

# Die Migration von Oracle-Datenbanken

Andreas Reinhardt, OPITZ CONSULTING GmbH

*Dieser Artikel erläutert die Begriffe „Migration“ und „Upgrade“ und zeigt Möglichkeiten der Übernahme einer Datenbank auf neue Plattformen. Dabei kommen Vor- und Nachteile sowie die erforderliche Downtime zur Sprache.*

Eine Migration von Datenbanken kann aus verschiedenen Gründen erfolgen: Für die eine Datenbank wurde der Wechsel der Hardware-Architektur oder der Austausch des Betriebssystems beschlossen. Eine zweite Datenbank soll an die neueste Version der Datenbank-Software angepasst werden. Die dritte Datenbank wird von einem Test- in ein Produktivsystem übernommen.

Der Begriff „Migration“ wird in diesem Beitrag verwendet für den Wechsel der Hardware und/oder des Betriebssystems. Oft wird beides gleichzeitig ausgetauscht. Um eine Datenbank auf den neuen Server zu übertragen, sind deshalb zwei Fragen zu beantworten: „Ändert sich die Architektur des Mikroprozessors und ändert sich auch das Betriebssystem?“

Mit der Prozessor-Architektur ändert sich die Byte-Reihenfolge, in der die Daten gespeichert sind. Zum Beispiel speichert man die Hexadezimalzahl „1A2B“ mit zwei Byte im Format „Big Endian“ als „1A2B“. Im Format „Little Endian“ sind diese zwei Byte in der Reihenfolge „2B1A“ gespeichert. Eventuell ist also auch ein „Endian-Wechsel“, also eine Konvertierung der Daten in das andere Speicherformat, notwendig. Das aktuelle Endian-Format und das des neuen

PLATFORM_ID	PLATFORM_NAME	ENDIAN_FORMAT
12	Microsoft Windows x86 64-bit	Little
17	Solaris Operating System (x86)	Little
20	Solaris Operating System (x86-64)	Little
1	Solaris[tm] OE (32-bit)	Big
2	Solaris[tm] OE (64-bit)	Big

Abbildung 1: Abfrage des Endian-Formats

Systems sind abrufbar in den Datenbank-Views „V\$database“ und „V\$transportable\_platform“ (siehe Abbildung 1).

Der Begriff „Upgrade“ steht für die Aktualisierung des Release der Software, beispielsweise von Oracle Database 10g auf Oracle Database 11g. Mit dieser Änderung eines Produkts auf eine höherwertige Version werden die Nutzung neuer Funktionen, die Beseitigung von Software-Fehlern und die Beibehaltung des zeitlich begrenzten Hersteller-Supports angestrebt. Aus der Kombination von Upgrade und Migration ergeben sich die Umstellungsvarianten (siehe Abbildung 2).

## Zertifizierung sichern

Wenn die Entscheidung für eine Migration gefallen ist, wird meistens auch ein

Upgrade der Datenbank-Software erwo-gen. Eine Downtime, in der das Upgrade erfolgen kann, ist dann ohnehin erforderlich. Hierbei ist zu prüfen, ob die neue Version der Datenbank-Software für die geplante Betriebssystem-Version schon zertifiziert ist. Nur so kann sich der Datenbank-Administrator auch sicher sein, dass die Support-Abteilung der Firma Oracle ein mögliches Fehlverhalten auf der jeweiligen Software-Plattform analysiert. Die derzeit gültige Zertifizierungsreferenz dafür findet sich auf der Seite <https://support.oracle.com> -> Certifications -> Select Product, Release, Platform. Zum Beispiel: „Oracle Database 11.2.0.2.0 is certified on Linux x86-64 Red Hat Enterprise Linux 5“. Auch die Anwendungsentwickler sind zu fragen, ob ihre Programme schon für die neue Datenbank-Version freigegeben wurden. Ist dies nicht der Fall, dann wird mit dem bisher gültigen Release weitergearbeitet.

