

# Migration einer terrabytegroßen Datenbank bei minimaler Downtime

André Schulte  
Freiberufler  
Köln

## Schlüsselworte:

Oracle Datenbank, Migration, rman, statspack.

## Einleitung

Bei diesem Erfahrungsbericht wird erörtert, wie man eine Datenbank im terrabyte Bereich von einer DB Maschine mit SAN Storage auf eine andere Maschine mit gleichzeitigem Austausch des Storage Subsystems bringen kann. Bei dem Projekt aus der Telekommunikationsbranche gab es bedingt durch den Online Betrieb die Anforderung einer minimalen Downtime.

## Motivation

Sieht ein Geschäftsmodell stetes Wachstum für einen Service vor, muss das zugehörige IT System und damit auch die Datenbank skalierbar sein. Beim Start oder Launch eines Service ist oft unklar, wie er angenommen wird und wie stark sein Wachstum sein wird. Ein System ist skalierbar, wenn der Durchsatz der verarbeitbaren Anfragen steigt, wenn Hardware-Ressourcen hinzugefügt werden. Multiprozessor Systeme können in zwei Skalierungsgruppen unterteilt werden. Die horizontale Skalierung, auch Scale-up genannt fügt einer Symmetric Multiprocessing Maschine (SMP) weitere Hardware hinzu. Bei dieser Art der Hardwarearchitektur sind zwei oder mehr identische CPUs mit einem gemeinsamen Speicher verbunden und werden von einer Betriebssystem Instanz gesteuert. Im Gegensatz dazu fügt die vertikale Skalierung, auch Scale-out genannt einer Gruppe von kleineren über einen Interconnect verbundenen Maschinen weitere hinzu. Ein klassische Migration findet statt, wenn ein Service ein paar Jahre alt ist und man feststellt, dass die Datenbank Maschinen und das Storage Subsystem zu alt sind. Bei einer Oracle Database Single-Instanz mit horizontaler Skalierung macht es keinen Sinn weiteren RAM hinzuzufügen, wenn die Prozessor Kernel Architektur und Bus Technologie veraltet sind. Beim RAC können die Nodes ohne Ausfallzeit getauscht werden, aber auch beim RAC hat man wie in der Single Instanz nur eine einzelne Datenbank. Ist das Storage System veraltet oder der Service so schnell gewachsen, dass das bestehende die Last nicht mehr bewältigen kann, wird in der Regel entschieden es zu wechseln. Aber auch ein Storage Subsystem, dass schnell genug ist, aber aufgrund seines Alters nicht mehr vom Hersteller oder Rechenzentrum supportet wird, kann ein Grund für einen Austausch sein.

Ein für die Migration wichtiger Faktor ist die Größe der Datenbank. Je größer die Datenbank, desto träger wird sie in Bezug auf ihre Beweglichkeit. Während Datenbanken mit einer Größe im terrabyte Bereich in der Vergangenheit selten gewesen sind, trifft man sie heute immer öfter an. Durch Entwicklungen im Hardwarebereich und durch das Web 2.0 gibt es immer mehr Verarbeitungen von Massendaten. Firmen speichern in den Fachverfahren mehr Daten, wobei in der Regel multimediale Inhalte schnell zu einer großen Datenbank führen. Mit dem Ansteigen des Datenwachstums ist auch in der Zukunft zu rechnen. So prognostiziert die EMC Corporation eine Steigerung der jährlich entstehenden Informationen bis 2020 um den Faktor 44. Insgesamt nimmt nicht nur die Häufigkeit, sondern auch die Schwierigkeit von Migrationen zu.

