

# Hochverfügbarkeit

Sebastian Solbach, ORACLE Deutschland B.V. & Co. KG

Bereits die Oracle Maximum Availability Architecture (MAA) in der Datenbank 11g R2 lässt im Bereich „Hochverfügbarkeit“ kaum Wünsche offen, um einen 7x24-Stunden-Betrieb mit perfektem Disaster Recovery zu gewährleisten. Hier stellt sich die Frage, wo 12c noch Verbesserungen einbringen kann.

Eine Lücke hatte die Architektur: Es gab keine vollständige Integration der Clients, um mit der Hochverfügbarkeit der Datenbank transparent umgehen zu können. Genau hier ist nun die größte Änderung der Hochverfügbarkeits-Funktionalitäten in 12c zu finden: „Transaction Guard“ und „Application Continuity“ verbessern für Applikationen und Applikationsserver den Umgang mit der Datenbank. Aber auch in den anderen Bereichen wie „RMAN“, „Grid Infrastruktur“ und „Data Guard“ hat sich einiges getan. Dabei können selbst kleine Features großes Potenzial aufweisen. Dazu gehört auch die Möglichkeit, Partitionen und Datendateien online zu verschieben – eine Grundlage für Advanced Data Optimization.

## Recovery Manager

Auch wenn kein Real Application Cluster (RAC) oder Data Guard im Einsatz ist, ein Backup der Datenbank ist Pflicht. Dementsprechend gibt es auch den Recovery Manager (RMAN). Die RMAN-Neuerungen sind deshalb für jeden Datenbank-Administrator von besonderem Interesse. Zu den wichtigsten gehört sicherlich die Möglichkeit, dass RMAN mit Oracle 12c nun eine einzelne Tabelle aus dem Backup wiederherstellen kann. Zwar hatte auch schon 11g R2 mithilfe der Flashback-Funktionalität entsprechende Möglichkeiten, Inhalte gelöschter Tabellen oder komplette Tabellen wiederherzustellen, diese basierten aber auf den Flashback-Mechanismen. Sie waren daher entweder von der UNDO-Retention (Flashback Table) oder dem verfügbaren Platz für den Recycle Bin (Flashback Drop) beziehungsweise dem zur Verfügung stehenden Platz für die Flashback-Logs (Flashback Database) abhängig. Waren alle Zeiträume abgelaufen, half nur ein auf-

wändiges Point-in-Time-Recovery der Datenbank (PITR) oder des Tablespace (TSPITR).

Mit dem neuen „RECOVER TABLE“-Kommando ist es nun möglich, einzelne Tabellen direkt aus dem Backup wiederherzustellen, wahlweise direkt zurück in die Datenbank oder zuerst einmal in eine Export-Dump-Datei, um die Tabelle später manuell einzulesen oder in ein anderes Schema zu importieren. Die Vorgehensweise ist dabei ähnlich wie bei einem TSPITR, es ist also eine Auxiliary-Datenbank-Instanz erforderlich, die RMAN aber selbstständig erzeugt, startet und wieder löscht. [Listing 1](#) zeigt das Recover-Table-Kommando, um die Tabelle „EMP“ wiederherzustellen. Das Ergebnis ist eine neue Tabelle „EMP\_RECVR“ mit dem Stand von vor vier Tagen.

Die zweite RMAN-Neuerung betrifft die Portabilität von Backups. So lässt sich ein komplettes Backup einer Datenbank etwa unter AIX auf Solaris wiederherstellen. Allerdings muss das Backup vorher für den Transport gekennzeichnet sein. Dies geschieht unter der Angabe „for Transport“ oder

besser noch durch die Angabe der Zielplattform ([siehe Listing 2](#)).

Hat die Zielplattform eine andere Byte-Reihenfolge (Endianness), können zwar nicht das gesamte Backup, aber die einzelnen Tablespaces transportiert werden. Andere RMAN-Verbesserungen betreffen die Möglichkeit, Image-Kopien einzelner Datendateien zu parallelisieren, um zum Beispiel mithilfe der Multi-Sektion Definitionen sowie Datenbank-Kopien mit inkrementellen Backups zu aktualisieren. Dazu dient der Befehl „RECOVER DATABASE ... FROM SERVICE“.

