

RMAN Reloaded

– Recovery Manager 12c News & Features –

Oliver Herges
Freelancer
Hannover / Germany

Schlüsselworte

Oracle Recovery Manager, RMAN, Backup & Recovery, Database 12c, New Features 12c, News 12c

RMAN Reloaded

Mit Oracle Database Release 12c bringt RMAN weitere Neuerungen und vor allem interessante Optimierungen mit. Der Vortrag stellt ausgewählte RMAN-Highlights des neuen Releases 12c (12.1.0.2) vor und zeigt, wie diese effektiv eingesetzt werden können.

Multi-Sections für inkrementelle Backups und Image-Kopien

Beginnend mit Database 11g Release 1 konnte ein Datenfile bei der Sicherung in ein Backup-Set in mehrere gleichgroße Abschnitte aus zusammenhängenden Blöcken – den sogenannten "Sections" – zerlegt werden. Jede Section konnte dabei parallel und unabhängig voneinander durch einen Channel gesichert werden.

Mit Database 12c können Multi-Section-Backups nun auch bei inkrementellen Backups verwendet werden:

```
RMAN> BACKUP INCREMENTAL LEVEL 1 SECTION SIZE 2G  
        DATAFILE '/u03/oradata/doag1/users01.dbf';
```

Neu ist ebenfalls, dass RMAN nun Image-Kopien von Datenfiles unter Verwendung von Multi-Sections erstellen kann. Hierbei werden mehrere Channels parallel zur Erstellung der Image-Kopie des Datenfiles herangezogen:

```
RMAN> BACKUP AS COPY SECTION SIZE 2G  
        DATAFILE '/u03/oradata/doag1/users01.dbf';
```

Erweiterungen bei Active Database Duplication

Echte Freude kam bei Database 11g mit dem netzwerkbasierten Duplizierungsmechanismus auf, der ebenfalls zum Aufbau von Standby-Datenbanken genutzt werden konnte. Mit dem DUPLICATE-Kommando und der Klausel "FROM ACTIVE DATABASE" konnte eine Datenbank direkt über das Netzwerk dupliziert werden.

In Database 11g mussten für Active Database Duplication zwingend Image-Kopien eingesetzt werden. Ferner wurden die Work-Prozesse als Target-Channels auf der Quell-Datenbank (Target-Datenbank) ausgeführt. Diese Variante wird weiterhin in Oracle Database 12c unterstützt und wird zur Differenzierung jetzt als "**Push**"-Methode bezeichnet.

Mit Database 12c wird Active Database Duplication als neue "**Pull**"-Methode mit Backup-Sets mit Work-Prozessen als Auxiliary-Channels auf der Ziel-Datenbank (Auxiliary-Datenbank) ausgeführt. Hierzu wird die Klausel "USING BACKUPSET" bei der Duplizierung angegeben:

```

RMAN> DUPLICATE TARGET DATABASE TO test1db
      FROM ACTIVE DATABASE
      USING COMPRESSED BACKUPSET
      SECTION SIZE 2G
      SPFILE
      PARAMETER_VALUE_CONVERT    '/prod1db', '/test1db'
      SET DB_FILE_NAME_CONVERT    '/prod1db', '/test1db'
      SET LOG_FILE_NAME_CONVERT   '/prod1db', '/test1db'
      SET MEMORY_TARGET='400M';

```

In Verbindung mit Backup-Sets können Compressed-Backups, Multi-Section-Backups und auch Verschlüsselung eingesetzt werden. Neben einer schnelleren Übertragung über das Netzwerk wird die Belastung der Quell-Datenbank bei der Duplizierung im Vergleich zur "Push"-Methode vermindert.

Cross-Platform Backup & Restore mit Backup-Sets

Mit "Cross-Platform-Backups & Restore" kann die gesamte Datenbank durch die Verwendung von Backup-Sets auf eine andere Plattform konvertiert und zurückgesichert werden. Dies ist auch mit einzelnen Tablespaces und Datenfiles möglich. Durch den Einsatz von inkrementellen Backups kann der "READ ONLY"-Modus zeitlich erheblich minimiert werden. Prädestiniert ist dieses Feature daher für heterogene Plattform-Migrationen.