## **Ausgangssituation**

Bei den bevorstehenden Migrationen aus der Telekommunikations-Branche gab es bedingt durch den Online Betrieb die Anforderung einer minimalen Downtime. Es gibt verschiedene Datenbanken für verschiedene Mandanten. Jede Datenbank läuft in einer aktiv/passiv Cluster Konfiguration unter Solaris 10. Jede Datenbank ist zwischen einem und drei Terrabyte groß, die dahinterliegende Hardware ist mehrere Jahre alt und soll getauscht werden. Dazu kommt eine gewachsene Kundenbasis, wodurch die Last auf den Maschinen größer geworden ist und die Antwortzeiten sich verschlechtert haben. Die Daten mussten außerdem in einen räumlich getrennten Teil des Rechenzentrums, der mit moderneren Techniken ausgestattet ist wechseln. Bei dem Service handelt es sich um ein Onlineportal, bei dem die Endkunden davon ausgehen, dass er ständig verfügbar ist. Es ergab sich die Problematik mit einer nur kurzen Ausfallzeit die Daten auf die neue Hardware zu migrieren, wobei jede Datenbank eine eigene Migration darstellt. Dazu kam erschwerend, dass die Datenbanken keine Partitionierung hatten.

## **Projektumfeld**

Das Projektumfeld bestand aus dem Rechenzentrum, das für die Hardware, das Betriebssystem und die Clusterware zuständig gewesen ist. Des Weiteren gab es den Betreiber, der den Managed Service angeboten hat sowie den Kunden aus der Telekommunikations-Branche unter dessen Name der Managed Service läuft. Das Projektmanagement lag bei dem Anbieter des Managed Service, der für die Durchführung und Koordinierung verantwortlich gewesen ist.

## **Planung der Migration**

Ein Runterfahren des Service, Kopieren der Daten und Starten des Service kam aufgrund der durch die Kopierzeit verursachte Downtime nicht in Frage. Das Transportieren des SAN Devices in das neue Rechenzentrum auch nicht. So das zwei Möglichkeiten übrigblieben Migration von Kundenbereichen als Form einer applikationsgesteuerten Migration oder eine Migration mit Hilfe von rman. Bei der applikationsgesteuerten Migration sperrt man einen Kundenbereich überträgt ihn und gibt ihn wieder frei. Der Nachteil liegt darin, dass die Applikation wissen muss, wo der aktuelle Kunde liegt. Man muss aus mehreren Tabellen die Daten übertragen. Der Weg ist nicht unmöglich. Es besteht aber die Gefahr von hohen Aufwänden und das Risiko eines korrupten Datenbestandes durch einen unvorhergesehen Fehler. Der gewählte Lösungsweg war ein rman Restore eines Hot Backups auf dem Zielsystem mit periodischen Nachfahren der Archivelogs bis zum Switch der Systeme. Es ist also der Aufbau eines klassischen selbstverwalteten Standby Systems. Da verschiedene Parteien wie Rechenzentrum, Testabteilung, Projektmanagement und Endkunden beteiligt sind, ist eine belastbare zeitliche Terminierung notwendig. Deshalb ist es Teil der Planung zu berechnen, ab wann man den Switch vornehmen kann. Dazu summiert der für die Migration benötigten Schritte:

1. Die Zeit eines rman Hot Backups der Produktion.
2. Die Zeit für das Übertragen der Daten in das Zielsystem. Dazu sollte man im Vorfeld Übertragungstests durchführen. Das Übertragen kann parallel gestartet werden, wenn die ersten Backupsets fertig sind.
3. Die Zeit des Disaster Recovery kann man aus der Zeit des Hot Backups herleiten. Es wird in der Regel schneller sein, da keine Produktionslast parallel vorhanden ist und das Zielsystem mit besserer Hardware ausgestattet ist. Man kann ca. 20 Prozent von der Hot Backup Zeit abziehen.
4. Von der Zeit des Hot Backups bis zur Fertigstellung des Disaster Recovery müssen alle Archivelogs gesammelt und übertragen werden. Da dies parallel zum Disaster Recovery

erfolgen kann, aber auch iterativ wiederholt werden muss, sollte man mindestens die Zeit für das Übertragen von einer Stunde Archivelogs einplanen.

5. Zeit für das Nachfahren der Archivelogs für die Schritte zwei bis fünf. Das Nachfahren geht schneller als das Erstellen der Archivelogs. Entweder hat man Erfahrungen von einem Disaster Recovery Test oder man schätzt die Zeit mit ca. 50 Prozent der Zeit von der Logerstellung ein. Dies ist aber Abhängig vom Service. Sowie Zeit für das Kopieren und Nachfahren iterativer Archivelogs.

