

Wien, im April 2008

Backup und Recovery in Oracle 10g

181.199 Seminar (mit Bachelorarbeit)

Eingereicht von:
Dorde Torbica



Technische Universität Wien
Fakultät für Informatik
Abteilung für Datenbanken und Artificial Intelligence

Betreuer:
Univ.Prof. Mag.rer.nat. Dr.techn. ReinhardPichler

Inhalt

| | |
|--|----|
| Kurzzusammenfassung | 3 |
| Einführung in Backups | 4 |
| Einführung in ORACLE Datenbank | 6 |
| Oracle Physische Komponenten | 6 |
| Oracle Logische Komponenten | 7 |
| RMAN Konfiguration und Architektur | 7 |
| Server managed recovery | 7 |
| RMAN Werkzeug | 8 |
| SYS Paketen | 8 |
| Recovery Catalog | 9 |
| Auxiliary Datenbank | 10 |
| Configure | 10 |
| Backup | 12 |
| Offline Backups | 12 |
| Backup Options | 13 |
| Online Backup | 16 |
| Kopien von Datenbanken mittels RMAN erstellen | 17 |
| Inkrementell RMAN Backups | 18 |
| Recovery | 20 |
| Restoring SPFILE | 21 |
| Restoring Control file | 21 |
| Restore und Recovery in NOARCHIVELOG Modus | 22 |
| Restore und Recovery in ARCHIVELOG Modus | 23 |
| Wiederherstellen von Tablespace in ein bestimmte Zeitpunkt | 24 |
| Der Vergleich mit MySQL Backup und Recovery Lösung | 26 |
| Schlussfolgerung | 30 |
| Skripten | 32 |
| Datenbanken ohne aktuelles Backup | 32 |
| Größe aller Datenbanken | 33 |
| Informationen über Backup | 34 |
| Literatur | 35 |

Kurzzusammenfassung

In dieser Arbeit wird Oracle Werkzeug für Backup und Recovery RMAN näher betrachtet werden. Dieses Werkzeug ist mit allen Oracle Datenbanken geliefert und wird standardmäßig in vielen Unternehmen durch „third party“ Werkzeugen (Veritas NetBackup, EMC NetWorker, Tivoli Storage Manager) erweitert, aber auch als einzelnes Werkzeug für Sicherung und Wiederherstellung der Oracle Datenbanken verwendet.

Jedes Backup und Recovery Verfahren involviert sorgfältige Planung und Entwicklung von einer Backup Strategie. Anforderungen an gute Backup Strategie und allgemeine Einführung in Backup Verfahren ist in erstem Kapitel erklärt.

RMAN ist eng mit Datenbank Kern verbindet und für erfolgreiche Implementierung von Backup Strategien ist unbedingt gutes grundlegendes Verstehen von Oracle und RMAN Architektur vorausgesetzt. Diese zwei Themen werden in nächsten zwei Kapiteln behandelt werden.

Verschiedene Arten von Backup Verfahren mit RMAN werden in Backup Kapitel betrachtet. Hier wird gezeigt, wie man mit RMAN online, offline und inkrementelle Backups erzeugen kann. Es wird auch kurz über praktische Möglichkeit für Kopieren von Datenbanken mittels RMAN diskutiert.

Recovery Verfahren ist zweites wichtiges Vorhaben bei Backup Strategie. Erst nach der Wiederherstellung der Daten wird festgestellt werden, wie gut Backup Strategie war. Datenbanken werden immer mehr von nicht erfahrenen Benutzern verwendet, und auf anderer Seite ihre Größe wächst ständig. Gerade deswegen ist Wiederherstellung in einen bestimmten Zeitpunkt äußerst wichtig. Diese Themen werden im Kapitel Recovery näher betrachtet.

Schließlich wird RMAN mit einem Werkzeug für Backup von einer anderen Datenbank verglichen werden. Für dieses Experiment ist Werkzeug mysqldump von populärer Open Source MySQL Datenbank gewählt.

Obwohl RMAN viele eigene Optionen und Features bietet, können mehrere wiederkehrende Aufgaben über Skripten viel komfortabler gelöst werden. Drei Skripten (Ermittlung von Datenbanken ohne Backups, Backup Informationen für alle Datenbanken und Größe von allen Datenbanken) werden in letztem Kapitel vorgestellt.

Einführung in Backups

Sehr oft erinnern sich Datenbank Administratoren (DBA) an ihre Backup Strategie erst nach einem Datenverlust. Dann kann man klar feststellen, wie stark Business Prozesse von der Datenverfügbarkeit abhängig sind. Trotz der Wichtigkeit von den Backups, viele Datenbank Administratoren geben nicht Aufmerksamkeit und Mühe in diesem Bereich. Eingut funktionierender Backup und Recovery System in einer großen Datenbank zu implementieren ist alles außer leichter Aufgabe. Die „faule Administratoren“ vermeiden diese Arbeit mit der Verwendung von einfachsten Werkzeugen wie z.B. copy Kommando in ihrem Betriebssystem. Leider, alle Ersparungen an Zeit, Geld und andere Ressourcen werden wahrscheinlich nach erstem Absturz der Datenbank mindestens doppelt bezahlt werden.

Obwohl es trivial scheint, ist die Frage „Wieso müssen wir Daten sichern?“ zu beantworten. Einige von häufigsten Antworten sind:

- Business-Anforderungen
- Hardware-Fehler
- Disaster Recovery (DR)
- Schutz vor Anwendung Absturz
- Schutz vor Anwender-Fehler
- Rechtliche Anforderungen

Ein zentrales Thema, wenn es um Backup geht, ist Backup und Recovery Strategie.

Es darf man nicht einfach eine Entscheidung „Ab Heute werde ich meine Daten sichern“ treffen. Die gute Backup und Recovery Strategie binden viel sorgfältige Planung und Denkarbeit ein. Ebenfalls muss man auf End-Benutzern der Datenbank achten. Um ihre bestimmte Anforderungen und Erwartungen von der Datenbank Software zu sammeln, müssen sie auf einige Fragen beantworten:

Wie großer Datenverlust ist erlaubt?

Das ist vielleicht die wichtigste Frage, da durch diese Antwort gesamte Planung bestimmt wird. Jede Datenbank hat ein Maß von erlaubtem Datenverlust. Einige Datenbanken sind wenig tolerant, einige mehr. Am Anfang muss man feststellen, wie viele Daten enthalten sind und wo befinden sich diese Daten in der Datenbank. Erst dann kann man bestimmen, welche Daten mit Backups abgedeckt werden. Es ist wichtig den Benutzern den Maß von Datenverlust deutlich zu erklären, damit sie verstehen, wie viel Risiken, mit ausgewähltem Backup und Recovery Plan kommt.

Was ist maximale Zeit für den Recovery Prozess?

Verschiedene Technologien für Recovery haben verschiedene Dauer. Im Prinzip es gilt die Regel „Je Recovery schneller ist, desto teurer ist“. So sind zB. die Festplatten teurere Lösung als Bänder. Es ist auf jeden Fall wichtig, dass End-Benutzer weiß, wie lang in Fall von Totalschaden Recovery dauern wird.

Wie sollen Backups Kopien geschützt werden?

Um Schädigungen durch Ereignisse in der Umgebung (Feuer, Wasser) zu vermeiden, sollen Backups Kopien auf einen alternativen Ort gehalten werden. Durch wachsende Internet Geschwindigkeit kann man immer schneller Backups Kopien über Internet schicken, damit Business Prozessen schnell fortgesetzt werden können. Andere Möglichkeit ist Verwendung von externen Medien wie z.B. Bänder, DVDs, CDs, aber man muss auch darauf achten, dass diese Medien eine beschränkte Nutzungsdauer haben.

Eine gute Gewohnheit ist mindestens zwei Backupsets auf je zwei verschiedene Medien zu sichern.

Kann Datenbank während dieses Prozesses außer Betrieb sein?

Das Backup Verfahren beeinflusst sehr stark die Effizienz der Datenbank. Offline Backups sind billigere Variante, aber Datenbank ist während Backup Prozesses für die Benutzerunerreichbar. Auf der anderen Seite ist Datenbank während Online Backups für alle Benutzer erreichbar.

Wieoftsollen Backups versichert werden?

Wenn man Antworten auf alle vorherige Fragen hat, sollte man schon eine grobe Schätzung über Frequenz von Backups haben. Das Hauptproblem ist ein Zeitplan zu finden, der das beste Verhältnis zwischen dem Grad der Sicherheit und dem Medienverbrauch bietet. Strategie, die Backups zu oft vorgesehen hat, ist sehr riskant, weil „oft Backups machen“ heißt mehrere Bänder, mehr Verschieben von Daten und mehr Administration. Alle diese Vorhaben bringen ein weiteres Risiko.

Einführung in ORACLE Datenbank

Die Backup und Recovery Prozesse sind eng mit der Datenbank verbunden. Trotz der Komplexität der Oracle Datenbank ist äußerlich wichtig grundlegende Teile der Architektur von der Datenbank gut zu verstehen. In diesem Kapitel werden die wichtigsten physischen und logischen Strukturen der Oracle Datenbank erklärt werden.

Oracle Physische Komponenten

Diese Komponenten sind physisch auf dem Dateisystem geschrieben. Die Aufgabe von Backup ist gerade Sicherung von diesen Daten. Um Datenbank wieder zu befähigen, ist ebenfalls notwendig diese wiederherzustellen.

Folgende Strukturen sind Bestandteile von Physische Architektur:

- Datenbank Datafiles

Alle Daten, die in Datenbank eingetragen sind, werden in diese Dateien geschrieben. Diese Dateien sind physische Abbildungen der Tablespaces. Jedes Datafile enthält eine Kopfzeile, wo aktueller Zustand des Datafiles geschrieben ist. Diese Kopfzeile ist nach jedem Checkpoint geupdated. Drei wichtigste Zustände, aus der Sicht des Backups sind: ONLINE, OFFLINE und RECOVER.

- Online redo Logs

Diese Dateien enthalten komplette Liste von einzelnen atomischen Veränderungen in der Datenbank. Jede Datenbank besitzt mindestens zwei Gruppen von online redo Logs, und eine gute Praxis ist wegen der Sicherheit und Performance noch mehrere Gruppen zu haben.

