auf der Hand. Oft wird ein Modus gewählt, der auf die Bestätigung der Standby-Datenbank verzichtet, aber trotzdem so synchron wie möglich arbeitet. Hier wird meist ein Zeitversatz von wenigen Sekunden zur Primärdatenbank erreicht – allerdings ohne größere Performance-Auswirkungen. Beide Modi laufen als Real-Time Apply, da der LGWR-Prozess parallel zum Schreiben in die Redolog-Dateien den Log Transport durchführt. Auch ein Zeitversatz von z.B. einer Stunde lässt sich konfigurieren, so dass logische Fehlersituationen (z.B. TRUNCATE TABLE) erkannt und der Apply-Prozess gestoppt werden kann, bevor die Transaktion auf der Standby-Datenbank ankommt.

Es gibt einige Voraussetzungen, die zumindest für einen späteren Broker-Einsatz bzw. das Fast-Start Failover unabdingbar sind.

- Die Primärdatenbank muss auf FORCE LOGGING geschaltet werden. Tut man dies nicht, so sind NOLOGGING-Operationen in der Datenbank möglich. Das führt im Data Guard-Betrieb zu undefinierten Blöcken in der Standby-Datenbank – was im Zweifelsfall heißt, dass die Standby-Datenbank nicht einsetzbar ist.
- Die Standby-Datenbank muss für das Real-Time Apply über Standby Redolog-Dateien verfügen. Es ist empfehlenswert, diese im Vorfeld auf der Primärdatenbank anzulegen: Sie werden dann bei der initialen Datenbankkopie übernommen und fehlen nicht bei einem späteren Rollenwechsel.
- Primär- und Standby-Instanzen müssen über einen statischen SID_DESC-Eintrag beim Listener verfügen. Das hat den einfachen Grund, dass dynamische Einträge nicht zur Verfügung stehen, wenn die Datenbank nicht mindestens im MOUNT-Status ist – und beim Data Guard-Rollenwechsel werden Instanzen auf jeden Fall neu gestartet. Die statischen Einträge sind ebenfalls beim DUPLICATE mit dem Recovery Manager erforderlich.
- Für Fast-Start Failover: Aktivierung des Flashback-Modus. Voraussetzung hierfür ist wiederum eine Fast Recovery Area.

Das Aufsetzen einer Data Guard-Konfiguration beginnt dann mit der Kopie der Primärdatenbank. Wichtig ist, dass hierbei nicht eine übliche Kopie der Kontrolldatei verwendet wird, sondern eine Standby-Kontrolldatei. Gängige Praxis ist die Verwendung des Recovery Managers, der u.a. die Besonderheit der Standby-Kontrolldatei mit dem Kommando DUPLICATE DATABASE FOR STANDBY unterstützt.

Eine Handvoll Server-Parameter genügen weiterhin zur Definition der Data Guard-Umgebung:

- Da die Datenbanken physische Kopien sind und per Definition gleiche Datenbanknamen haben, muss die Unterscheidung über den Parameter DB_UNIQUE_NAME erfolgen. (Anmerkung: Die oft verwendeten Namensanhänge PRIM bzw. STDBY und deren Varianten haben den Nachteil, dass sie nach einem Rollenwechsel schlicht falsch ist. Hier sind A, B, C… die besseren Varianten.)

- Der Log Transport wird mit dem Parameter LOG_ARCHIVE_DEST_<n> konfiguriert. Hier werden neben dem TNS-Namen im SERVICE-Parameter wichtige Eigenschaften des Log Transports definiert.
- Weitere Parameter sind LOG_ARCHIVE_CONFIG, FAL_SERVER und STANDBY_FILE_MANAGEMENT sowie ggfs. DB_FILE_NAME_CONVERT und LOG_FILE_NAME_CONVERT.

Hier gilt die strikte Empfehlung, alle Parameter symmetrisch zu definieren, auch wenn nicht alle für das Aufsetzen des Data Guards sofort benötigt werden. Damit erspart man sich hektische Konfigurationsarbeit, falls der erste Rollenwechsel in einer kritischen Situation erfolgt.

Als letzten Schritt bringt man nun die Standby-Datenbank in den Managed Recovery-Modus. Falls man keine Active Data Guard-Lizenz besitzt, sollte man dies allerdings nur dann tun, wenn die Datenbank im Mount-Status, aber nicht geöffnet ist.

Der Data Guard Broker
Das ist nun die Stelle, an der optional der Broker ins Spiel kommt. Er wird aktiviert, indem man auf beiden Seiten den Server-Parameter DG_BROKER_START auf TRUE setzt. Der Broker ist nun als zusätzlicher DMON-Hintergrundprozess sichtbar.

Die weitere Administration und Konfiguration erfolgt nun über das Kommandozeilenwerkzeug DGMGRL, das wiederum über eine eigene Syntax und Parametrierung verfügt. Im täglichen Umgang zeigt sich z.B. ein gewisser Einarbeitungsbedarf beim Setzen von Hochkommata, die an einigen Stellen erforderlich, an anderen nicht erlaubt sind. Beispiel:

```
CREATE CONFIGURATION 'DGConfig' AS
PRIMARY DATABASE IS 'TESTA'
CONNECT IDENTIFIER IS TESTA;
```

Wichtig ist nun, dass der Broker die Steuerung der Data Guard-Konfiguration komplett übernimmt. Man kann das sofort nach dem Aktivieren einer Konfiguration erkennen, wenn der Broker die Server-Parameter nach seinen Vorstellungen neu setzt (s. Alert-Log-Datei). Parameteränderungen dürfen ab jetzt ausschließlich über EDIT CONFIGURATION-Befehle in DGMGRL abgesetzt werden. Leider werden manuelle Änderungen mit ALTER SYSTEM in SQL*Plus nicht verhindert, so dass man eine Broker-Konfiguration recht einfach „sabotieren“ kann.