Mit der Zeitabschätzung legt man den Termin verbindlich fest. Ein Switch der Systeme kann nur erfolgen, wenn alle Parteien zu dem Zeitpunkt der Migration nach ihren Tests zustimmen. Beim Switch wird die alte Datenbank geschlossen, die neue lesend und schreibend geöffnet und die Verbindungen der Applikationsserver auf die neue Datenbank konfiguriert.

## Durchführung

Hot Backup mit rman / backup Archivelogs (Nur Grob, von DB und rman Konfiguration abhängig):

```
run{
  backup as backupset database
  plus archivelog format '/global/backupdatabase/AL_%d_%t_%s_%p' delete input;
}

run{
  backup as backupset archivelog all format '/global/ backupdatabase/AL_%d_%t_%s_%p' delete input;
}
```

Auf dem Zielsystem müssen alle Verzeichnisse wie auf dem Quellsystem angelegt werden und ORACLE\_HOME bzw. ORACLE\_SID gesetzt sein. Das Disaster Recovery beginnt wie folgt.

```
RMAN> connect target

connected to target database (not started)

RMAN> startup nomount

startup failed: ORA-01078: failure in processing system parameters
LRM-00109: could not open parameter file '/opt/oracle/product/10.2/dbs/initMYDB.ora'

trying to start the Oracle instance without parameter files ...
Oracle instance started
...

RMAN> restore spfile from '/opt/oracle/backup/controlfile_c-4058565783-20070216-00';

RMAN> restore controlfile from '/opt/oracle/backup/controlfile_c-4058565783-20070216-00';

RMAN> startup force mount (force impliziert shutdown abort, deshalb nicht auf Produktionen verwenden!)

Starten von Restore und Recovery mid dem Controlfile aus dem Backup.
run{
  restore database;
  recover database;
}
```

nicht startet aufgrund eines shared memory Fehlers, erstellt man ein init.ora, File, indem man die Speicherparameter (mit max\_shared\_servers = 200, processes = 300, large\_pool\_size = 384M, shared\_pool\_size = 256M ) angibt. Dies kann bei Mehrkernprozessoren passieren. Man startet die Instanz dann mit dem neuen init.ora File:

```
startup mount pfile='/local/oracle/app/oracle/product/10.2.0/db_1/dbs/init.ora'
```

Neu übertragene ArchiveLogs werden mit folgendem Befehl dem rman Katalog hinzugefügt. Bsp.:

```
catalog backuppiece '/backup/MYSID/01dmsbj4_1_1.bcp';
```

Danach werden die ArchiveLogs mit dem rman Befehl „recover database“ automatisch genutzt. Sind alle ArchiveLogs angewendet kann man die Datenbank nur lesend öffnen und kontrollieren, danach muss sie für die neue Produktion mit Resetlogs mit einer neuen Inkarnation geöffnet werden.

### **Nacharbeit**

Dazu zählen das Umkonfigurieren der Applikationsserver Verbindungen, Durchsicht der Logfiles, fachliche Tests bis zur Freigabe. Das alte System wird Runtergefahren, damit nicht aus versehen eine Produktionsverbindung in die alte DB erstellt wird. Das alte System ist das Fallback System bei misslungenem Test. Es sollte noch einige Tage vorhanden bleiben, für den Fall, dass eine spätere Recherche bei einer Fehleruntersuchung notwendig wird. Es ist wichtig nach dem Switch Personal für das Monitoring der Systeme zu haben. Durch die geänderte Hardware können neue Fehler auftreten bspw. wie bei dieser Migration Performance Engpässe durch ein anderes Verhalten bei der Skalierung von wenigen auf viele CPU Cores. Des Weiteren gab es bei der Migration ein Stuck Recovery, das beim Projektmanagement als Risiko eingeplant werden sollte. Es ist sinnvoll die Performance Änderung per Statspack/AWR Report vorher und nachher zu vergleichen.

Kontaktadresse:

André Schulte  
Radiumstrasse, 14  
51069 Köln

Telefon: +49 (0) 151-14421734  
E-Mail contact@andschulte.de  
Internet: andschulte.de