Jede Gruppe kann mehrere physische Dateien enthalten und diese können auf den verschiedenen Medien sein. Oracle schreibt gleichzeitig mit round-robin Art in mehrere Dateien innerhalb von einer Gruppe, aber immer nur in eine Gruppe.

„Change Vectors“ sind kleinere Einheiten innerhalb online redo logs und jede repräsentiert eine atomische Veränderung in der Datenbank. Die „Change Vectors“ sind nach „System Change Number“ (SCN) geordnet. SCN ist ein Zähler, und jeder Wert des Zählers repräsentiert Datenbank in einem Zeitpunkt. Während Recovery der Datenbank werden diesen „Change Vectors“ in umkehrende Reihenfolge an Datenbank angewendet werden.

Die genauere Funktionsweise von Redo Logs ist im Kapitel 6 von [\[2\]](#) beschrieben.

- Archived redo Logs

Sogenannte „log switch“ passiert sich, wenn Oracle das Schreiben in ein online redo Log beendet und fängt das Schreiben in Andere an. Wenn Datenbank im ARCHIVELOG Modus eingestellt ist und ARCH Prozess läuft, wird eine Kopie von online redo Log. Diese Kopie ist Archived redo Log genannt. Oracle kann diese Kopie nach zehn verschiedenen Lokationen kopieren.

- Database ControlFiles

Database Control file enthält verschiedene wichtige Informationen über Datenbank, wie zB. aktuelle SCN, der Zustand von „Database Datafiles“ und aktuelles Status der Datenbank. In Bezug auf RMAN ist Database Control File wichtig, da alle kritischen Informationen über die Backup und Recovery Operationen hier gespeichert werden. Wenn diese Datei beschädigt ist, muss man die unbedingt wiedererzeugen.

- Flashback Logs

Eine neue Feature von Oracle 10g sind Flashback Logs, die ein Rückkehr der Datenbank in bestimmten Zeitpunkt ermöglichen. Diese Dateien sind in Flash Recovery Area (FRA) gespeichert. Oracle alleine sorgt über das Verwalten von Flashback logs. Diese Loge werden von RMAN bei Backup nicht berücksichtigt und für Recovery sind nicht notwendig.

- Oracle Tablespaces

Wegen der Komplexität von Oracle Architektur sind einige Komponenten schwer zwischen physischen und logischen zu unterscheiden. Diese Komponenten werden oft als physisch-logische Komponente bezeichnet. Perfektes Beispiel sind Tablespaces. Tablespaces sind logische Zusammenhang zwischen mehreren logischen Komponenten, aber auch physisch auf der Festplatte (oder Band) in mehrere Datafiles dargestellt. Wichtig ist, dass jede Datafile zu genau einem Tablespace gehört. Die Tablespaces können mehrere (selbsterklärende) Zustände haben: ONLINE, OFFLINE, READ ONLY, READ WRITE.

Oracle Logische Komponenten

Oracle unterstützt mehrere logische Komponenten wie zB. Tabellen, Indizes, Sichten (Views), Clusters, benutzerdefinierte Objekten usw. Allerwichtigstes Ziel von jeder Backup und Recovery Strategie ist Konsistenz von allen Daten in einen bestimmten Zeitpunkt. Die Folgerung von Inkonsistenz ist sogar immer Datenverlust und Oracle bietet RMAN als bestes Werkzeug um diese zu vermeiden.

RMAN Konfiguration und Architektur

Server managed recovery

In Zeiten vor RMAN war es üblich einige Skripten für Backup und Recovery zu schreiben. Skripten mussten zuerst Dateinamen sammeln, dann Dateien mit Tablespaces verknüpfen und schließlich mit Betriebssystem Kommando kopieren. Wegen der Komplexität der Aufgabe haben DBAs viel Zeit investiert, aber trotzdem waren Skripten sehr oft fehlerhaft. Heutzutage, wenn Grid Datenbanken immer mehr an Bedeutung bekommen, ist es praktisch unmöglich sich über alle Details zu kümmern. RMAN ist ein Werkzeug, welches „server managed recovery“ (SMR) implementiert. SMR heißt einfach, dass sich Datenbank selbst über ihren Zustand kümmert, um die Backup zu ermöglichen. Niemand kennt sich besser über Datenbank als sie selbst!

RMAN Werkzeug

RMAN ist eine Konsole-Anwendung und auf ersten Blick kann man sehr leicht RMAN mit SQLPlus vermischen. RMAN ist von ORACLE empfohlen und ist mit allen Versionen von ORACLE 10g (auch Version 11g) mitgeliefert. Es ist sehr kernnahe geschrieben, und macht nicht mehr als Übersetzung von dem Benutzer eingegebenen Kommandos in „Remote Procedure Calls“ (RPC), welche in Datenbank ausgeführt werden.

Ein wichtiger Punkt, welcher oft von den DBAs missverstanden ist, dass RMAN nur koordiniert, und Backup und Recovery werden von der Datenbank selbst geleistet. Jede ORACLE Datenbank kommt mit vorinstallierten PL/SQL Blocken, die von RMAN angerufen werden.

Zwei Hauptteile von RMAN sind ausführbares Programm und recover.bsq Datei. Die Versionen von beiden Dateien müssen übereinstimmen! Während der Ausführung nimmt RMAN aus recover.bsq alle Regeln für Übersetzung von Benutzer Kommandos in RPCs. Es ist richtig zu sagen, dass Backup von recover.bsq gesteuert ist.

Um RMAN zu benutzen, muss Benutzer unbedingt über sysdba Rechte verfügen. Die Gründe sind folgende:

- Zugang zu Paketen aus der SYS-Schema
- Starten und Stoppen von der Datenbank

Einloggen als sysdba ist standardmäßig eingestellt, und muss man nicht immer „as sysdba“ wie bei SQLPlus in connect string eintragen:

```
RMAN>connect target sys/password
```

```
Mit Ziel-Datenbank verbunden: ORCL (DBID=1175756132)
```

SYS Paketen

Die ganze Funktionalität von RMAN Server Prozess ist von zwei Paketen aus SYS-Schema ermöglicht:

- **SYS.DBMS_RCVMAN**
Dieses Paket liest aus Control File die Informationen über Datenbank Schema, und leitet diese zu RMAN Server Prozess, damit diese richtige Dateien für Backup und Recovery wählen kann. Ebenfalls ist dieses Paket für TIME und Checkpoints Operators von Datafiles Headers verantwortlich. Es überprüft Pfad und Größe von Dateien und beeinflusst stark die Effizienz von gesamtem Prozess.
- **SYS.DBMS_BACKUP_RESTORE**
Die Paket SYS.DBMS_RCVMAN führt die Verifikation von angeforderten Informationen und schließlich schickt diese Informationen an RMAN zurück, welche PL/SQL Quellcode basierend auf recover.bsq erzeugt. Diese PL/SQL Quellcode enthält eine Reihenfolge nach Effizienzbalancierte Anrufer Methoden aus SYS.DBMS_BACKUP_RESTORE Paket, die eigentlich Backup von Datafiles, Control Files und archived redo logs gesichert. Nach erfolgreicher Operation schreibt dieses Paket in ein Log Datei die Informationen über Backup (wie zB. Name, Zeit, Größe).

Recovery Catalog

Recovery Catalog ist, neben RMAN Werkzeug und interne DatenbankenPaketen, der dritte wichtige Punkt in RMAN Architektur, obwohl seine Verwendung optional ist. Recovery Catalog ist eine Sammlung von Tabellen,Sichten und Indexes, die Metadaten über RMAN Backups enthalten. Es sind Informationen über Anzahl,Größe und Umfang von jedem Backup für alle Datenbanken gespeichert und können jederzeit von SQLPlus angerufen werden. Diese Daten werden mit Kommando rsync von RMAN mit Control File der Datenbank synchronisiert. Recovery Catalog im Vergleich mit Control File verfügt über Backup Informationen nicht für einzige Datenbank sondern für alle Datenbanken in Enterprise.

Die Basis Tabellen sind sehr kompliziert und nicht für DBAs geeignet. Deswegen bietet Oracle eine Reihenfolge von Sichten mit Präfix RC_, welche für Abholen von Informationen benutzt werden können.Jede manuelle DML Anweisung gegen diese Tabellen ist auf jeden Fall zu verweigern!

Recovery Catalog bringt mit sich einige Vorteile:

- Stored RMAN Skripten
- Er bietet eine zentralistische Informationsquelle für alle Datenbanken in Enterprise
- Er bietet mehr Flexibilität bei Berichten.
- Nur über Recovery Catalog kann man einige Einstellungen als standardmäßige für alle Channels einsetzen.

Mit folgendemKommando kann man auf Recovery Catalog verbinden und initial Synchronisation durchführen:

```
RMAN> connect catalog RMAN/password@rcat;
```

Wenn Recovery Catalog nicht erstellt ist,muss man ihn zuerst initalisieren:

```
RMAN>create catalog tablespace example;
```

Um Datenbank bei Recovery Catalog zu registrieren,sind folgende Schritten durchzuführen:

```
set ORACLE_SID = orcl
RMAN target = sys/password
RMAN>CATALOG=rcat_user/password@recover
RMAN>register database;
```

RCAT_USER muss das Mitglied von den Gruppen connect,resource und recovery_catalog_owner sein:

```
GRANT connect,resource,recovery_catalog_owner TO rcat_user;
```

Wichtig ist dennoch zu erwähnen, dass Benutzer für Verbindung an Recovery Catalog keine sysdba Rechte haben muss. Ein paar nützliche Tipps über Recovery Catalog Sichten werden im Kapitel Skripten behandelt.

Auxiliary Datenbank

Unter Auxiliary Datenbank versteht man eine Instanz der Datenbank, in welcher die wiederherstellte Daten von Zieltatenbank im Fall von Tablespace Point in Time Recovery (TSPITR), Klonieren oder Erstellung von standby Datenbank von RMAN gespeichert werden. Während Durchführung von jeder dieser Tätigkeit muss man sich unbedingt gleichzeitig sowohl mit der Zieltatenbank als auch mit der Auxiliary Datenbank verbinden:

```
RMAN> connect auxiliary sys/password@aux1
```

Um auf SYS.DBMS_BACKUP_RESTORE und SYS.DBMS_RCVMAN zu zugreifen, fordert auxiliary Datenbank Einloggen mit dem Benutzer aus der sysdba-Gruppe. Es ist auch nötig Oracle Net Verbindung richtig einzustellen, weil man sich nur remote mit Auxiliary Datenbank verbinden kann. Die Versionen von der Zieltatenbank und der Auxiliary Datenbank müssen übereinstimmen, sonst können Kompatibilität-Problemen eintreten.