## Migration durch Import

Zwei häufig genutzte Tools für die Migration einer Datenbank sind Data Pump Export und Data Pump Import. Zunächst wird die neue Datenbank auf dem neuen Server mit der gewählten Datenbank-Version erstellt. Danach wird mit dem Export-Tool aus der alten Datenbank ein Dumpfile geschrieben, das die CREATE-Kommandos aller Datenbank-Objekte sowie die Daten der Tabellen enthält. Das Import-Tool der

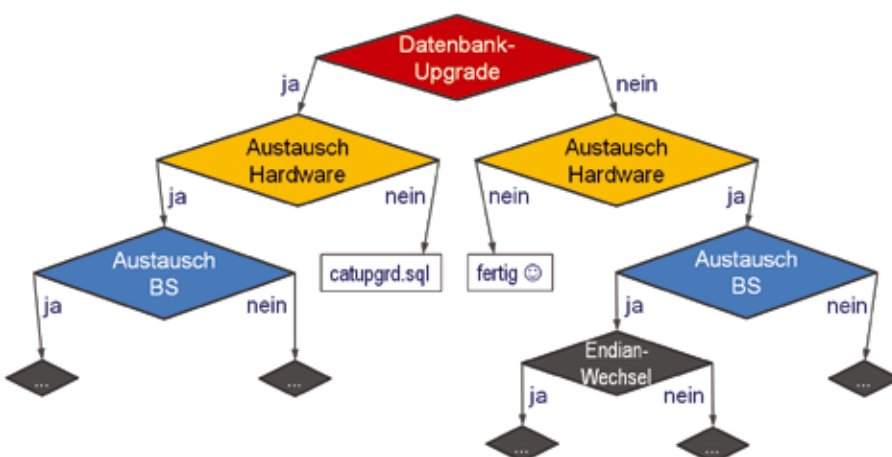


Abbildung 2: Varianten für Upgrade und Migration

neuen Datenbank liest aus diesem Dumpfile die Kommandos, legt die Objekte neu an und füllt die Tabellen mit den Daten. Die Objekte aus dem Verwaltungsbereich der Datenbank – Oracle nennt diesen Bereich das „Data Dictionary“ oder das Schema „SYS“ – werden dabei nicht in die neue Datenbank übertragen, denn diese Objekte gibt es dort schon.

Die Vorteile:

- Das Dumpfile ist unabhängig vom Betriebssystem und mit Einschränkungen auch unabhängig von der Datenbank-Version. Deshalb wird diese Migration oft mit einem Upgrade kombiniert.
- Es ist eine detaillierte Auswahl der zu importierenden Objekte möglich. Oft werden nur die Objekte bestimmter Schemata (= Datenbank-User) importiert.
- Da die Tabellen und Indizes beim Import neu erzeugt werden, sind sie gleichzeitig in einem reorganisierten Zustand. Damit spart man Speicherplatz und die Performance der Datenbank-Abfragen verbessert sich.

Es gibt aber auch Nachteile:

- Der Aufbau der Indizes kostet viel Zeit, weil ein Index eine sortierte Baumstruktur darstellt. Man kann die Indizes auch später erzeugen, wenn mit der neuen Datenbank schon gearbeitet wird. Dazu dient ein SQL-File, welches mit einem Pseudo-Import erzeugt wird und die entsprechenden Kommandos enthält. Allerdings dauern die Datenbank-Abfragen in der Zwischenzeit ohne Indizes natürlich länger.
- Wenn eine ganze Datenbank (außer dem Data Dictionary) importiert wird, sind in der neuen Datenbank immer schon einige Objekte vorhanden, die das Import-Tool neu erzeugen möchte. Im Protokoll erscheinen dadurch Fehlermeldungen, die gründlich geprüft werden müssen.