Eine Konvertierung der Endianess ist auf diesem Wege nur für einzelne Tablespaces bzw. Datenfiles möglich, aber nicht für die Umstellung der gesamten Datenbank. Die bekannten Voraussetzungen bei der Endianess-Konvertierung von "Cross-Platform Transportable Tablespaces" (seit 10gR1) und "Cross-Platform Transportable Database" (seit 10gR2) bestehen weiterhin. Im Gegensatz dazu ist der Transport mit "Cross-Platform-Backups & Restore" wesentlich einfacher und eleganter durchführbar.

Das Backup-Set muss explizit als Cross-Platform-Backup durch die neuen Klauseln "FOR TRANSPORT" bzw. "TO PLATFORM" angefertigt werden. Die Konvertierung des Backups in eine andere Plattform kann entweder auf der Quell-Datenbank oder auf der Ziel-Datenbank erfolgen (siehe hierzu `V$TRANSPORTABLE_PLATFORM`).

Mit der Klausel "**FOR TRANSPORT**" kann das Cross-Platform-Backup für alle unterstützten Plattformen ohne Konvertierung – quasi generisch – angefertigt werden:

```

RMAN> BACKUP FOR TRANSPORT
      FORMAT 'c:\crossover\db_migration.rman' DATABASE;

```

Mit der Klausel "**TO PLATFORM='...'**" wird das Backup bereits während der Erstellung auf der Quell-Datenbank für eine bestimmte Ziel-Plattform konvertiert:

```

RMAN> BACKUP TO PLATFORM='Linux x86 64-bit'
      FORMAT 'c:\crossover\db_migration.rman' DATABASE;

```

Beim Restore wird mit der Klausel "**ALL FOREIGN DATAFILES**" ein bereits während des Backups konvertiertes Backup auf der Ziel-Plattform wiederhergestellt:

```

RMAN>RESTORE ALL FOREIGN DATAFILES FORMAT '/u03/oradata/datafile_%U'
      FROM BACKUPSET '/u03/crossover/db_migration.rman';

```

Mit der Klausel "**FROM PLATFORM** '...' FOREIGN DATABASE TO NEW" wird während des Restores eine Konvertierung des Backups in die unterstützte Ziel-Plattform durchgeführt:

```
RMAN> RESTORE FROM PLATFORM 'Microsoft Windows x86 64-bit'  
          FOREIGN DATABASE TO NEW  
          FROM BACKUPSET '/u03/crossover/db_migration.rman';
```

Über die Klausel "FOREIGN DATABASE TO NEW" werden alle Objekte – also die gesamte Datenbank – spezifiziert.

DUPLICATE ohne RESETLOGS beim Öffnen der duplizierten Datenbank

Weniger ist mehr. Bei der Duplizierung der Datenbank mit dem DUPLICATE-Kommando wurde die Datenbank bislang abschließend automatisch mit der RESETLOGS-Option geöffnet.

Mit Database 12c kann dies nun mit der "**NOOPEN**"-Klausel verhindert werden:

```
RMAN> DUPLICATE TARGET DATABASE TO testdb  
          FROM ACTIVE DATABASE NOOPEN;
```

Dies ist beispielsweise notwendig, um Einstellungen der Datenbank vor dem Öffnen zu ändern (z.B. für die Konfiguration von Flashback-Datenbank oder schnellen inkrementelle Backups). Ebenfalls ist dies bei Upgrade-Szenarien sinnvoll, wenn die duplizierte Datenbank für ein nachfolgendes Upgrade in den "OPEN UPGRADE"-Mode gebracht werden soll. Das "NOOPEN"-Feature funktioniert im Übrigen auch mit Active Database Duplication.

Restore & Recovery von Files über das Netzwerk mit Services im DataGuard-Verbund

Doppelt hält besser. Dies gilt besonders im DataGuard-Verbund. Mit Database 11g wurde bereits der Austausch von Controlfiles und Backups zwischen Primär- und Standby-Datenbank und umgekehrt unterstützt.

Mit Database 12c können Datenfiles der Primär- und Standby-Datenbank direkt über das Netzwerk für eine Wiederherstellung herangezogen werden. Hierbei werden von RMAN Backup-Sets angefertigt, die zur Wiederherstellung über Oracle-Net zur Ziel-Datenbank transferiert werden.

