

# Data Guard mit oder ohne Broker: die nächste Runde in 12.2

**Dierk Lenz**  
**Herrmann & Lenz Services GmbH**  
**Burscheid**

## Schlüsselworte

Oracle Database, Data Guard, Data Guard Broker

## Einleitung

Data Guard ist Bestandteil der Enterprise Edition und automatisiert sowohl Log Transport als auch Log Apply für Standby-Datenbanken. Einfache Konfiguration und Robustheit sind bekannte Stärken von Data Guard.

Man hat allerdings bei der Konfiguration die Wahl zwischen manuellem Betrieb und der Verwendung des Data Guard Brokers, wobei beides Vor- und Nachteile hat. Der Vortrag enthält neben neuen Features in 12.2 den aktuellen Stand der Diskussion, ob der Broker verwendet werden soll oder nicht.

## Basics

Zur Vorbereitung des Data Guard Betriebs muss eine physische Kopie der Primärdatenbank erstellt werden – unabhängig von der Broker-Frage. Es empfiehlt sich, einige Konfigurationsschritte vor dem Kopieren der Datenbank durchzuführen; man erspart sich die Konfiguration *auf beiden Seiten*. Hierzu gehört z.B. die Aufgabe, Standby Redologs anzulegen, die für das *Real Time Apply* benötigt werden. Real Time Apply besagt, dass der Log Transport durch den Log Writer LGWR der Primärinstanz durchgeführt. Die Alternative hierzu ist, dem Archiver ARCH den Log Transport zu überlassen. Hierdurch ergibt sich jedoch eine Verzögerung beim Log Transport, da vor der Übertragung auf einen Log Switch sowie das Archivieren gewartet werden muss.

Falls die Frage aufkommt, warum man die Standby Redologs auf beiden Seiten anlegt, wo man sie doch nur für die Standby-Datenbank braucht: Die dringende Empfehlung lautet, eine Data Guard Konfiguration so symmetrisch wie möglich zu halten, damit später ein Rollenwechsel (Switchover oder Failover) auch funktioniert!

Eine weitere Voraussetzung für das vollständige Funktionieren der Standby-Datenbank ist der Force Logging-Modus. Hierbei werden eventuell abgesetzte NOLOGGING-Optionen für bestimmte Befehle übersteuert und ein vollständiges Logging erzwungen.

Beim Erstellen der Datenbankkopie ist der Recovery Manager hilfreich: Das `DUPLICATE DATABASE`-Kommando hat eine `FOR STANDBY`-Option. Hierbei wird der Datenbankname nicht - wie beim `DUPLICATE` sonst üblich - geändert; zusätzlich wird der Wert `STANDBY` für die Datenbankrolle in der Kontrolldatei gesetzt.

Zum Betrieb von Data Guard sind u.a. Folgende Einstellungen zu konfigurieren:

- für den Log Transport einen der verfügbaren `log_archive_dest_<N>`-Parameter,
- unterschiedliche Werte für `db_unique_name` in Primär- und Standby-Datenbank sowie
- die Liste aller beteiligten Datenbanken in `log_archive_config`.

Zusätzlich muss nach dem MOUNT der Standby-Datenbank die automatische Wiederherstellung eingeschaltet werden:

```
ALTER DATABASE RECOVER MANAGED STANDBY DATABASE...
```

Sollte die Standby-Instanz neu gestartet werden, so muss das Managed Recovery ebenfalls erneut gestartet werden.

### **Was übernimmt der Data Guard Broker**

Data Guard stellt als Basispaket Funktionen und Parameter zur Verfügung, um Standby-Datenbanken zu betreiben. Typischerweise befinden sich Primär- und Standby-Datenbanken auf unterschiedlichen Servern, so dass bei der Konfiguration oft hin und her gewechselt werden muss. Die Administration erfolgt "pro Server".

Der Data Guard Broker geht einen Schritt weiter: In einer Konfiguration werden die beteiligten Datenbanken (eine primäre, eine oder mehrere Standbys) beschrieben. Sobald die Konfiguration aktiviert ist, kann man von einer beliebigen Stelle aus alle erforderlichen Administrationsschritte veranlassen. In den meisten Fällen ist es so, dass eine Aktion im Broker viele Schritte bei den beteiligten Datenbanken umfasst. Dies ist insbesondere dann von Vorteil, wenn z.B. gerade die primäre Datenbank ausgefallen ist und schnell ein Failover durchgeführt werden soll.

Die Administration des Brokers kann entweder über den Enterprise Manager Cloud Control (EMCC) oder über das Kommandozeilenwerkzeug DGMGRL erfolgen. Im Folgenden gehen wir von der Verwendung von DGMGRL aus.

Um dies zu bewerkstelligen, gibt es einen eigenen Hintergrundprozess DMON (Data Guard Monitor). Dieser Prozess nimmt die Befehle vom EMCC oder DGMGRL entgegen und führt entsprechende Aktionen aus. Die DMON Prozesse der beteiligten Datenbanken kommunizieren miteinander, so dass die Administration von einer beliebigen Stelle aus erfolgen kann. Zusätzlich automatisiert er einige Abläufe, z.B. das Starten des Managed Recoveries beim Mount der Instanz.

Sind sowohl Broker als auch die Grid Infrastructure GI in Verwendung, so kommunizieren diese ebenfalls miteinander. Z.B. kann der Broker dafür sorgen, dass bei einem Rollenwechsel Services über die GI gestoppt werden, wenn diese einer Rolle PRIMARY oder STANDBY zugewiesen wurden.