Die Downtime:

- Die Daten sollen in einem konsistenten Zustand übernommen werden. Ein realistischer Ablauf dafür wäre: Export der alten Datenbank und Import in die neue, Test der Fachabteilung auf der neuen Datenbank, Beginn der Downtime, Löschen der Anwendungs-User in der neu-

en Datenbank und nochmaliger Import dieser User mit deren aktuellen Daten, Umschalten der Connect-Aufrufe (Net Service Names) der Anwendung auf die neue Datenbank, Ende der Downtime.

- Der Import lässt sich durch mehrere parallele Import-Prozesse beschleunigen.
- Wenn auf der neuen Datenbank ein Database-Link zur alten Datenbank erzeugt werden kann, dann ist ein Import im Netzwerk-Modus möglich. Dabei wird gar nicht erst das Dumpfile erzeugt, sondern gleich aus der alten Datenbank gelesen. Das spart Zeit und Plattenplatz. Dieser Modus ist nicht möglich, wenn Tabellen mit Spalten vom Datentyp LONG oder LONG RAW übernommen werden oder wenn NESTED TABLES importiert werden sollen.

Wichtig: Die Migration mit Data Pump Import ist gut vorzubereiten und zu testen. Dabei kann auch die notwendige Import-Zeit geschätzt werden, wenn man bei einem Test-Import nur einen Teil der Datenbank-Objekte importiert und die dafür benötigte Zeit dann auf die gesamte Datenbank hochrechnet. Durch eine korrekte Einstellung des Zeichensatzes beim Anlegen der neuen Datenbank sowie durch Zeichensatz-Variablen in der Systemumgebung ist eine unbeabsichtigte Konvertierung der Daten zu vermeiden. Wenn nur die Objekte ausgewählter User importiert werden, darf man den Pseudo-User „PUBLIC“

nicht vergessen, weil manche User Objekte vom Typ „PUBLIC SYNONYMS“ besitzen; diese würden sonst invalid.

Nach dem Import sollte immer auch eine sofortige Statistiksammlung mit dem Package „DBMS\_STATS“ erfolgen. Sollte es bei der Syntax Probleme mit den Maskierungen geben (beispielsweise beim „exclude=schema: „in („system‘,sysman‘)““), dann empfiehlt es sich, diese Parameter in ein File zu schreiben, welches dann als Aufruf-Argument in der Kommandozeile steht: „oracle@host\_new> impdp system/<password> parfile=impfull\_par.txt“.

## Migration durch

### Transportable Tablespace

Data Pump Import bietet auch den Modus „Transportable Tablespace“. Ein Tablespace besteht aus Datendateien. Hierbei wird ebenfalls eine neue Datenbank aufgebaut, die vorerst noch keine Anwendungs-Tablespaces enthält. Danach werden zunächst die Objekt-Typen „profile“, „role“ und „role\_grant“ aus der alten Datenbank importiert. Dann importiert man die Anwendungs-User mit deren Prozeduren, Views und System-Privilegien. Diese Definitionen können mit dem Parameter „content=metadata\_only“ übernommen werden, der sowohl beim Export als auch beim Import gültig ist. Anschließend werden die Anwendungs-Tablespaces der alten Datenbank in den Modus „read only“ geschaltet und deren Dateien, welche die Anwendungs-Tabellen