Sobald der Broker aktiviert ist, kümmert er sich beim Startup der Datenbankinstanzen um den Managed Recovery-Modus. Ein manuelles Eingreifen bzw. angepasstes Startup-Skript ist nicht mehr erforderlich.

Enterprise Manager
Der Enterprise Manager Cloud Control kann dazu verwendet werden, eine Data Guard-Konfiguration zu erstellen, inklusive aller Voraussetzungen, wie z.B. dem Kopieren der Datenbank. Eine solche Konfiguration ist immer Broker-basiert. Wer also den Enterprise Manager zur Verwaltung von Data Guard einsetzt, kann den Broker nicht außen vor lassen.

Ein Vorteil der Konfiguration mit dem Enterprise Manager ist, dass man nicht zunächst manuell eine Data Guard-Konfiguration aufsetzen muss, um diese dann in eine Broker-Konfiguration zu überführen. Einige Erfahrungen zeigen aber, dass auch mit dem Enterprise Manager die Komplexität von Data Guard nicht vollständig versteckt werden kann.

Zusätzliche Möglichkeiten des Brokers
Der erste große Vorteil des Brokers ist die wesentlich geringere Komplexität beim Rollenwechsel. Bei der manuellen Konfiguration ist sowohl ein Switchover (kontrollierter Rollenwechsel) als auch ein Failover (Reaktion auf einen Fehler der Primärdatenbank) eher eine Vorgehensweise mit mehreren Befehlen. Beides ist im DGMGRL ein kurzer Einzeiler, z.B.:

FAILOVER TO 'TESTB';

Falls der Flashback-Modus konfiguriert ist, ist es möglich, nach einem Failover ein Reinstatement der „kaputten" Datenbank zu versuchen. Hierbei wird ein FLASHBACK DATABASE mit der SCN durchgeführt, zu der die Standby-Datenbank zur Primärdatenbank wurde. Sollte die Datenbank physisch beschädigt sein, so wird dies naturgemäß nicht funktionieren. In vielen anderen Fällen erspart dies ein vollständiges Neuaufsetzen.

Ein viel diskutiertes Feature ist der Fast-Start Failover (FSF). Durch eine Observer-Software, die auf einer nicht in der Data Guard-Konfiguration befindlichen Maschine laufen sollte, wird nach vordefinierten Regeln ein Failover auf eine Standby-Datenbank automatisch durchgeführt. Dabei ist der Failover die Variante, bei der tatsächlich Datenverlust entstehen kann. (Man kann hierfür eine obere Grenze setzen, indem man den maximale „Nachlaufabstand" für einen solchen Failover definieren kann.) Die Observer-Software ist in den DGMGRL-Client integriert, so dass die Installation mit dem „Administrator Client" erfolgen kann.

Für Fast-Start Failover sind der Flashback-Modus sowie Standby-Redolog-Dateien zwingend erforderlich.

Was der Broker nicht löst
Es gibt eine Komponente, die bei Data Guard-Konfigurationen oft zu wenig Beachtung findet: Der Zugriff der Anwendungen auf die Datenbank. Immerhin läuft die Datenbank nach einem Switchover bzw. Failover auf einem System mit einer anderen Netzwerkadresse. In vielen Fällen muss hier manuell nachgearbeitet werden, z.B. bei den Einträgen in den TNSNAMES.ORA-Dateien.

Soll der Verbindungsaufbau automatisch zum neuen System erfolgen, müssen diverse Probleme gelöst werden:

- Die TNS-Adressierung muss über Adresslisten erfolgen.
- Damit die Verbindungen nicht in der falschen Instanz landen muss ein selbstdefinierter Datenbankdienst verwendet werden.
- Die Definition von Diensten für die unterschiedlichen Rollen (Primär- oder Standby-Datenbank) ist mit der Grid Infrastructure möglich. Hat man diese nicht zur Verfügung, kann das z.B. über einen Datenbank-Startup-Trigger gelöst werden.

To Broker Or Not To Broker?
Als Vorteile kann der Broker folgende Punkte für sich verbuchen:

- Einfache Befehle für Switchover und Failover
- Mögliches Reinstatement
- Observer mit Fast-Start Failover
- Integration mit Enterprise Manager Cloud Control
- Automatischer Managed Recovery-Modus beim Startup

Demgegenüber stehen folgende Nachteile:

- Neue Syntax für DGMGRL und/oder Enterprise Manager Cloud Control erforderlich
- Mögliche Beschädigung der Broker-Konfiguration durch manuelle Parameteränderungen
- Konfiguration ohne Fast Recovery Area nicht möglich

Wer noch nicht mit Data Guard gearbeitet hat, die Technologie jedoch genauer kennen lernen möchte, sollte zunächst mit einer manuellen Konfiguration anfangen. Dies ist zwar auf den ersten Blick mehr Aufwand, erlaubt aber bessere Reaktionen auf Konfigurationsfehler. Danach sollte mit Broker-Konfigurationen getestet werden, je nach Vorliebe mit DGMGRL oder dem Enterprise Manager Cloud Control.

Kontaktadresse:
Dierk Lenz
Herrmann & Lenz Services GmbH
Höhestraße 37
51399 Burscheid

Telefon: +49 2174-6712-0
Fax: +49 2174-6712-22
E-Mail dierk.lenz@hl-services.de
Internet: www.hl-services.de