Verwendet man DGMGRL, so stößt man auf einen Nachteil des Brokers: Das Werkzeug wird mit einem eigens hierfür entworfenen Befehlssatz gesteuert. Dieser Befehlssatz ist nicht extrem umfangreich, entpuppt sich jedoch als etwas *spröde*, da er sich nur teilweise an typische Oracle-Muster hält. So ist z.B. die obligatorische Verwendung von Anführungszeichen um Datenbanknamen nicht typisch und führt immer wieder zu *Typos*.

Zudem werden auf diversen Ebenen Parameter abgelegt (Konfiguration, Datenbank, Instanz). Diese werden meist auf Server-Parameter abgebildet und haben ähnliche Namen, aber keine identischen. Somit muss man als Administrator zusätzlich zu den *bekannt* Namen noch die Entsprechungen für die Broker-Konfiguration lernen.

Der Data Guard Broker geht zunächst davon aus, dass für die beteiligten Datenbanken konsistente TNS-Aliasnamen konfiguriert sind, mit denen die Datenbanken von überall erreichbar sind. Sobald der Broker gestartet ist, wird eine Konfiguration erstellt, in der für jede Datenbank der zu verwendende TNS-Alias bekannt gemacht wird.

Sind Primär- und Standby-Datenbank konfiguriert, so kann die Konfiguration aktiviert werden (`ENABLE CONFIGURATION`). Hierbei übernimmt der Broker folgende Aufgaben:

- Konfiguration der Server-Parameter `log_archive_dest_<N>`, `log_archive_config` usw.
- Start des Managed Recovery

Der Broker übernimmt auch nach einem Neustart einer Instanz die Aktivierung des Managed Recovery.

Der *Fast Start Failover*, d.h. die Durchführung eines Failovers bei Erkennung eines schwerwiegenden Fehlers ohne Eingreifen eines Administrators, ist ausschließlich mit dem Einsatz des Brokers möglich. Hierzu wird ein sog. *Observer* gestartet, der die Data Guard Komponenten überwacht und ggfs. mit einem Failover eingreift.

### **Beispiel: Das Client-Verbindungs-Dilemma**

Unabhängig von der Tatsache, ob der Broker nun verwendet wird oder nicht, wird in Projekten mit Data Guard oft vergessen zu diskutieren, wie sich Clients an die Datenbank verbinden sollen - insbesondere nach einem Rollentausch der Datenbanken.

Da Primär- und Standby-Datenbank sinnvollerweise auf unterschiedlichen Servern laufen, muss bei der Adressierung über einen TNS-Alias eine Adressliste statt einer einzelnen Adresse verwendet werden. Falls die Anwendung keinen Oracle Client, sondern JDBC nutzt, muss man dort statt der JDBC-Standardsyntax die Oracle TNS-Syntax bei der Definition der Verbindung verwenden.

Für den im TNS-Alias angegebene Service muss nun gelten, dass er ausschließlich auf der aktuellen Primärdatenbank gestartet ist. Sollte er auf beiden Datenbanken gestartet sein, so wird ein Teil der Verbindungen zur Standby-Datenbank aufgebaut werden. Dort laufen sie in einen Fehler (da die Instanz nur gemounted ist).

Das Starten des passenden Service kann man z.B. durch einen Datenbank-Startup-Trigger lösen. Eleganter ist das Zusammenspiel zwischen Broker und GI, bei dem passende Services automatisch gestartet und gestoppt werden.

### **Neue Features beim Data Guard Broker**

Mit Oracle Database 12c Release 2 gibt es einige neue Features:

Zunächst ist im DGMGRL der Befehl `VALIDATE DATABASE DATAFILE` verfügbar, mit dem man die physische Konsistenz einer Datendatei über Primär- und Standby-Datenbank prüfen kann. Hiermit können *Lost Writes* entdeckt werden.

Mit dem `@`-Kommando kann DGMGRL Skripte ausführen. Zusätzlich kann mit `SPOOL` eine Log-Datei erzeugt werden. Mit `SET TIME` werden Zeitstempel ausgegeben, so dass Ausführungszeiten von Kommandos festgehalten werden können.

Die immer größer werdende Berücksichtigung von Pluggable Datenbanken findet sich auch im Data Guard Broker wieder: Mit dem Befehl `MIGRATE PLUGGABLE DATABASE` kann eine Pluggable Datenbank zwischen zwei Container-Datenbanken, die sich in zwei unterschiedlichen Broker-Konfigurationen befinden, migriert werden.

Für eine Broker-Konfiguration können nun mehrere Observer zuständig sein. Somit kann ein Fast Start Failover auch nach dem Ausfall eines Observer durchgeführt werden.

### **Fazit**

Der Data Guard Broker hat über die letzten Versionen stetig an Features zugelegt. Hierdurch sind insbesondere in kritischen Situationen Administrationsschritte schnell und einfach durchführbar. Viele Kleinigkeiten, z.B. Skriptmöglichkeiten beim DGMGRL, runden das Bild ab. Daher kann das Fazit nur lauten:

### **Benutze den Broker!**

#### **Kontaktadresse:**

Dierk Lenz  
Herrmann & Lenz Services GmbH  
Am Ziegelfeld 28  
D-51399 Burscheid

Telefon: +49 (2174) 30710 11  
Fax: +49 (2174) 30710 22  
E-Mail: [dierk.lenz@hl-services.de](mailto:dierk.lenz@hl-services.de)  
Internet: [www.hl-services.de](http://www.hl-services.de)