Nicht nur diese Änderungen sind insbesondere für Migrations- und Data-Guard-Umgebungen interessant, da nun das Datenkopieren schneller und detaillierter geschehen kann, sondern auch die Tatsache, dass bei einer Klon-Operation in 12c die eigentlichen Prozesse auf dem Ziel ablaufen und nicht mehr auf dem Primärsystem.

## Transaction Guard and Application Continuity

Ein Kernproblem in der Entwicklung von Applikationen, die mit dem Aus-

```
RECOVER TABLE SCOTT.EMP
UNTIL TIME 'SYSDATE-4'
AUXILIARY DESTINATION '/tmp/backups'
REMAP TABLE 'SCOTT'. 'EMP': 'EMP_RECVR';
```

[Listing 1](#)

```
BACKUP
TO PLATFORM='Linux x86 64-bit'
FORMAT '/tmp/xplat_backups/db_trans_lin.bck'
DATABASE;
```

[Listing 2](#)

fall der Datenbank transparent umgehen sollen, ist der sogenannte „ungewisse Transaktionskontext“. Denn stürzt die Datenbank-Session ab, sind alle Informationen über die aktuelle Transaktion, alle Session-Variablen etc. verloren. Zwar konnten schon seit der Datenbank-Version 8i Funktionalitäten wie TAF ein abgebrochenes Select-Kommando weiterführen, bei abgebrochenen DMLs oder sogar beim Ausbleiben der Commit-Bestätigung war die Applikation aber auf sich allein gestellt. Auch die mit 10g eingeführte Fast-Connect-Failover-Technologie (FCF) hatte eher im Fokus, Applikationen mit Connection Pools so schnell wie möglich zu benachrichtigen, als Applikationen bei der Fehlerbehebung der abgebrochenen Statements zu unterstützen.

Diese fehlenden Informationen über den ungewissen Transaktionskontext kann nun die Datenbank 12c liefern. Zusätzlich bietet sie mithilfe von Transaction Guard sogar ein API, um eine ausgebliebene Commit-Bestätigung zu überprüfen. Hierzu führt die Datenbank eine logische Transaktions-ID für die Session der Clients ein, die der Client kennt und jederzeit (auch nach einem Verbindungsverlust) abfragen und überprüfen kann. Dazu muss die Applikation allerdings über einen Service mit der Datenbank verbunden sein, bei dem der sogenannte „Commit Outcome“ protokolliert wird (siehe Listing 3).

Mithilfe des Transaction-Guard-API kann nun jeder Client, egal ob OCI oder Java, den Erfolg der letzten Transaktionen abfragen. Im Java-Bereich geht die Implementation von Transaction Guard sogar noch einen Schritt weiter. Muss unter OCI die Applikation noch selbst den Status über das API abfragen, was eine Applikationsänderung wäre, ist im neuen Universal-Connection-Pool diese Funktionalität durch die sogenannte „Application Continuity Technologie“ (AC) implizit implementiert.

Für diese Funktionalität merkt sich der Connection-Pool die zuletzt ausgeführten Statements inklusive des abschließenden Commits und kann damit im Fehlerfall genau diese Statements wiederholen. Nachdem Application Continuity im Connection-

```
srvctl modify service -database orcl -service GOLD -commit_
outcome TRUE
```

### Listing 3

Pool zuvor über Transaction Guard den Transaktionsstatus geprüft hat, startet AC dann die Wiederholung der Statements. Allerdings mit allen Eigenheiten eines Replay: Wurden also in diesem Context andere Aktionen durchgeführt, die nicht im Verantwortungsbereich der Datenbank liegen (etwa ein externer Aufruf einer Prozedur), so würde dieser auch nochmals durchgeführt werden. Deshalb ist AC nicht automatisch transparent und erfordert immer noch einen Test daraufhin, ob die Applikation für AC geeignet ist. Andere Voraussetzungen sind:

- Verwendung von J2EE oder JPA unter Einsatz des Standard-JDBC-API

- Verwendung des Universal-Connection-Pools
- Keine externen Aktionen, die nicht wiederholt werden können/dürfen
- Keine Verwendung von XA

Selbstverständlich steht die Funktionalität in den aktuellen Versionen des aktuellen WebLogic-Servers als Erweiterung von Active Grid Link zur Verfügung.