Configure

Erster Schritt bei Konfiguration von RMAN ist sogar immer Diskussion, ob Datenbank in ARCHIVELOG oder NOARCHIVELOG ausgeführt wird. Standardmäßig werden alle Versionen von Oracle 10g in NOARCHIVELOG Modus gesetzt. Umsetzung in ARCHIVELOG ist durch folgende Schritten möglich:

1. Zuerst müssen Größe und Pfad für Recovery File bestimmt werden:

```
SQL>alter system set db_recovery_file_dest_size=4G;
SQL>alter system set db_recovery_file_dest= 'c:\oracle\product\
flash_recovery_area';
```

2. Es ist möglich bis zu 10 verschiedenen Destinationen für ArchiveLogs zu wählen. Destination 10 stellt immer Flash Recovery Area:

```
SQL>alter system set log_archive_dest_1='location=g:\archive';
SQL>alter system set log_archive_dest_10='location=
USE_DB_RECOVERY_FILE_DEST;
```

3. Shut down und mount von Datenbank:

```
SQL>shutdown immediate;
SQL>startup mount;
```

4. Setzen Datenbank in ARCHIVELOG Modus und das Öffnen der Datenbank

```
SQL>alter database archivelog;
SQL>alter database open;
```

Mit RMAN lässt sich Backup Prozess völlig automatisieren. Um diese Möglichkeit zu erleichtern, bietet RMAN eine Reihenfolge von Einstellungen, die man per Kommando configure einstellen kann. Wenn man Default Einstellungen gut wählt, kann man sehr viel Zeit im gesamten Prozess sparen.

Channel ist ein RMAN Server Prozess, welche gestartet wird, wenn Datenbank mit einem I/O Gerät (Festplatte oder Band) kommunizieren will. Standardmäßige Channel für Backups ist mit der Festplatte verbunden. Andere Möglichkeit ist diese Channel mit dem Band zu assoziieren:

```
RMAN>CONFIGURE DEVICE TYPE TO SBT;
```

Jede Channel kann mit verschiedenen Arten von Backups verbunden werden (außer bei Kopieren, das immer mit DISK verbindet werden muss):

```
RMAN>CONFIGURE DEVICE TYPE DISK BACKUP TYPE TO BACKUPSET;  
RMAN>CONFIGURE DEVICE TYPE DISK BACKUP TYPE TO COMPRESSED BACKUPSET;  
RMAN>CONFIGURE DEVICE TYPE DISK BACKUP TYPE TO COPY;
```

Über Parameter PARALLELISM ist es möglich mehrere Channels, die verschiedene Pfade haben, für ein Backup allozieren. Die Zahl bei PARALLELISM wird nach Anzahl von Laufwerken bestimmt werden:

```
RMAN>CONFIGURE DEVICE TYPE DISK PARALLELISM 2;  
RMAN>CONFIGURE CHANNEL 1 DEVICE TYPE DISK FORMAT 'd:\backup_%U';  
RMAN>CONFIGURE CHANNEL 1 DEVICE TYPE DISK FORMAT 'e:\backup_%U';
```

Wenn Bände als Medium benutzt werden, bietet RMAN sehr feine Einstellungen für diese Channels. Über den Parameter maxpiecesize kann man Größe von einzelnen Backup Pieces kontrollieren. Wegen der Effizienz bei dem Schreiben der Daten (All Bände schreiben sequentiell) ist möglich Anzahl von geöffneten Dateien zu beschränken (Parameter maxopenfiles) und Größe der Paketen über dem Parameter rate mit mehreren Einheiten zu bestimmen.

```
RMAN>CONFIGURE CHANNEL 1 DEVICE TYPE DISK MAXPIECESIZE 100m  
MAXOPENFILES 4 RATE 100m;
```

Maximale Größe von den Backup Pieces soll nicht mit gesamter Größe der Backup Sets vermischt werden. Über den Parameter maxsetsize kann man diese beschränken oder Beschränkung loschen.

```
RMAN>CONFIGURE MAXSETSIZE TO 4g;  
RMAN>CONFIGURE MAXSETSIZE CLEAR;
```

Alle Einstellungen für ein Chanel sind über den Parameter clear wiederherzustellen:

```
RMAN>CONFIGURE CHANNEL DEVICE TYPE DISK CLEAR;
```

Backup

Nachdem Backup Strategie ausgeplant ist, Datenbankkonfiguriert ist, und RMAN eingestellt ist, ist Zeit gekommen, dass man Hauptaufgabe zu erledigt. Da sie komplex sind, werden Online und Offline Backups getrennt betrachtet.

Offline Backups

Bevor Offline Backup durchgeführt ist, sind zwei obligatorische Vorbedingungen nötig:

- Datenbank muss in gemounted Zustand sein
- RMAN muss mit der Datenbank verbunden sein (CONNECT Kommando)

```
RMAN> shutdown
RMAN> startup mount
RMAN> backup database
RMAN> startup
```

Mit diesen Befehlen wird ein Backup mit standard vordefinierten Optionen gemacht. Wenn standard Optionen nicht geändert sind, werden Datenbank Datafiles, Control File und SPFILE gesichert. RMAN unterstützt in aktueller Version Sicherung von den Konfigurationsdateien (INIT.ORA, LISTENER.ORA, SQLNET.ORA, TNSNAMES.ORA) nicht.

Das Kommando run ermöglicht dem DBA standardmäßige Einstellungen zu überladen:

```
RMAN> shutdown
RMAN> startup mount
run {
allocate channel c1 device type disk format 'd:\Backup\db1_%U';
allocate channel c2 device type disk format 'f:\Backup\db1_%U';
backup as compressed backupset database;
backup current controlfile;
}
```

Parameter %U ermöglicht automatische eindeutige Benennung von Backup Sets. Der Schlüssel für achtstellige Suffix ist Backup Set Bezeichner plus Zeit von der Erstellung des Backup Sets. Schließlich kommen zwei Unterstriche, nachdem ersten Unterstrich kommt die Nummer von Backup Piece, und nachdem zweiten Unterstrich kommt Nummer von Kopie. So wäre z.B. „db1_16E112V9_1_1“ ein gültiger Name.

Obwohl Channels in Kapitel über Konfiguration schon diskutiert sind, müssen diese wenig weiter erklärt werden. „Allocate channel“ Kommando bestimmt Laufwerke für einzelne Backup Sets. Das Laufwerk kann entweder eine Festplatte (Parameter type ist „disk“) oder Bänder (Parameter type ist „sbt“) sein. Sehr oft muss man für verschiedene Bänder weitere Parameter über params übergeben:

```
allocate channel t1 type sbt params= 'ENV=(NB_ORA_CLASS=RMAN_db01)';
```

Immer wenn ein Channel geöffnet ist, wird automatisch ein Backup Set erstellt. Backup Sets stellen logische Einheit von Backups in Oracle dar, und „Backup Pieces“ stellen physische Einheiten (dh. Dateien auf dem Laufwerk). Über das Backup Kommando kann man gesamte Größe von Backup Set einstellen. Dasselbe kann man für Backup Pieces über allocate channel Kommando einstellen.

```
RMAN> allocate channel t1 type disk maxpiecesize = 1g;
```

Parameter „1g“ beschränkt Größe von einzelnen Backup Pieces auf ein Gigabyte. Andere mögliche Kürzel sind k (für das Kilobyte) und m (für das Megabyte). Diese Parameter bieten feine Einstellung von Größe Backup Pieces, damit diese zB. auf einem Band gespeichert werden können.

Backup Options

Kommando Backup bietet auf den ersten Blick nur grundlegende Funktionalität. Trotz der einfachen Syntax lässt sich Backup Kommando über den Parametern sehr fein konfigurieren zB. die Benennung Regeln für einzelne Backup Pieces, der Einfluss auf Effizienz der Datenbank, die Komprimierung usw.

Komprimierung

RMAN bietet zwei Möglichkeiten für Komprimierung, um Größen von Backup Sets zu reduzieren:

- **NULL Data Block Komprimierung**
Diese Komprimierung ist standardmäßige Option. Jeder Block, in welchem niemals Daten geschrieben sind, wird automatisch aus Backup Set ausgeschlossen werden. Wenn ein leer Block bestimmte Bedingungen erfüllt, wird dieser nicht in Backup inkludiert.
- **RMAN Backup Komprimierung**
Eine von den meist diskutierten Neuheiten in Oracle 10g ist sicher Komprimierung von Backup Sets. Oracle benutzt selbstentwickelte Algorithmen, die eine Ersparnis zwischen 67%-72% bringt. Bei Backup von mit Oracle gelieferte Tablespace „example“ ist Unterschied rund 70% (20 Mb im Vergleich zu 66 Mb ohne Komprimierung). Um diese Komprimierung zu aktivieren, muss man Parameter „as compressed backupset“ anwenden.