Für das RESTORE-Kommando wurde hierfür die "**FROM SERVICE**"-Klausel neu eingeführt:

```
RMAN> RESTORE DATAFILE '/u03/oradata/users01.dbf'  
          FROM SERVICE <standby_service_tns> SECTION SIZE 2G;
```

Mit inkrementellen Backups kann über diesen Weg auch ein Roll-Forward und somit eine schnelle Synchronisation der Physical-Standby-Datenbank mit der Primär-Datenbank durchgeführt werden. RMAN generiert auf der Primär-Datenbank ein inkrementelles Backup und transferiert dieses via Netzwerk auf die Standby-Datenbank. Anschließend wird die Standby-Datenbank mit dem inkrementellen Backup recovert:

```
RMAN> RECOVER DATABASE  
          FROM SERVICE <primary_service_tns> NOREDO SECTION SIZE 2G;
```

Eine Synchronisation über inkrementelle Backups ist beispielsweise dann erforderlich, wenn NOLOGGING-Operationen auf der Primär-Datenbank stattgefunden haben, die nicht über den Redo-Log-Stream an die Standby-Datenbank übertragen wurden. Über die "RECOVER ... FROM SERVICE"-Klausel können Datenbanken somit sehr schnell synchronisiert werden. Mit diesem Feature können u.a. auch Multi-Sections und Backup-Compression zur Erhöhung der Performance verwendet werden.

Wiederherstellung von Tabellen und Tabellen-Partitionen aus RMAN-Backups

RMAN 12c ermöglicht ein Point-In-Time-Recovery von Tabellen bzw. Tabellen-Partitionen aus bestehenden Backups. Somit ist mit RMAN im neuen Release eine logische Wiederherstellung möglich, obwohl ein physikalisches Backup durchgeführt wird. Dies ist beispielsweise dann sinnvoll, wenn die Daten nicht mehr via Flashback wiederhergestellt werden können, die Tabelle gelöscht wurde oder logisch korrumpiert ist.

Die Wiederherstellung mit Point-In-Time-Recovery auf Tabellenebene kann altbewährt mit UNTIL TIME, UNTIL SCN oder UNTIL SEQUENCE erfolgen. Beispielsweise:

```
RMAN> RECOVER TABLE SCOTT.EMP, SCOTT.DEPT
        UNTIL TIME 'SYSDATE-1'
        AUXILIARY DESTINATION '/u03/oradata/aux'
        DATAPUMP DESTINATION '/u03/pumps'
        DUMP FILE 'exp_dump.exp'
        NOTABLEIMPORT;
```

RMAN erzeugt für das Point-In-Time-Recovery für Tabellen automatisiert eine Auxiliary-Datenbank mit dem Tablespace des spezifizierten Objektes. Die Tabellen bzw. Partitionen können dann mit Data-Pump aus der Auxiliary-Datenbank wieder in die Ziel-Datenbank importiert werden.

Ferner können Remappings für Tabellen angegeben werden und der Import kann auch zeitversetzt manuell durchgeführt werden ("NOTABLEIMPORT"). Darüber hinaus können Ziele für die Auxiliary-Datenbank ("AUXILIARY DESTINATION") und das Dump-File ("DATAPUMP DESTINATION") festgelegt werden.

Die Methode kann als verbessertes TSPITR-Verfahren betrachtet werden und ist feingranularer direkt auf Objekte bezogen und daher schneller umsetzbar. Ein "BACKUP TABLE ..." -Kommando für eine logische Sicherung auf Objekt-Ebene existiert daher nicht.

RMAN mit Multitenant: Container Databases (CDBs) & Pluggable Databases (PDBs)

Zu den umfassendsten Veränderungen des Datenbank Releases 12c gehört die neue Multitenant-Architektur. RMAN ist in dieses neue Konzept vollständig integriert und kann eine gesamte Container-Datenbank (CDB) oder einzelne Pluggable-Databases (PDBs) sichern und wiederherstellen.

Das "BACKUP ... DATABASE"-Kommando wird im jeweiligen Kontext der Datenbankanmeldung ausgeführt. Bei einer Anmeldung an der CDB an dem Root-Container wird der gesamte Container mit allen PDBs gesichert. Bei einer Anmeldung an der PDB wird die einzelne PDB gesichert.

Der Root-Container der CDB kann auch explizit ohne PDBs gesichert werden:

```
RMAN> BACKUP DATABASE ROOT;
```

Für die Sicherung und Wiederherstellung von Pluggable-Datenbanken (PDBs) wurde die zusätzliche Ebene "PLUGGABLE DATABASE" als Syntax eingeführt.