### Global-Data-Services

Einen zentralen Einstiegspunkt für alle Datenbanken innerhalb eines Unternehmens bieten die Global-Data-Services (GDS). Eine ihrer Aufgabe ist es, alle eingehenden Verbindungsanfragen an die optimale Instanz weiterzu-



Abbildung 1: Active Data Guard mit GDS

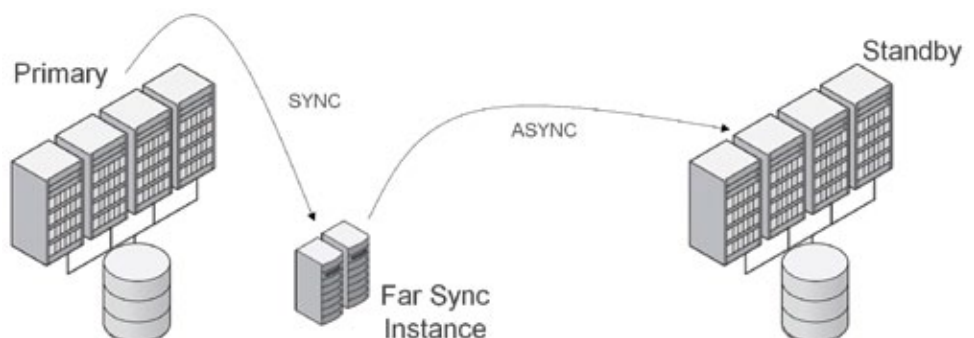


Abbildung 2: Systematische Darstellung von „Far Sync“

leiten und die verfügbaren Datenbank-Services zentral zu verwalten. Das Ziel besteht darin, eine optimale Verteilung zwischen Benutzern und verfügbaren Datenbanken sicherzustellen.

Am besten kann man die positive Auswirkung bei einer Active-Data-Guard-Umgebung (ADG) sehen, in der der Read-Only-Service nur auf der ADG-Seite angeboten wird. Stand die Standby-Datenbank in einer solchen Umgebung unter 11g R2 nicht zur Verfügung, gab es auch keinen Read-Only-Service. Es wäre aber durchaus möglich gewesen, diesen in dem Fall auch auf die Produktiv-Seite oder vielleicht auf eine andere Standby-Datenbank zu verlagern. Genau die Aufgabe erfüllt nun GDS und startet den ausgefallenen Service beispielsweise auf der Primär-Datenbank (siehe [Abbildung 1](#)).

GDS versorgt also nicht nur die Clients mit Load-Balancing-Informationen, etwa darüber, welche RAC-Instanz, Data Guard oder replizierte Umgebung für den Client die besten Antwortzeiten liefert, sondern kann zusätzlich auch die Services dort starten, wo sie gebraucht werden. Vereinfacht lässt sich die GDS-Technologie mit dem SCAN-Listener in 11g R2 vergleichen, allerdings mit zusätzlichen Funktionalitäten bezüglich unternehmensweiter Service-Verteilung für alle Datenbanken und nicht nur für einen RAC-Cluster.

### Data Guard

Der „Zero Data Loss“-Modus von Oracle Data Guard erlaubt es, eine Disaster-Recovery-Umgebung immer auf dem aktuellsten Stand zu halten. Allerdings erfordert dieser Modus, dass die Umge-

bung mit synchronem Redo-Log-Transport arbeitet. Das bedeutet, dass die Primär-Datenbank auf eine Bestätigung der Standby-Umgebung warten muss, bevor sie weiterarbeiten kann. Je größer die Distanz zwischen der Primär-Datenbank und der Standby-Umgebung, desto stärker machen sich die Einbußen bei der Performance bemerkbar.