Tags

Damit verschieden Backup Sets besser organisiert sind, hat Oracle 10g Tags eingeführt. Volles Backup, Tablespace Backup, Datafile Backup, inkrementell Backup und sogar Backup Kopien können mit verschiedenen Tags versetzt werden. Dasselbe Tag kann zu mehreren Backup Sets zugeordnet werden.

```
RMAN> backup database tag='optional backup';
```

Beschränkung vom Backup Einfluss auf der Datenbank

Die Größe von heutigen Datenbanken ist meistens über mehrere Gigabytes. Die Dauer von Offline Backup hat sehr große Bedeutung für große Datenbanken. Damit I/O Kosten möglichst wenig von Offline Backup beeinflusst sind, bietet RMAN „Duration“ Parameter:

```
RMAN> backup duration 0:45 database;
```

In diesem Fall wird Backup nach 45 Minuten automatisch abgebrochen. RMAN hat zwei wichtige Kriterien für Backup, mit denen möglich ist, Einfluss von Backup auf Datenbank zu beschränken:

- das Dauer von Backup

Mit diesem Parameter lässt sich Dauer von Backup minimieren. RMAN ist standardmäßig mit dieser Option eingestellt. Ebenfalls, mit diesem Parameter wählt RMAN für Backup zuerst die längst nicht gesicherte Datafiles. Im Fall von Absturz des Backups kann diese Strategie oft helfen.

```
RMAN> backup duration 00:45 minimize time database;
```

- die I/O Belastung von Backup

Mit diesem Parameter wird I/O Belastung möglichst niedrig in gegebener Dauer gehalten werden.

```
RMAN> backup duration 00:45 minimize load database;
```

Beschränkung der Größe von Backup Set

Neben der Beschränkung von einzelnen Backup Pieces, ist es auch möglich gesamte Backup Set zu beschränken. Wichtiger Hinweis ist, wenn ein Datafile größer als eingestellte Backup Set Größe ist, wird Backup Prozess abgebrochen!

```
RMAN> backup duration 00:45 minimize load database;
```

Überprüfung der Datenbank

Viele DBAs finden Möglichkeit für Überprüfung von der Datenbank auf logische und physische Fehler, ohne Backup wirklich durchzuführen, als eine sehr schnelle und praktische Weise Datenbank zu testen. Diese Feature ist mit folgendem Kommando möglich:

```
RMAN> backup validate database;
```

Überspringen von bestimmten Datafiles

Es wird nicht immer der Fall sein, dass sich alle Datafiles im ONLINE Zustand befinden. RMAN enthält Kommando skip, über welche ist möglich, verschiedene Datafiles aus Backup auszuschließen. Mit Parameter inaccessible werden alle Datafiles, die sich nicht auf der Festplatte befinden, übersprungen. Andere Parameter sind offline, damit alle nicht zugreifbare Datafiles ausgeschlossen sind. Nur lesbare Datafiles können mit skip readonly ausgeschlossen werden.

```
RMAN> backup database skip inaccessible;
```

```
RMAN> backup database skip offline;
```

```
RMAN> backup database skip readonly;
```

Set Kommando

Im Kapitel über Konfiguration von RMAN ist detailliert Kommando configure behandelt. Einziger Unterschied zwischen configure und set ist, dass Einstellungen mit set nur für eine Session wirksam sind. Es gibt bestimmte Parameter von set Kommando, die nur innerhalb von run Block startbar sind. Deswegen ist gute Praxis set Kommando in run Block zu schreiben. Meist benutzte Parameter sind:

- set newname

Wenn man eine Duplikation von Datenbank oder Recovery im bestimmten Zeitpunkt auf andere Datei System durchführt, kann man mit set newname Datafiles umbenennen. Dieses Kommando wird oft zusammen mit switch verwendet.

- set maxcorrupt for datafile
Anzahl von erlaubten fehlerhaften Datablocks für einzige Datafile wird mit diesen Parametern spezifiziert. Sobald diese Zahl überschritten ist, wird Backup Prozess abgebrochen.
- set until
Mit until Parameter kann man Zeitpunkts Nummer oder log Sequenz Nummer für sogenannte „point in Time“ Recovery einsetzen.
- set backup copies
Dieses Parameter definiert Anzahl der Kopien von jeden Backup Pieces innerhalb Backup Set.
- set echo
Nach diesem Kommando wird jede RMAN Kommando in „Message Log“ geschrieben werden.

Online Backup

Immer mehr und mehr DBAs entschieden sich nur Online Backups zu verwenden.

Viele Business Prozesse müssen eine ständige Konnektivität mit der Datenbank haben. Der Einsatz mit mehreren parallelen Datenbanken erhöht nur Komplexität vom gesamten Prozess und hat sich als ineffizient gezeigt. Obwohl online Backups viel Komplexität mit sich bringen, sehen das die Endbenutzer kaum. Sogar in vielen Fällen sind Online Backups noch leichter zu erledigen.

Da Datenbank nicht ausgeschaltet und eingeschaltet werden muss, kann man Online Backup mit einzeiligem Kommando starten:

```
RMAN>backup database plus archivelog;
```

Erster Schritt von Online Backup ist sogenannte log switch (über system archivelog current). Weiteres werden von Archived redo Logs und aktuelle Database Datafiles gesichert. Dann folgt zweite log switch und RMAN sichert verbleibende Archived redo Logs (mittels backup archivelog all). Am Ende werden Control file und SPFILE gesichert.

RMAN ermöglicht auch Sicherung von einzelnen Tablespaces, um die Zeit und das Speicher auf Medium zu sparen:

```
RMAN> backup tablespace example plus archivelog;
```

Da Tablespace eine logische Einheit von Oracle ist, wird RMAN einfach nach Liste mit diesen Tablespace verbundenen Datafiles von Datenbank anfordern und nur diese in Backup inkludieren. Sei es noch erwähnt werden, dass alle Einstellungen über configure Kommando gelten auch für Online Backup.

Ebenfalls, RMAN mit dem Parameter datafile unterstützt Backup von Datenbank Datafiles als physische Einheit der Datenbank. Diese Option kann nützlich sein, um Datafiles vor Kopieren an anderem Medium zu sichern.

```
RMAN>backup datafile 'E:\oracle\product\10.2.0\oradata\orcl\EXAMPLE01.DBF' plus archivelog;
```

Die Archive Logs können auch ohne Datenbank gesichert werden. Hier kann man auch ein Vielfalt an Parameter von RMAN sehen:

```
RMAN>backup archivelog all;  
RMAN>backup archivelog from time 'sysdate-1';  
RMAN>backup archivelog from sequence 353;
```

Damit alle Archive Logs nach der Sicherung gelöscht werden, reicht es Parameter delete input einzufügen. Nach der Ausführung folgendes Kommandos, werden alle Archive Logs älter als zwei Tagen gelöscht:

```
RMAN>backup archivelog until time 'sysdate-2' delete all input;
```

Wenn man die Konfiguration von der Datenbank stark verändern soll, bringt das ein weiteres Risiko für jede DBA. RMAN kommt in der Hilfe, und bietet wie bei Archive Logs eine praktische Möglichkeit nur aktuelle Server Parameter Dateien und aktuelles Control File zu sichern.

```
RMAN>backup spfile;  
RMAN>backup current controlfile;
```

Standardmäßig ist RMAN so konfiguriert, dass Backups auf lokaler Festplatte gespeichert werden. Wenn DBA irgendwann später diese Backups auf Bände kopieren will, kann das wegen der Größe der Bände problematisch sein. Eine Lösung ist, diese Dateien mit externen Programmen aufzuteilen. Dann ist die Recoveryerschwert, da auch externes Programm für das Verschmelzen (Merge) nötig ist. Viele DBAs haben folgende Strategie:

- 1) Sichern Datenbank auf der Festplatte
- 2) Sichern Backups von letzten x Tagen auf Bände
- 3) Löschen alle Backups von der Festplatte, welche älter als y Tagen sind

Diese Strategie sieht folgend in RMAN (x und y sind Anzahl der Tage) aus:

```
RMAN>backup database;  
RMAN>backup backupset completed before ,sysdate-x' delete input;  
RMAN>backup backupset completed after ,sysdate-y' delete input;
```

Kopien von Datenbanken mittels RMAN erstellen

Am Anfang von neunziger Jahren war es in kleineren Unternehmen verbreitet, statt Backup von Datenbank zu machen, einfach Datenbank auf andere Festplatte zu kopieren. RMAN hilft auch in diesem Fall, obwohl dieses Verfahren inzwischen wegen mehreren Gründen veraltet ist:

- Kopieren kann man nur auf eine Festplatte (Bände sind nicht erlaubt)
- Datenbank muss geöffnet werden
- Keine inkrementelle Backups

Die Kopien können wie Backup Sets auf logische und physische Fehler überprüft werden, und erst seit dieser Version ist möglich Kopieren von einzelnen Tablespaces und Datafiles. Das Control File und Archive Logs können ebenfalls kopiert werden. Mächtiges Kommando backup wird wieder benutzt werden:

```
RMAN>backup as copy database;  
RMAN>backup as copy tablespace users;
```

```
RMAN>backup as copy datafile 1;  
RMAN>backup as copy current controlfile;  
RMAN>backup as copy archivelog all;
```

Inkrementell RMAN Backups

Das Hauptmerkmal von inkrementell Backups ist Sicherung nur von diesen Data Blocken, die sie sich seit dem letzten Backup geändert haben. Auf diese Weise brauchen Backups weniger Speicherplatz auf der Festplatte oder dem Band, Netzwerk bandwidth ist weniger belastet und Backups können signifikant schneller durchführen. Wie es in der Datenbankwelt üblich ist, müssen Ersparnisse auf einem Ort die Kosten auf den anderen verursachen. Im Fall von inkrementell Backups, das ist RecoveryVerfahren, welches jetzt viel länger dauern wird, da für einzelne Data Blöcke RMAN in mehrere Backup Sets gesucht werden muss.