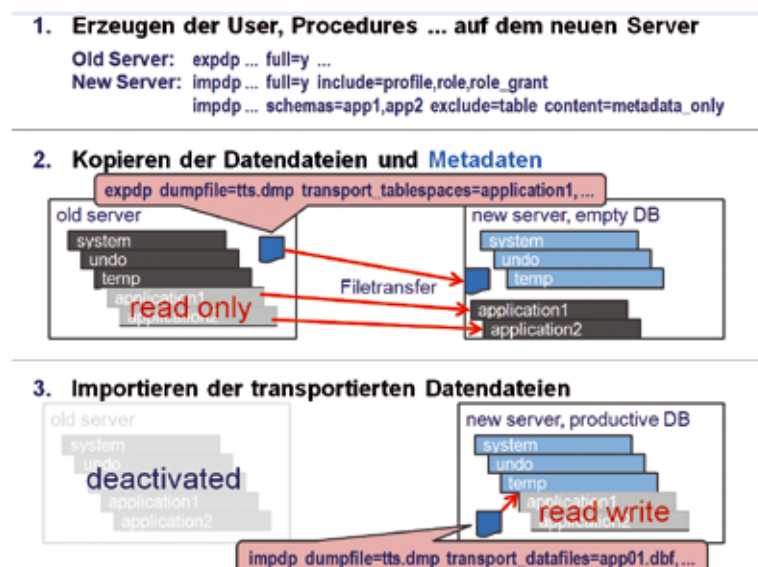


Abbildung 3: Transportable Tablespace

enthalten, zur neuen Datenbank kopiert. Sie sollen dort mit einem Import-Kommando an die neue Datenbank angehängt werden.

Die Beschreibungen der Tablespaces sowie der darin enthaltenen Tabellen und Indizes, die ja beim Import in das neue Data Dictionary eingefügt werden müssen, werden danach mit einem Export-Kommando in ein kleines Dumpfile geschrieben, welches nur diese Metadaten enthält. Dieses Metadaten-File wird ebenfalls zur neuen Datenbank kopiert und durch dasselbe Import-Kommando ausgelesen, welches die Dateien anhängt (siehe Abbildung 3).

Die Vorteile sind:

- Die Tabellen und Indizes werden nicht neu erzeugt, sondern im aktuellen Zustand übernommen. Das spart Zeit.
- Diese Migration ist mit einem Upgrade kombinierbar.
- Ein Endian-Wechsel ist möglich. Dafür werden die Datendateien auf die neue Plattform kopiert und dort mit dem Tool RMAN (= Recovery Manager) konvertiert.

Die Nachteile:

- Die betreffenden User müssen in der neuen Datenbank vor dem Transport der Tablespaces mit allen Profiles, Procedures, Views, Synonyms etc. erzeugt werden.
- Nach dem Import müssen eventuell noch Objekt-Privilegien vergeben werden.
- Es erfolgt keine Reorganisation der Tabellen und Indizes.

Die Downtime:

- Wenn gewiss ist, dass sich an den Definitionen der Anwendungs-User, deren Prozeduren und Privilegien nichts mehr ändert, kann die Downtime nach dem Import dieser Objekte durch die Umschaltung der Tablespaces auf „read only“ starten.
- Nach Beginn der Downtime wird das Metadaten-File erzeugt, siehe Abbildung 3, Punkt 2. Das geht schnell. Dann ist die Zeit für das Kopieren der Datendateien zu berücksichtigen. Zum Test kann auch hier beim Kopieren mittelgroßer Dateien die Zeit gemessen und auf die gesamte Datenbank hochgerechnet

werden. Weiterhin ist die Zeit für den Endian-Wechsel zu addieren, bei dem die Dateien ja auch in ein TEMP-Verzeichnis kopiert werden. Zuletzt wird das Import-Kommando die Dateien an die neue Datenbank „anhängen“, was wiederum recht schnell geht. Obwohl die Downtime nun beendet ist, sollten alle PL/SQL-Objekte noch rekompiliert werden.

- Das Kopieren großer Dateien kann länger dauern als der Import einzelner User. Besonders dann, wenn die Dateien einer Test-Datenbank, die in die Produktion überführt werden soll, nicht mehr viele Objekte enthalten und deshalb zur Hälfte leer sind. Andererseits kann diese Migrationsvariante jedoch auch schneller sein als das Laden von Milliarden von Datensätzen und der langwierige Aufbau der Indizes beim Import.