Bei der Anmeldung an der CDB können PDBs somit explizit gesichert werden:

```
RMAN> BACKUP PLUGGABLE DATABASE pdb1, pdb2;
```

Beispielsweise kann für ein oder mehrere PDBs ein Database-Point-In-Time-Recovery durchgeführt werden, wenn man an der CDB angemeldet ist:

```
RMAN> ALTER PLUGGABLE DATABASE pdb1 CLOSE;
run
{
  SET UNTIL SCN 969696;
  RESTORE PLUGGABLE DATABASE pdb1;
  RECOVER PLUGGABLE DATABASE pdb1;
}
ALTER PLUGGABLE DATABASE pdb1 OPEN RESETLOGS;
```

Bei der Wiederherstellung werden alle Datenfiles der PDB "In-Place" an die ursprüngliche Lokation zurückgesichert und dann recovert. Aufgrund der gemeinsamen Ressourcen-Nutzung müssen die Tablespace UNDO, SYSTEM, SYSAUX und SYSEXT gesondert behandelt werden. Diese müssen zunächst über eine Auxiliary-Datenbank im Root der CDB wiederhergestellt werden.

Durch die Sicherung auf CDB-Ebene kann das Backup für mehrere Datenbanken (PDBs) erheblich vereinfacht und konsolidiert werden. Beispielsweise ist für die Sicherung von Datenbanken keine zeitliche oder ereignisgesteuerte Synchronisation mehr notwendig.

Interface-Verbesserungen am RMAN-Prompt

Prompt mit 12c geliefert: RMAN und SQL-Plus wachsen zusammen. SQL-Kommandos können nun direkt ohne das vorher zwingend benötigte "SQL '<sql_statement>' "-Kommando mit Quotes ausgeführt werden.

Das "**SELECT**"-Kommando ist jetzt auch im RMAN implementiert, was auch einem vereinfachten Skripting zugute kommt. Ebenfalls ist das "**DESCRIBE**"-Kommando implementiert. Formatierungsbefehle wie in SQL-Plus sind jedoch nicht vorhanden.

Neues "**SYSBACKUP**"-Privileg

Für privilegierte Backup- und Recovery-Aufgaben kann jetzt das "SYSBACKUP"-Privileg verwendet werden. Hierdurch lassen sich Backup-Operationen von dem privilegierten Daten-Zugriff als "SYSDBA" rechtemäßig unterscheiden.

Zusammenfassung & Resümee

Mit dem neusten Oracle Database Release 12c perfektioniert und optimiert Oracle bekannte Features. Der Weg zu einem logischen Restore wird mit RMAN 12c für Tabellen und Tabellen-Partitionen eröffnet. Hier ist zu erhoffen, dass es zukünftig weitere Möglichkeiten für die logische Sicherung und Wiederherstellung von Objekten mit RMAN geben könnte.

Highlights von Oracle Database 12c sind "Cross-Platform Backup & Restore" und "Restore & Recovery über das Netzwerk mit Services". Die stringente Integration von RMAN in die neue Multitenant-Architektur rundet das neue Release ab.

Die weitere Integration von SQL-Plus-Funktionalitäten in das RMAN-Interface vereinfachen das "Daily Life" des Oracle DBAs.

Literaturnachweis:

Oracle Database Backup and Recovery User's Guide 12c Release 1 (12.1), E50658-05, September 2014

Oracle Database Backup and Recovery Reference 12c Release 1 (12.1), E50791-04, September 2014

Oracle Database New Features Guide 12c Release 1 (12.1), E49322-08, July 2014

RMAN Enhancements in Oracle 12c (Doc ID 1534487.1)

RMAN Pluggable Database Point in Time Recovery (Note 1521075.1)

RMAN Pluggable Database Backup and Recovery (Note 1521005.1)

RMAN RECOVER TABLE Feature New to Oracle Database 12c (Note 1521524.1)

Kontaktadresse:

Oliver Herges, Freelancer / freiberuflicher Principal Consultant

Oliver Herges Database Consulting

Prinzenstraße 3

D-30159 Hannover

XING: www.xing.com/profile/Oliver_Herges

Telefon: +49 (0) 511 - 67 950 500

E-Mail: Oliver.Herges@gmx.de