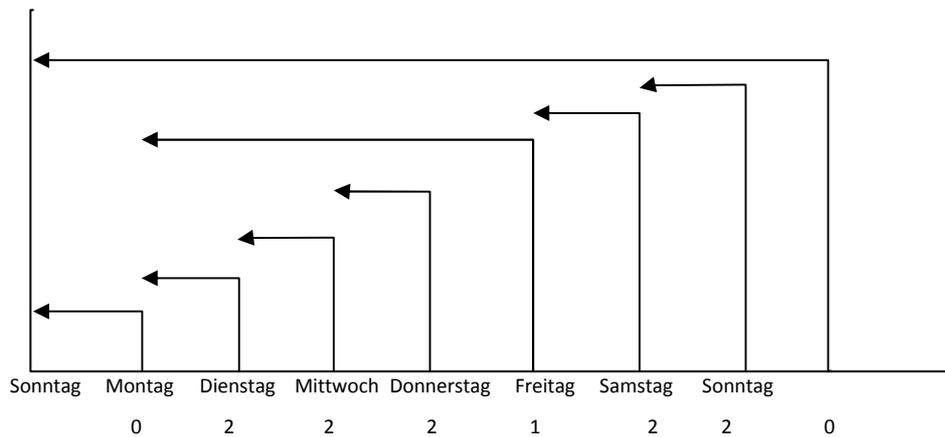
Inkrementelle Backups können in online und offline Modus, mit und ohne Archive Logs durchgeführt werden. Damit nur geänderte „data blocks“ gesichert werden, muss Datenbank ein Log führen, in welche alle Änderungen geschrieben werden. Die Größe von dieser Datei ist durch Größe der Datenbank und Anzahl von redo log Threads bestimmt und wächst in Schritten je 10 Megabytes. Es werden Informationen über Änderungen auf der Datenbank in letzten 8 Tage gespeichert. „Block change tracking“ ist in RMAN mit folgendem Kommando einzuschalten:

```
SQL>alter database enable blockchange tracking using file  
'c:\oracle\product\tracking.fil';
```

Der Grund von jedem Inkrementell Backup ist basische vollständige Backup, auf welche RMAN alle zukünftige Backups aufbauen wird. Alle Backups sind mit einem inkrementell Level Parameter versehen, und basische vollständige Backups sind mit 0 Level bezeichnet. Wenn kein basischer Backup vorhanden ist, und Benutzer trotzdem inkrementell Backup durchführen will, wird RMAN automatisch basischen Backup erzeugen:

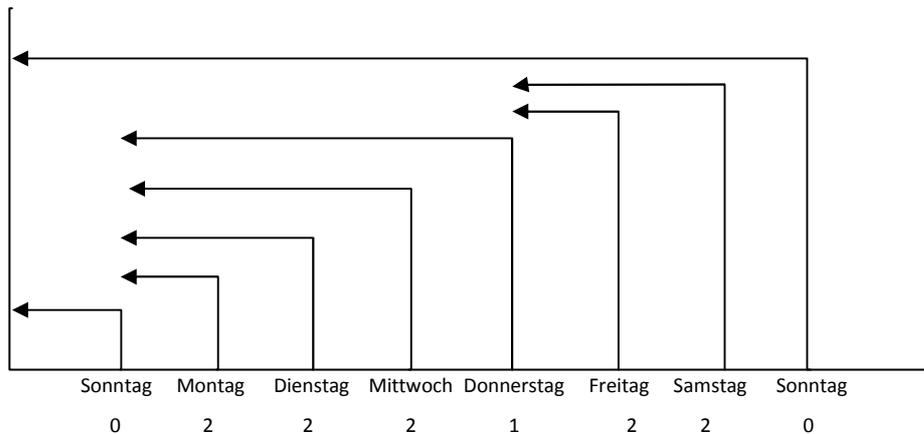
```
RMAN> backup incremental level=0 database;
```

DBAs können zwischen zwei Arten von inkrementell Backups entscheiden und zwar zwischen differential und kumulative. Differential ist standardmäßiger Typ, und ist als Backup von allen geänderten Data Block seit letztem Backup, die niedrige inkrementell Levels haben, ein bisschen verwirrend definiert. Ein Beispiel ist auf dem Bild gegeben:



Sonntags werden Backups auf dem Level 0 durchgeführt. Am Donnerstag werden Backups vom Level 1 durchgeführt und am jeden anderen Tag vom Level 2. Das Backup am Sonntag erfasst ganze Woche, und Backup am Donnerstag erfasst alle Backups zwischen Montag bis Mittwoch. Die Backups mit Level 2 werden nur Daten Blocks sichern, welche sich in diesem Tag geändert haben.

Bei Kumulativen Backups einziger Unterschied zu Differential ist Inklusion von geänderten „data blocks“ mit gleichem Level in Backup. Das erhöht insgesamt Dauer von Backup, aber erleichtert wesentlich Wiederherstellung, da nur basische und letzte kumulative Backup nötig sind.



Wenn man Datenbank beispielsweise am Mittwoch wiederherstellen muss, genügt basische Backup von Sonntag und kumulativer Backup von Dienstag zu benutzen. Parameter cumulative wird bei backup Kommando benutzt:

```

RMAN> backup incremental level=2 cumulative database;

```

Es ist, wie mit RMAN schon gewöhnt, möglich einzelne Tablespaces und Datafiles als Teil des inkrementellen Backup zu spezifizieren. Archive Logs, Control Files und backup Sets können nicht inkrementell gesichert werden.

```

RMAN> backup incremental level=2 tablespace example;

```

```

RMAN> backup incremental level=1 datafile 1;

```

Recovery

Das Backup ist nur der erste Teil einer Backup Strategie. Über Ergebnisse und Qualität von Backup Strategie kann nach einem erfolgreichen Wiederherstellungsprozess diskutiert werden. Erst dann kann man alle Vorteile und Nachteile von einer Backup Lösung herausfinden. Selbst Recovery Verfahren muss sehr sorgfältig geplant werden, da hier RMAN weniger als bei Backup hilft. In [6] ist auf Wichtigkeit von Planung und menschlicher Einsatz hingewiesen:

“Additionally, recovery from a disaster must be performed with human supervision, either at the operational level (which can be automated) or at the managerial level (which cannot be automated), and a chain of commands must be established. A disaster recovery team must be formed to devise a recovery plan. The team first studies the data processing center, including hardware and software configuration and system environment, input and output data, classification of critical and noncritical jobs, database backup procedure, and existing disaster recovery plans, if any. Finally, the recovery plan is tested. The details of the recovery plan depend heavily on the supports of the system.”

Das Betrachten von allen möglichen Optionen für Recovery von nur einer Datenbank ist sehr schwer durchzuführen. Selbst müssen DBAs über viele Kleinigkeiten aufpassen, wie zB.:

- Das Betriebssystem ist für Oracle richtig konfiguriert (zB. Oracle Umgebungsvariablen, Firewall, Rollen von Benutzern)
- Die Festplatten sind genug groß und formatiert.
- Das Netzwerk ist funktionell.
- Oracle RDBMS Software ist installiert und eingestellt. (Enterprise Manager, SQLPLUS, RMAN)
- Konfiguration Dateien sind eingestellt. (sqlnet.ora, listener.ora usw.)

Am Anfang muss man darauf Aufmerksamkeit geben, dass Recovery und Restore von RMAN zwei völlig getrennte Tätigkeiten sind. Unter Restore versteht man das Herausnehmen eines Objektes aus Backupset und Speicherung von diesem auf die Festplatte. Unter Recovery versteht man die Recovery von der gesamten Datenbank durch Archive und Redo Logs, und das Bringen von der Datenbank in konsistentem Zustand für genau einen bestimmten Zeitpunkt. Der konsistente Zustand wird durch fehlerfreien Start von der Datenbank bestimmt werden. Das Oracle hält Konsistenz als wichtigste Eigenschaft der Datenbank und erlaubt keine Öffnung von der Datenbank, wenn diese nicht konsistent ist. Die Inkonsistenz entsteht in meisten Fällen durch mehrere Ursachen:

- der Energieverlust,
- schwere Hardware-Ausfälle,
- der Netzwerk Ausfall,
- das Ausschalten der Datenbank mit „shutdown abort“,

Oracle hat mehrere Mechanismen für das Finden von Inkonsistenz. Wichtigste Überprüfung erfolgt nach mount der Datenbank, und genauer, es wird SCN (System Change Nummer) aus Control File mit SCN aus Header von Datafile verglichen werden. Sobald diese Nummer nicht übereinstimmen wird der Start von der Datenbank abgebrochen und die Fehler über Inkonsistenz Datafiles gezeigt werden.

Wenn Inkonsistenz gefunden ist, versucht sich Datenbank selbst zuerst mittels online redo logs in konsistentem Zustand wiederherzustellen, und wenn das nicht möglich ist, bietet ein crash recovery mittels Archive Logs oder instance Recovery, wenn es um Real Application Clusters geht.

Transactions in Oracle sind nach ACID Regel gestaltet, aber oft DBAs erstellen sogenannte „Bad Transactions“. Recovery der Datenbank ist sicherste Lösung von diesem Problem. Interne fortgeschrittene Algorithmen für Recovery sind in [\[5\]](#) diskutiert.

In diesem Kapitel werden zuerst Methoden für Recovery und Restore von SPFILE und Control File behandelt. In Folge wird über Recovery der Datafiles von den Datenbanken in NOARCHIVELOG und ARCHIVELOG Modus diskutiert.

Restoring SPFILE

RMAN ist standardmäßig so konfiguriert, dass SPFILE automatisch in \$ORACLE_HOME/dbs gesichert wird. Erste Schritte bei Recovery von SPFILE sind Einstellung des Namen von der Datenbank Instanz (über ORACLE_SID Variable) und Einloggen in RMAN. Zweiter Schritt ist die Eintragung von DBID und die Ausführung von Kommando restore spfile from autobackup. Dann muss noch Restart von der Datenbank gemacht werden, um Einstellungen aus wiederhergestellter SPFILE neu zu laden. Über den Parameter maxsey und maydays kann Zeitraum von Control file beschränkt werden.

```
set oracle_sid=orcl;
RMAN target sys/password
set DBID = 2539725638;
startup nomount;
run{
set controlfile autobackup format for device type disk to
'd:\backup\%F';
allocate channel c1 device type disk;
restore spfile form autobackup maxsey 200 maydays 100 to
'c:\oracle\product\db_2\spfile.backup';
}
shutdown immediate;
```

In diesem Beispiel ist von einer nicht standardmäßigen Lokation, die über channel definiert ist, zu einer selbstgewählten Lokation SPFILE wiederhergestellt.

Wenn man Flash Recovery Area oder Recovery Catalog für die Sicherung verwendet wird, ist die Recovery noch weiter erleichtert, da man DBID nicht eintragen muss. Es genügt Kommando restore spfile from autobackup auszuführen und neu starten von der Datenbank.