Transportable Tablespaces werden vor dem Kopieren in den Status „read only“ geschaltet. Tablespaces, die zu Datenbanken mit Versionen kleiner 11g gehören und schon in diesem Status „read only“ sind, sollten je-

# SYSTEM.OUT.PRINTLN ("HELLO WORLD.");

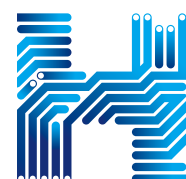
**HAYS** Recruiting experts  
in Information Technology

**WIR FINDEN SPEZIALISTEN,  
DIE DIESE SPRACHE SPRECHEN.**

**Hello World. Sie fühlen sich angesprochen?**

Dann sprechen Sie die Sprache unserer Kunden. Denn wer in der IT die richtigen Codes nutzt, gewinnt den entscheidenden Vorsprung. Deshalb rekrutieren wir die passenden IT-Experten, die Unternehmen vorwärtsbringen.

[hays.de/it](http://hays.de/it)



doch vorher einmal auf „read write“ geschaltet werden. Damit verhindert man Fehler, die durch alte Kompatibilitätseinträge verursacht werden können. Nach dem Import müssen die anderen Tablespace natürlich auch wieder auf „read write“ geschaltet sein. Zu beachten ist auch, dass nach einem Export oder Import mit „content=metadata\_only“ die Tabellen- und Indexstatistiken gesperrt sind und damit nicht automatisch erneuert werden. Die Statistiken sind also mit der Prozedur „dbms\_stats.unlock\_schema\_stats“ zu entsperren.

Zuletzt sei erwähnt, dass vor einem Transport der Tablespaces geprüft werden sollte, dass keine Abhängigkeiten bestehen zwischen Objekten in diesen Tablespaces und solchen in anderen Tablespaces, die nicht mit transportiert werden. Der Import-Parameter „transport\_full\_check=y“ hilft bei dieser Prüfung.

### Migration durch Konvertierung der Datenbank

Ähnlich wie bei der Migration einzelner Tablespaces kann auch die gesamte Datenbank mit allen Datendateien auf eine neue Hardware-Plattform migriert werden. Allerdings ist diese Sonderform der Transportable Tablespaces nur möglich bei übereinstimmendem Endian-Format, etwa von Microsoft Windows nach Linux. Es reicht dabei jedoch kein einfaches Kopieren aus, denn einige Blöcke der Tablespaces „system“ und „undo“ müssen dennoch reformatiert werden. Dafür wird das RMAN-Kommando „convert database“ verwendet.

Wie schon bei der Konvertierung einzelner Datendateien kann das sowohl auf dem alten als auch auf dem neuen Server geschehen. Zunächst prüft man mit der Prozedur „dbms\_tdb.check\_db“, ob die Datenbank alle Voraussetzungen für eine solche Migration erfüllt. Des Weiteren ermittelt man mit der Prozedur „dbms\_tdb.check\_external“, ob Datenbank-Objekte wie External Tables, Directories oder Binary Files (bfile) existieren, die manuell auf den neuen Server kopiert werden müssen. Dann öffnet man die alte Datenbank im Modus „read only“ und startet das Kommando „convert database“. Dabei werden alle Datendateien in ein TEMP-Verzeichnis konvertiert. Zusätzlich wird ein Transport-Script erzeugt, welches unter anderem das SQL-Kommando

zum Erzeugen eines neuen Control-Files enthält. Nach dem Kopieren aller Dateien (konvertierte Datendateien, Datenbank-Parameter-File, Transport-Script) auf den neuen Server werden dort das Parameter-File und das Transport-Script editiert, um die neuen Dateipfade einzutragen. Dann erstellt man ein neues Password-File. Zuletzt wird das Transport-Script ausgeführt und durch dieses die neue Datenbank mit der Option „resetlogs“ geöffnet. Dieses Script legt außer dem Control-File auch die Redo-Log-Files sowie die Temp-Files an und rekompiliert alle PL/SQL-Objekte.

Die Vorteile:

- Die neue Datenbank muss nicht extra vor der Migration erzeugt werden.
- An den Inhalten der Datenbank wird keine Änderung vorgenommen.

Die Nachteile:

- Für die Tablespaces „system“ und „undo“ ist kein Endian-Wechsel möglich.
- Die alte Datenbank muss im Modus „read only“ geöffnet sein, während das Tool RMAN die Konvertierung vornimmt.
- Ein gleichzeitiges Upgrade wird nicht durchgeführt.
- Tabellen und Indizes werden nicht reorganisiert.