Oracle 12c Active Data Guard Far Sync behebt dieses Problem, indem es der Primär-Datenbank eine „Far Sync“-Instanz zur Seite stellt. Diese sehr abgespeckte Oracle-Datenbank besteht nur aus folgenden Teilen:

- Standby Control File
- Standby Redo Logs
- Archived Redo Logs

Diese Instanz besitzt keine eigenen Daten-Dateien und dient als eine Art „Redo Log Repeater“, der die Redo-Log-Informationen asynchron an die Standby-Datenbank schickt. Die Primär-Datenbank sendet ihre Redo-Informationen wiederum synchron an diesen Redo Log Repeater (siehe [Abbildung 2](#)).

Im Fehlerfall erhält die Standby-Datenbank die aktuellen Informationen von der Far-Sync-Instanz und ist somit trotz viel größerer Distanz im „Zero Data Loss“-Modus. Damit ist es auch möglich, ein Netzwerk-Problem zwischen Primär- und Standby-Datenbank abzufedern.

Selbstverständlich kann auch der Standby-Datenbank eine „Far Sync“-Instanz zur Seite gestellt werden, damit Zero Data Loss auch bei umgekehrter Rollenverteilung zur Verfügung steht. Die „Far Sync“-Instanzen erfordern dabei keine Lizenz, nur die Data-

Guard-Umgebung für Active Data Guard muss lizenziert sein.

Im Grunde handelt es sich bei einer „Far Sync“-Standby um eine Art kaskadierendes Standby-System – allerdings bei 12c verbessert, da in 11g R2 die kaskadierende Standby-Datenbank immer auf einen Log-Switch der Primärseite warten musste. Neu im Data-Guard-Umfeld ist auch der folgende Transport-Service „SYNC NOAFFIRM“: „LOG\_ARCHIVE\_DEST\_3=’SERVICE=stdby1 SYNC NOAFFIRM’“. Hier reicht es Data Guard aus, dass die Redo-Log-Informationen im Memory des Standby-Servers angekommen sind. Es wird nicht auf die Bestätigung gewartet, dass die Redo-Log-Informationen auch in die Redo-Log-Files geschrieben wurden.

Erheblich vereinfacht wird auch das Rolling Upgrade mit Active Data Guard und Transient Logical Standby. Ab 12c werden nun auch Datentypen wie „Spatial“, „Image“, „Oracle Text“, „DICOM“, „Varrays“ etc. unterstützt, sodass es nur noch wenige Applikationen geben dürfte, die nicht vom Rolling Upgrade profitieren. Die einzelnen Schritte für ein Rolling Upgrade sind bei Active Data Guard nun auch einfach über das Package „DBMS\_ROLLING“ verfügbar. Das vereinfacht die Handhabung des kompletten Prozesses erheblich und schließt viele potenzielle Fehlerquellen aus.

Sonstige Änderungen in Active Data Guard betreffen vor allem die Auslagerung von Read-Only-Services auf Standby-Seite. „Unique Sequences“ in Data-Guard-Umgebungen und die Auslagerung des UNDO von temporären Tabellen in den Temp Tablespace sind dabei von Bedeutung: Dies ermöglicht das Arbeiten mit temporären Tabellen nun auch in einer „Read only“ geöffneten Instanz und bietet somit reinen Reporting-Applikationen einen viel größeren Handlungsfreiraum.

### Flex ASM

Im Bereich des Real Application Clusters und der Grid Infrastruktur 12c ist Flex ASM eine der interessantesten Verbesserungen. Es hebt die starre Verknüpfung der Datenbank-Instanzen auf die lokale ASM-Instanz auf und erlaubt somit auch eine Remote-Verbindung