Restoring Control file

Viele Unterschiede zwischen Recovery von SPFILE und Control File gibt es nicht. Es sollen einfach Schlüsselwörter „spfile“ und „controlfile“ umtauschen. Parameter maxsey und maxdays funktionieren auf dem gleichen Prinzip.

```
set oracle_sid=orcl;
RMAN target sys/password
set DBID = 2539725638;
startup nomount;
restore spfile from autobackup maxsey 200 maydays 100;
shutdown immediate;
```

Oft möchten DBAs Control File bis zu einem Zeitpunkt wiederherzustellen, was durch den Parameter until ermöglicht ist.

```
RMAN>restore controlfilefrom autobackup until time
„to_date(,01/03/2008 09:00:00’, ‘MM/DD/YYYY HH24:MI:SS’)“;
```

Um alle erreichbare Backups zu finden, bietet RMAN Kommando list:

```
RMAN> list backup of controlfile;
```

In Versionen von Oracle vor 10g inkludiert Recovery von Control File, wenn diese nicht mit autobackup gesichert ist, das Schreiben von mehreren Zeilen von PL/SQL. Oracle 10g bringt als sehr nützliche Neuheit Parameter from. damit DBA ein bestimmte Backupset wählen kann:

```
RMAN> restore controlfile from ,d:\backup\C-2539725638-00’;
```

Wenn die Recovery von Control File erfolgreich durchgeführt ist, bleibt noch Datenbank zu öffnen und möglich beschädigte RMAN Kataloge zu wiedererzeugen:

```
RMAN> alter database mount;
RMAN> recover database;
RMAN> alter databse open resetlogs;
RMAN> catalog recovery area;
```

Restore und Recovery in NOARCHIVELOG Modus

Wenn sich die Datenbank im NOARCHIVELOG Modus befindet, dann ist Recovery nur von komplettem offline Backup möglich. Ebenfalls, keine Wiederherstellungen zu einem bestimmten Zeitpunkt oder von nur bestimmten Tablespace sind erlaubt. Ein guter Rat ist das Löschen von alten Datafiles, online redo logs und Control File. Nachdem Recovery von Control File sind folgende Kommandos nötig für ein Recovery:

```
RMAN>startup mount;
RMAN>restore database;
RMAN>recover database nored;
RMAN>alter database open resetlogs;
```

Kommando restore database liest aus den Control Files, welche Data Files zu der Datenbank gehören und nur diese wiederherstellt. Nächste Kommando vorbereitet Datenbank für das Öffnen. Da keine Archive Logs und online redo Logs vorhanden sind, ist Parameter nored notwendig. Schließlich wird Datenbank geöffnet und online redo Logs werden wiedererzeugt.

Wenn man eine andere als letzte Backup Set für Recovery verwendet wird, kann man Kommando set time anwenden.

```

startup nomount;
run{
SET UNTIL TIME „to_date(,01/03/2008 09:00:00‘, ‘MM/DD/YYYY
HH24:MI:SS‘)“;
restore database;
recover database noredo;
alter database open resetlogs;
}

```

Nicht alle Wiederherstellungsprozesse können so leicht, wie im vorigen Beispiel gezeigt, durchgeführt werden. RMAN speichert immer Datafiles zu dem Ort, woher die gesichert sind. Nach schweren Fehlern in der Datenbank ist besser diese Datafiles auf andere Lokation wiederherzustellen. RMAN bietet Lösung von diesem Problem mit Kommandos `set newname unspod switch`. Mit `set newname` kann man dem Datafile den neuen Namen geben, und mit `switch` wird die Datenbank Datafile von dieser Lokation verwendet.

```

run
{
set newname for datafile , c:\oracle\db1.dbf' to
,c:\oracle\recovery\db1recover.dbf';
restore database;
recover database noredo;
alter database open resetlogs;
switch datafile all;
}

```

Restore und Recovery in ARCHIVELOG Modus

Es bringt folgende Vorteile, wenn Datenbank in ARCHIVELOG Modus ist:

- Das Wiederherstellen im bestimmten Zeitpunkt
- Minimale Zeit für das Wiederherstellen
- Online Datenbank Backups
- Das Wiederherstellen von bestimmten Datafiles, während die Datenbank benutzt wird

Falls Control File und online redo logs vorhanden sind, kann man komplettes Wiederherstellen der Datenbank ausführen:

```

shutdown;
startup mount;
restore database;
recover database;
alter database open;

```

Auf paar wichtige Punkte müssen DBAs achten, obwohl dieses Verfahren sehr einfach aussieht. RMAN überprüft ob wiederherstellendes Datafile bereits existiert, und wenn ja vergleicht dieses Datafile mit diesem aus Backup Set, wenn die beide übereinstimmen überspringt RMAN das Wiederherstellen dieses Datafile. Im Fall des Scheiterns des komplettes Wiederherstellens der Datenbank werden wegen der Konsistenz alle bereits wiederherstellte Datefiles gelöscht.

Oft ist nur Datenfile von einem bestimmten Tablespace verloren, und dann muss nicht das komplette Wiederherstellen der Datenbank durchgeführt werden sondern nur von dieser Tablespace. Da Datenbank in ARCHIVELOG Modus ist, können andere Tablespaces benutzt werden. Es ist nicht möglich ein Tablespace in anderem Zeitpunkt im Vergleich zu dem Rest der Datenbank wiederherzustellen. Mit Kommando sql kann man aus RMAN zu SQLPLUS Kommandos zugreifen:

```
sql alter „tablespace users offline“;
restore tablespace users;
recover tablespace users;
sql „alter tablespace users online“
```

RMAN findet automatisch, wenn es um inkrementell Backup geht, und fordert keine weitere Aktion von DBAs in diesem Fall. RMAN versucht immer meist aktuelle basische Backup zu verwenden, was zu einer großen Zeitersparnis führt, da nur letzte online redo logs an Datenbank angewendet werden.

Wiederherstellen von Tablespace in ein bestimmte Zeitpunkt

RMAN in der Version 10g hat das Wiederherstellen von Tablespace zu dem bestimmten Zeitpunkt im Vergleich mit der Version 9 wesentlicher vereinfacht. Wieso ist das eine wichtige Feature Einfach, viele Benutzer löschen falsche Tabellen oder führen „truncate“ auf falsche Objekte in der Datenbank. Die Programmier können auch fehlerhaftes Code auf Deployment Datenbank ausführen und logische Konsistenz von der Datenbank zerstören. Bevor man sich mit dieser Art vom Wiederherstellen weiter beschäftigt, einige Termine müssen erklärt werden.

Unter dem Instanz der auxiliary Datenbank versteht man den temporären Instanz für die Durchführung des Wiederherstellens im Zeitpunkt, das später gelöscht werden kann. Auf anderer Seite stellt auxiliary Set den Rest von benötigten Dateien aus der Zieldatenbank (backup control file, system tablespace, undo segment tablespaces datafiles), um das Wiederherstellen fehlerfrei zu erledigen.

Die schwerste Aufgabe ist sorgfältig zu bestimmen, zu welchem Zeitpunkt man zurückkehren will. Wenn man keinen Recovery Katalog verwendet hat, gibt es keine zweite Chance für das Wiederherstellen des Tablespaces. Leichtere Aufgabe ist zu überprüfen, ob wiederherstellende Tablespace mit einem anderen verknüpft ist. Für diesen Zweck bietet RMAN das Sicht TS_PITR_CHECK, mit welchen man leicht Selbstständigkeit von Tablespace überprüfen kann:

```
select obj1_owner, obj1.name, obj1_type, reason
where (TS1_NAME IN ('EXAMPLE'))
AND TS2_NAME NOT IN ('EXAMPLE'))
OR (TS1_NAME NOT IN ('EXAMPLE'))
AND TS2_NAME IN ('EXAMPLE'))
```

Die Anfrage soll keine Rows selektieren, wenn keine Konflikte vorhanden sind. Nach einem Wiederherstellen von Tablespace im Zeitpunkt beispielsweise am 01.03.2008. um 15:00 beendet ist, sind alle Änderungen von diesem Tablespace nachdem diese Zeitpunkt verloren. Um man dem Zustand Datenbank nach dem Wiederherstellen zu überprüfen, enthält Oracle 10g das Sicht TS_PITR_OBJECTS_TO_BE DROPPED:

```
select * from TS_PITR_OBJECTS_TO_BE DROPPED
where tablespace_name='EXAMPLE';
```

Der Vergleich mit MySQL Backup und Recovery Lösung

Oracle Datenbank befindet sich fast 30 Jahren auf dem Markt und hat sich inzwischen als Marktführer etabliert. Wie auch immer hohe Qualität und Zuverlässigkeit muss man ebenso hoch bezahlen. Deswegen sind immer interessant aktueller Entwicklungsstand von kostenlosen Datenbanken mit der „teureren“ Oracle Datenbank zu vergleichen. Am Markt von den kostenlosen Datenbanken hat sich in letzten Jahren MySQL von der Firma MySQL AB als klarer Marktführer auskristallisiert. In diesem Kapitel werden Backup und Recovery Lösungen zwischen Oracle (das Version XE) und MySQL 5.1 verglichen werden.

MySQL verfügt über verschiedene Speicher-Engines für den Zugriff auf die Daten. Für dieses Experiment wird Default installiertes InnoDB verwendet werden. InnoDB erweitert Original Speicher-Engine mit transaktionssicheren Lese- und Schreib-Zugriffen für Tabellen und mit der Möglichkeit, Fremdschlüssel-Beziehungen zu überprüfen. InnoDB Oy, der Hersteller des Produkts, wurde im Oktober 2005 von Oracle Systems übernommen. MySQL bietet die Werkzeuge für:

- Online und offline Backups
- Vollständige und inkrementelle Backups
- Komprimierung von den Backup Sets
- Zeitpunktbezogene Wiederherstellung

Da alle diese Features von beiden Datenbanken unterstützt sind, werden sie genauer betrachtet. Für Testzwecken ist mit Oracle gelieferte Tablespace EXAMPLE in MySQL über Migration Tool eingefügt.

Backup

Mit MySQL ist keines Program für physische Backups geliefert. Das Hilfsprogramm mysqlhotcopy kann physische Backups erzeugen, aber nur für myISAM Tabellen. Für InnoDB Tabellen existiert tibbackup Skript, welches aber kostenpflichtig ist und für Windows Umgebung nicht verfügbar ist.