Die Downtime:

- Die Downtime beginnt, wenn die alte Datenbank im Modus „read only“ geöffnet wird, um das CONVERT-Kommando zu starten. Der RMAN kopiert nun alle Datendateien in das TEMP-Verzeichnis. Von hier aus werden danach alle notwendigen Dateien auf den neuen Server kopiert. Wenn das Ziel-Verzeichnis des neuen Servers gleich als TEMP-Verzeichnis zur Verfügung gestellt werden kann, spart man sich die zweite Kopier-Aktion. Das anschließende Erzeugen des neuen Control-Files und der anderen Dateien geht schnell. Das Rekompilieren läuft dann, während die Datenbank schon geöffnet ist.

Bei einer solchen Konvertierung der gesamten Datenbank scheint die Vorbereitung recht aufwändig zu sein: Konvertieren, Kopieren aller Dateien, Editieren von Parameter-File und Transport-Script. Der Aufwand

ist jedoch nicht größer als das vorher notwendige Erzeugen einer neuen Datenbank in den anderen Migrations-Varianten.

Nicht immer ist auch ein komplettes Kopieren aller Dateien notwendig. Geprüft werden sollte beispielsweise, ob eine direkte Verwendung der Speichereinheiten durch physischen Transport der Platten oder durch Umhängen der LUNs im SAN möglich ist.

Das neu erzeugte Control-File enthält natürlich nicht das RMAN-Repository der alten Datenbank. Ein baldiges Backup ist aber auch deshalb notwendig, weil die Backups der alten Datenbank in der neuen ohnehin nicht nutzbar sind.

### Migration durch Data Guard

Eine Data-Guard-Konfiguration enthält eine produktive Primär-Datenbank, die mit einer oder mehreren Standby-Datenbanken verbunden ist. Eine Standby-Datenbank ist eine Kopie der Primär-Datenbank, in der die produktiven Änderungen nachgezogen werden. Sie dient also der Hochverfügbarkeit, indem alle Daten und alle Manipulationen daran dupliziert werden. Fällt die Primär-Datenbank aus, dann ist ein schnelles Umschalten auf die Standby-Datenbank möglich. Data Guard überwacht beide Seiten und erleichtert das gezielte Umschalten im Betrieb (Switchover) und das Umschalten im Fehlerfall (Failover).

Weil das Switchover in ein bis zwei Minuten erfolgen kann, eignet sich eine Standby-Datenbank für eine Migration mit geringster Downtime. Dabei sind zwischen dem alten Server (Primär-Datenbank) und dem neuen (Standby-Datenbank) folgende Kombinationen denkbar:

1. Gleiches Betriebssystem, jedoch unterschiedliche Distributionen (RedHat / SuSE) oder verschiedene Hardware (Hersteller, CPU, RAM etc.)
2. Gleiche Betriebssystem-Familie, jedoch verschiedene Architekturen (Linux x86 / Linux x86\_64)
3. Verschiedene Betriebssysteme mit dem gleichen Endian-Format (Windows 32 bit / Linux 64 bit)

Abbildung 4 zeigt die verschiedenen Möglichkeiten. Die ID in der obersten Zeile der Tabelle ist die „platform\_id“ aus der View „V\$transportable\_platform“.

# Wir liefern das Know-how für Ihren Erfolg

## PROJEKTE

Keine „fixen Ideen“, sondern Projekte mit klarer Ziel- und Ablaufplanung.

## PROFILE

Logistikmanager berichten. Direkt, authentisch und unverschnörkelt.

## PROZESSE

Der Blick auf die Supply Chain – von der Beschaffung über Produktion und Distribution bis zur Entsorgung.

## PRODUKTE

Neuheiten für die Praxis. Was sagen Anwender über Produkte und logistische Lösungen?

## PROGNOSEN

Die Rubrik für Theorie und Strategie, neue Ideen, Trends und Märkte.