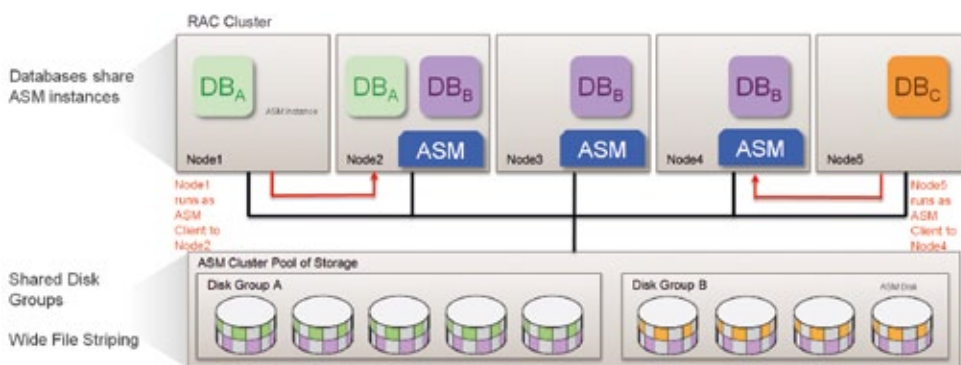


Abbildung 3: Flex ASM

an die ASM-Instanz (siehe [Abbildung 3](#)). Dadurch sind die Datenbank-Instanzen nicht mehr von der lokalen ASM-Instanz abhängig und können auch bei deren Ausfall weiterarbeiten. Nebenbei schont dies bei größeren Clustern die Ressourcen, da nicht mehr auf jedem Knoten eine ASM-Instanz laufen muss.

### Oracle Cloud FS und ACFS

Auch auf Seiten der Lizenzierung gibt es interessante Neuerungen. Diese betreffen insbesondere die Lizenzierung und den Datenbank-Support von ACFS. So sind die Basis-Funktionalitäten vom ASM-Cluster-Filesystem und ASM-Dynamic-Volume-Manager nun für alle Dateien kostenfrei verfügbar. Nur bei Verwendung der erweiterten Funktionalitäten wie „Snapshots“, „Replikation“, „Verschlüsselung“, „Security“ und „Auditing“ ist eine Cloud-Filesystem-

Lizenz notwendig. Dies gilt übrigens auch für 11g-R2-ACFS-Installationen.

Darüber hinaus unterstützt 12c-ACFS nun auch Datenbank-Dateien und damit direkt die Oracle-Datenbank. Zwar sollte aufgrund der Performance die Oracle-Datenbank immer noch direkt in ASM liegen, für Test- und Entwicklungssysteme kann dies aber interessant sein, da ACFS eine Snapshot-Funktionalität besitzt und somit das Klonen von Plugable Databases erheblich beschleunigt.

### Fazit

Database 12c bietet für jeden Bereich der Hochverfügbarkeit kleine, aber feine Neuerungen. Die größten sind mit Sicherheit „Application Continuity“, um die Anbindung von Applikationen zu verbessern, „Far Sync Standby“ für entfernte Disaster-Recovery-Datenbanken im „Zero Data Loss“-Modus sowie

„online Partition“ und Datendatei-Movement für die Implementation einer echten ILM-Strategie. Aber auch in Standard-Funktionen wie RMAN oder ASM gibt es sinnvolle Erweiterungen, die für jede Datenbank im Unternehmen die Hochverfügbarkeit nochmals steigern können.

Sebastian Solbach  
sebastian.solbach@oracle.com





# Sicherheit.

## EURaix-Rechenzentrum. Und Ihre Daten sind sicher!

**Ihre Vorteile:**

- Redundante unabhängige Netzanbindung über mehrere Provider
- Leistungsstarke Internetleitungen
- Höchste Betriebssicherheit, 3 unabhängige Rechenzentrumsstandorte
- Betrieb und Betreuung der IT-Infrastruktur rund um die Uhr, 365 Tage im Jahr
- Höchste Verfügbarkeit der gesamten RZ-Infrastruktur
- Videoüberwachung & Einbruchmeldeanlage



Die regio IT  
ist zertifiziert nach  
ISO 9001, ISO 20000  
und ISO 27001.

```

1100:  .0100110
10011:  .10010100110
..01101110:  .10110***11011
1011011101:  0110  01101
***10011:  01101  101001
01100:  110011  101001
100011 100101  100001
000001 00001  0 11
000101 01000
011 01  0100.  -
1 1  1 01 1
1

```

 regio IT gesellschaft für informationstechnologie mbh  
Sitz der Gesellschaft: Aachen · Niederlassung: Gütersloh · [www.regioit.de](http://www.regioit.de)

 regio IT Jahre