Hilfsprogramm mysqldump wird in den meisten Fällen mit dem MySQL verwendet werden. Dieses Programm erstellt logische Backups, dh. der Zustand der Datenbank wird über mehrere INSERT Befehlen in einer textuellen Datei gespeichert werden. Wenn man andere Speicher Engine als InnoDB benutzt, sollte man mysqlhotcopy verwendet werden. Um Datenbank EXAMPLE zu sichern, reicht es folgendes Kommando auszuführen:

```
shell>mysqldump -tab=d:\backup--opt example
```

Mysqldump wird online ausgeführt, dh. Lese- und Schreiboperationen werden nicht beeinträchtigt. Mit Parameter opt kann man Datenbank für Backup bestimmen.

Wie Oracle auch MySQL bietet Option für inkrementelle Backups über binäre Update-Logdatei. Das Server muss mit dem Parameter --log-bin gestartet werden. Nach jedem durchgeführten Inkrementellen Backup muss man dieses Binäre Log mittels Kommando FLUSH LOGS

rotieren. Schließlich müssen alle Binär Logs, die aus dem Zeitraum zwischen dem letzten vollständigen oder inkrementellen Backup und dem vorletzten erstellten Log stammen, an die Sicherungsposition kopiert werden. Binär Logs kann man als „redo logs“ aus der Oracle Welt verstehen.

```
shell>mysqldump --flush-logs --single-transaction example>
basebackup.sql
```

Parameter `--single-transaction` Parameter sperrt von `mysqldump` verwendeten Daten, damit andere Prozessen diese nicht verändern können. Für `myISAM` Tabellen gilt die Sperre nicht, und muss man manuell sicherstellen, dass diese nicht während Backup Prozesses geändert werden. (zb. Administrativen Änderungen an Benutzer Konten sind nicht erlaubt.)

Recovery

Da es um logisches Backup bei `mysqldump` geht, ist Recovery Verfahren trivial:

```
shell> mysql < basebackup.sql
```

Nach diesem Recovery befindet sich Datenbank in gleichem Zustand, wie sie im Zeitpunkt des Backups war. Um inkrementelle Backups zu wiederherstellen, benutzt man folgendes Kommando:

```
shell> mysqlbinlog gbichot2-bin.000007 gbichot2-bin.000008 | mysql
```

Mit `mysqlbinlog` werden Binär logs an Datenbank angewendet und inkrementelle Recovery ist durchgeführt. Seit der Version 4.0 ist auch möglich in MySQL Recovery in einem bestimmten Zeitpunkt zu erledigen:

```
shell>mysqlbinlog --stop-date „2008-03-30 00:00:00“ \ gbichot2-
bin.000007 | mysql
```

Nach diesem Kommando werden alle Daten aus Binär Log bis `“stop-date“` in Datenbank wiederhergestellt. Weiters ist möglich `start-date` nennen, um eine zb. falsches Löschen aus der Datenbank zu überspringen:

```
shell>mysqlbinlog --start-date „2008-03-30 00:00:05“ \ gbichot2-
bin.000007 | mysql
```

Alternativ zu der SCN Nummer in Oracle befindet sich in MySQL Binär Logs sogenannte Position Numbers, die eine feine Wiederherstellung ermöglichen, zb. wenn viele schädliche SQL Anweisungen gleichzeitig auftraten. Über Parametern `--start-position` und `--stop-position` kann man genauere Logposition für Datenbank Recovery bestimmen. Die Logpositionen der schädlichen SQL

Anweisungen kann man mit dem Kommando `mysqlbinlog` herausfinden und in einer textuelle Datei umleiten:

```
shell>mysqlbinlog --start-date„2008-03-30 00:00:00“ \--stop-  
date=„2008-03-30 00:00:05“ \gbichot2-bin.000007 > mysql_restore.sql
```

Nach der Analyse der `mysqlrestore.sql` führt man Wiederherstellung vor und nach schädlichen SQL Anweisungen aus:

```
shell>mysqlbinlog --stop-position „1000“\gbichot2-bin.000007 | mysql  
shell>mysqlbinlog --start-position „1010“\gbichot2-bin.000007| mysql
```

Beschluss

MySQL speichert einzelne Tabellen in Dateien (abhängig von Format in `.frm`, `.MYD` und `.MYI`). Das Einfachste Backup Verfahren wäre einfach das Kopieren von diesen Dateien auf andere Medium. Diese Lösung ist aber mit großen Datenbanken nicht flexibel, da bereits gesicherte Daten immer wieder kopiert werden. Anderer Nachteil ist der Bedarf an Sperren von Tabellen mit Kommando „`LOCK TABLES`“, um ein konsistentes Backup zu bekommen.

Wie in vorigem Kapitel bereits erwähnt ist, mit MySQL ist keine Lösung für physische Backups geliefert. Deswegen wird hier Diskussion meist in Richtung logisches gegen physisches Backup gehen. Als Beispiel für logisches Backup wird `mysqldump` benutzt und für physisches Backup `RMAN`. Als Testdatenbank wird mit Oracle gelieferte „HR“ Schema verwendet, die über Migration Tool in MySQL angepasst ist. Als Backup Medium wird eine externe USB Festplatte benutzt.

Beide Lösungen verwenden Kern Funktionen von unterliegender Datenbank um Meta Informationen über Struktur der Datenbank zu sammeln. In der folgenden Tabelle gibt es die Übersicht von Dauern und Größen der Backups in jeweilige Datenbank:

| | MySQL- <code>mysqldump</code> | Oracle - <code>RMAN</code> |
|--|-------------------------------|----------------------------|
| Größe der Datenbank | 88 kb (10 mb) | 77 mb |
| Größe des Backups | 24 kb | 66 mb |
| Größe des komprimierten Backups | 8 kb | 20 mb |
| Dauer von Backup | 1 Sekunde | 7 Sekunden |
| Dauer von Recovery | 3 Sekunden | 3 Sekunden |

Wegen verschiedener Weisen von Speicherung der Daten ist die Größe von Tablespaces nicht vergleichbar und ist nicht das Thema der Diskussion. InnoDB hat eine zusätzliche gemeinsame Tablespace für alle InnoDB Tabellen. Interessant ist auf jeden Fall Grad der Komprimierung, welche bei MySQL 66% und bei Oracle 70% beträgt. MySQL benutzt originale ZIP Archivier von dem Jahr

1996. Neben integrierte Datenkomprimierung in Datafiles ,die sehr oft Tablespaces um 50% reduzieren, was nur zu 5% Belastung der Prozessors bei Anfragen führt [\[8\]](#), verwendet Oracle für „compressed backupsets“ BZIP2 Algorithmus. Beide bieten den ähnlichen Grad der Kompression.

Dauer vom Backup Verfahren ist im Verhältnis mit der Größe der Datafiles und ist in Ordnung zu behaupten, dass mit größeren Datenbanken RMAN sicher schnellere Lösung ist. Der Hauptgrund ist nicht eine perfekte Implementierung von RMAN, sondern Tatsache dass mysqldump ein logisches Backup Werkzeug ist, und muss ständig Meta-Informationen vom Datenbank Server anfordern und diese in logisches Format umwandeln. Auf andere Seite RMAN verfügt über bereits für Backup vorbereitete Daten.

Beim Recovery Verfahren sieht man bereits auf dieser kleinen Datenmenge den wichtigsten Nachteil von logischen Backups. Das ist die Geschwindigkeit. Obwohl mysqldump Backup Datei nur 24 kb beträgt, dauert das Wiederherstellen wie bei RMAN Backup Datei von 66 mb. Da diese Geschwindigkeit konstant ist, kann Wiederherstellung von zb. 10 GB Datenbank mehrere Tagen dauern. Beispielsweise würde dieses Recovery mit RMAN im schlechtesten Fall paar Stunden dauern.

Eine wichtige Eigenschaft von logischen Backups ist Portabilität. Ein logisches Backup kann man leicht an verschiedene Versionen von der Zieldatenbank und verschiedener Dateisysteme wiederherstellen, was bei physische Backups oft nicht der Fall ist. Sobald Architektur von Datafiles in neuer Version von Zieldatenbank verändert ist, sind alte Backupsets nicht wiederherstellbar.

Beide Werkzeuge verfügen über Feature für Wiederherstellung in einem bestimmten Zeitpunkt. Es ist möglich entweder nach dem Datum oder nach dem Log Nummer (bei MySQL Binär Log und bei Oracle SCN Nummer) Datenbank in einem bestimmten Zustand wiederherzustellen. Keine Werkzeuge bietet bessere Feature in diesem Gebiet im Vergleich zu anderen.

Inkrementelle Backups sind sowohl von mysqldump als auch von RMAN unterstützt. RMAN hat sich als praktischer gezeigt, weil weniger Aktionen von DBA erwartet sind. Bei mysqldump muss man alle Logs zwischen zwei Backupsets manuell sichern.

Schlussfolgerung

RMAN wurde erstmals mit Oracle Version 8.0.3 mitgeliefert. Erste Versionen enthalten hin und zu kleinere Bugs, Kommandos hatten den ungewöhnlichen Namen, Parameter für Kommandos waren inkonsistent, aber mit jeder Version hat sich RMAN verbessert. Schon in der Version 9i war RMAN meist verwendetes Werkzeug für Oracle Backups.

Die wichtigsten Kommandos für Backup und Recovery haben in RMAN eine einfache und erwartete Syntax. Viele Kommandos haben einheitliche Parameters, die in den meisten Fällen selbsterklärend sind. Trotz verbreitenden Vorurteilen in unserer Zeit zu Konsole-Anwendungen ist Produktivität mit RMAN auf höchstem Niveau. Die übliche alltägliche Backup und Recovery Aufgaben sind mit dem paar einfachen Kommandos erledigt. RMAN lässt sich auch mit einigen GUI steuern wie zB. Enterprise Manager, NetBackup, EMC NetWorker, Tivoli Storage Manager, aber keine bietet so viele Komfort wie RMAN, was man besonders mit komplizierten Vorhaben bemerken kann.

Ebenfalls kann RMAN leicht erweitert werden, um spezifische Hardware Umgebung besser auszunutzen. Fujitsu Eternus Festplatten verfügen über fortgeschrittene Features für Kopieren, die durch Benutzung von Fujitsu Advanced Copy Manager zusammen mit RMAN höhere Performances leisten. [\[7\]](#)

Im Vergleich mit Skripten für Backup und Recovery hat RMAN folgende Vorteile:

- Online Backups sind mit Skripten nicht möglich
- Mit RMAN muss man nicht neue Datafiles und Tablespaces für Backup melden
- RMAN sichert nur verwendete Data Blocks
- RMAN bietet sehr gute Komprimierung
- Zentralistische Berichten über Recovery Catalog
- Möglichst niedrigste Speicherbedarf für inkrementelle Backups
- Testen von Backups ohne diese wiederherstellen zu müssen
- Integration mit third party GUI Tools
- Dokumentation

Das neue Feature „Flash Recovery Area“ erleichtert wesentlich Steuerung von Backup und Recovery und Management von verschiedenen Mediums, da alle relevante Daten zentralistisch gespeichert sind. Flash Recovery Area warnt automatisch DBAs von veralteten Backups und mangelndem Speicherplatz.

Alle Vorteile von neuen Features in RMAN 10g kann man erst in Real Life Benutzung beweisen. Das Unternehmen Starwoods Hotels, ein von den führenden Hotels und Freizeit Unternehmen in der USA, verwendet RMAN seit Version 8 und hat ein Case Study nach halbjähriger Verwendung von RMAN Version 10g veröffentlicht. Datenbank verfügt über Informationen für 230.000 Zimmer in 82 Ländern und war in der Zeit von Case Study rund 8 TB groß. Für Backup und Recovery zuständige Maschine hat 24 Prozessoren und 48 GB Arbeitsspeicher.

Das Ergebnis ist:

- Die Dauer von Backup ist von 19 Stunden auf 2 Stunden gesunken
- Das Auslassen von Voll Backups, da inkrementelle updatet Backups verwendet werden
- Weniger Prozessor Belastung , da volle Datenbank Überprüfungen wegen „Block Change Tracking“ vermeiden werden
- Höhere Return of Investment , weil low-cost Festplatten mit FRA und ASM verwendet werden

Am 11. Juli 2007 hat Oracle das neue Oracle Database Version 11g vorgestellt. RMAN wird mit weiteren neuen Features (Data Recovery Advisor, Multisection Backups, Fast Backup Compression mit ZLIB Algorithmus, Virtual Private Catalog, Integration with Windows Volume Shadow Copy Services) zu mächtigstem Werkzeug für Backup und Recovery der Oracle Datenbanken geworden.

Die gesamte Zeit für das Erlernen des RMANs ist sicher eine gute Investition. Gleich nach dem ersten Absturz der Datenbank und erfolgreiches Recovery werden das gesamte Bemühen und die Arbeit belohnt werden.

Skripten

Obwohl RMAN viele eigene Kommandos zur Verfügung stellt, können mehrere alltägliche Aufgaben über Skripten viel komfortabler gelöst werden. Es ist mehrmals in dieser Arbeit erwähnt, dass eigene Skripten für Erzeugung und Wartung von Backups oft nicht genug gut geschrieben sind und können schwere Problemen mit der Datenbank verursachen. Skripten als Hilfsmittel, um zB. verschiedene Informationen und Reports zu bekommen, sind eine praktische und weit verbreitete Lösung.

Datenbanken ohne aktuelles Backup

Alltägliche Aufgabe von DBAs ist Überprüfung, ob alle Datenbanken regelmäßig gesichert sind. Da RMAN report Kommando nur Information für eine Datenbank abholen kann, weil man an Zieldatenbank verbunden werden muss, generiert dieses Skript Bericht für alle Datenbanken. Es werden sich auf der Liste auch gelöschte Datenbanken befinden, was eine Erinnerung für DBAs ist, dass Kataloge Einträge von diesen Datenbanken erfrischt werden soll.

```
set lines 74 pages 400 verify off termout on feedback off
accept period number prompt 'Period in Days: '
```

```
define ds = "'DD.MM.YY HH24:MI'"
define date_format = "form a14 head"
define bkp_vol = "(blocks * block_size)"
```

```
col name form a8 head "Database"
col start_time &&date_format 'Backup Start'
col end_time &&date_format 'Backup End'
col speed form 990.99 head 'Speed|MB/Sec'
col duration form a8 head 'Duration'
col start_time &&date_format 'Last Backup'
```

```
SELECT    d.name,
          TO_CHAR(MAX(s.start_time), &&ds) start_time
FROM      rc_backup_set s, rc_database d
WHERE     s.db_key = d.db_key
GROUP BY d.name
HAVING    MAX(s.start_time) < SYSDATE - &&period
ORDER BY MAX(s.start_time);
```

Größe aller Datenbanken

Für Planung und Entwicklung der Backup Strategie ist obligatorisch Größe auf Festplatte von alle Datenbanken zu kennen. Wie im vorigen Beispiel ist unmöglich report Kommando zu benutzen, da eine Verbindung für jede einzelne Datenbankinstanz hergestellt werden muss. Die Sichten rc_datafile und rc_tempfile enthalten alle benötigten Informationen. Die Größe von Control Files und Redologe wird nicht beachtet. Der Bericht wird in Form von einer HTML Tabelle ausgegeben, die sich in Datei dbsizes.html befindet.

```
set markup html on spool on
spool dbsizes.html
set lines 74 pages 400 verify off termout on feedback off

define ds = "'DD.MM.YY HH24:MI'"
define date_format = "form a14 head"
define bkp_vol = "(blocks * block_size)"

col name form a8 head "Database"
col start_time &&date_format 'Backup Start'
col end_time &&date_format 'Backup End'
col speed form 990.99 head 'Speed|MB/Sec'
col duration form a8 head 'Duration'
col start_time &&date_format 'Last Backup'

col dbsize form 999G999D9 head 'Size in GB'
break on report
compute sum of dbsize on report

SELECT    f.db_name NAME,
          round(SUM(f.bytes) / &&giga,1) dbsize
FROM      (SELECT db_name,
                  dbinc_key,
                  bytes
            FROM    rc_datafile
            WHERE   (drop_time IS NULL)
            OR      (drop_time = creation_time)
            UNION ALL
            SELECT db_name,
                  dbinc_key,
                  bytes
            FROM    rc_tempfile
            WHERE   (drop_time IS NULL)
            OR      (drop_time = creation_time)) f,
          rc_database_incarnation i
WHERE     i.dbinc_key = f.dbinc_key
AND       i.current_incarnation = 'YES'
GROUP BY db_name
ORDER BY 1 ASC;
spool off;
```

Informationen über Backup

Mit Kommando list kann man viele Informationen über Backup Sets herausholen. Das Hauptproblem ist sehr schlechte Lesbarkeit von gegebenen Informationen. Es wird nicht der Name der Datenbank sondern ihre Schlüssel herausgegeben, und es fehlt Informationen über Volumen, Dauer und Durchsatz von jeder Datenbank. Es bleibt ein Eindruck, dass diese Informationen mehr für ein „Third Party Software“ als für DBAs geeignet sind. Als Parameter für dieses Skript ist den Name der Datenbank zu geben.

```
set lines 74 pages 400 verify off termout off

alter session set nls_date_format='DD.MM.YY HH24:MI';
set termout on

define db = "&&1"
define date_format = "form a14 head"
define byte_displ = "form a14 head"
col start_time &&date_format 'START TIME'
col backup_type form a3 head 'BKP|TYP'
col incremental_level form 99 head 'INC|LVL'
col status form a1 head 'S'
col ORIGINAL_INPUT_BYTES_DISPLAY &&byte_displ 'VOLUME'
col ORIGINAL_INPRATE_BYTES_DISPLAY &&byte_displ 'VOLUME|PER SEC'
col TIME_TAKEN_DISPLAY form a8 head 'DURATION'

SELECT      sd.start_time,
            DECODE(sd.backup_type, 'D', 'FUL', 'I', 'INC', 'L', 'ARC')
            backup_type,
            sd.incremental_level,
            sd.status,
            sd.original_input_bytes_display,
            sd.original_inprate_bytes_display,
            sd.time_taken_display
FROM        rc_backup_set s, rc_backup_set_details sd, rc_database d
WHERE       d.NAME = UPPER('&&db')
AND         s.db_key = d.db_key
AND         s.bs_key = sd.bs_key
AND         sd.start_time > SYSDATE - 10
ORDER BY   sd.start_time;
```

Literatur

- [1] Matthew Hart, Robert G. Freeman: RMAN Backup & Recovery. The McGraw-Hill Companies 2003, ISBN 0-07-226317-2
- [2] Philip A. Bernstein, Vassos Hadzilacos, Nathan Goodman: Concurrency Control and Recovery in Database Systems. Addison-Wesley 1987, ISBN 0-201-10715-5
- [3] Alexander Kick: Oracle Datenbankadministration mit SQL-Skripten. Carl Hanser Verlag 2006, ISBN 3-446-40727-8
- [4] Kevin Loney: Oracle Database 10g The Complete Reference. The McGraw-Hill Companies 2003, ISBN 0-07-225351-7
- [5] David Lomet, Zougrafoula Vagena, Roger Barga: Recovery from "Bad" User Transactions. SIGMOD 2006
- [6] Manhoi Choy, Hong Va Leong and Man Hon Wong: Disaster Recovery Techniques for Database Systems. November 2000 Communications of the ACM, Volume 43 Issue 11es
- [7] Isao Igarashi: High-Speed Backups without Stopping Business Applications. Fujitsu Sci.Tech. Journal, 42, 1, January 2006
- [8] Meikel Poes, Dmitry Potapov: Data Compression in Oracle. VLDB 2003. [<http://www.vldb.org/conf/2003/papers/S28P01.pdf>]
- [9] Donna Cooksey, Arup Nanda: Starwood Hotels: Case Study. Oracle 2004 [http://www.oracle.com/technology/Deploy/availability/pdf/Starwood_RMAN_CaseStudy.pdf]
- [10] W. Curtis Preston: Backup and Recovery. O'Reilly Media 2007, ISBN 0-596-10246-1
- [11] David B. Little, David A. Chapa: Implementing Backup and Recovery: The Readiness Guide for the Enterprise. John Wiley 2003, ISBN 0-471-22714-5
- [12] MySQL 5.1 Reference Manual. MySQL AB 2008. [<http://downloads.mysql.com/docs/refman-5.1-en.a4.pdf>]