## EXTRA

Die Spezialrubrik für den vertiefenden Blick auf besondere Themen.

## NACHGEFRAGT

Die Rückschau auf Entwicklungen und Erfahrungen, Erfolge und Enttäuschungen interessanter Projekte.



## ONLINE-ANGEBOT

Unter **logistik-heute.de** finden Abonnenten u.a. tagesaktuelle News, Marktübersichten, Studien, ein Karriere-Portal und haben **kostenfreien** Zugriff auf die gesamte Artikeldatenbank.

## Noch mehr Logistik-IT?



Jetzt kostenloses Heft bestellen!  
[www.logistik-heute.de](http://www.logistik-heute.de)  
Tel. +49 (0)89/32391-314

**LOGISTIK**  
**HEUTE**

Standby-DB \ DB	MS Win 32bit (ID=7)	MS Win 64bit (ID=12)	Linux 32bit (ID=10)	Linux 64bit (ID=13)	HP-UX PA-RISC (ID=3)	HP-UX Itanium (ID=4)	Solaris x86-64 (ID=20)	Solaris SPARC (ID=2)	IBM AIX POWER (ID=6)
MS Win 32bit		10g	11g	11g					
MS Win 64bit	10g			11g					
Linux 32bit	11g			10g					
Linux 64bit	11g	11g	10g				11g		
HP-UX PA-RISC						10g			
HP-UX Itanium					10g				
Solaris x86-64				11g					

Abbildung 4: Plattform-Kombinationen für Data Guard (Auswahl)

Wenn noch keine Standby-Datenbank vorhanden ist, werden zunächst der neue Server bereitgestellt, die Primär-Datenbank dupliziert und Data Guard konfiguriert. Dann erfolgt das Switchover. Die Standby-Datenbank arbeitet nun als Primär-Datenbank und alle Connect-Aufrufe der Clients kommen automatisch dort an. Zuletzt kann die Data-Guard-Konfiguration wieder entfernt werden.

Die Vorteile:

- Sehr kurze Downtime
- In einigen Betriebssystem-Familien ist ein Wechsel der Architektur von 32 Bit auf 64 Bit und umgekehrt möglich
- Diese Migration kann auch mit einem „Rolling Upgrade“ kombiniert werden. Dabei wird die bisher angenommene Physical-Standby-Database in eine Logical-Standby-Database überführt. „Physical Standby“ heißt, dass die Änderungen an den Datenblöcken nachgezogen werden. „Logical Standby“ bedeutet, dass stattdessen die SQL-Kommandos ablaufen, die in der Primär-Datenbank zu den Änderungen geführt haben.

Die Nachteile:

- Eine Reorganisation der Tabellen und Indizes erfolgt nicht.
- Wenn ein Architekturwechsel von 32 Bit auf 64 Bit oder umgekehrt erfolgt, ist anschließend ein Rekompilieren aller PL/SQL-Objekte in der neuen Datenbank notwendig, bevor diese genutzt werden kann.
- Eine Logical-Standby-Database unterstützt nicht alle Datentypen, weshalb eventuell die Kombination der Migra-

tion mit einem „Rolling Upgrade“ nicht möglich ist. Diese Kombination ist auch nur möglich von 32 Bit auf 64 Bit, nicht umgekehrt.

Die Downtime:

- Der Aufbau einer Data-Guard-Konfiguration erfolgt ohne Downtime.
- Das Switchover ist eine kurze Unterbrechung, die nur ein bis zwei Minuten dauert.
- Die Zeit für das Rekompilieren der PL/SQL-Objekte beim Wechsel von 32 Bit auf 64 Bit (oder umgekehrt) hängt von der Anzahl der Objekte ab.

Der Aufbau und die Administration einer Data-Guard-Konfiguration können mit den Tools RMAN, Data Guard Manager Line-mode (DGMGRL) oder mit dem Enterprise Manager erfolgen. Der Enterprise Manager ist jedoch nur verwendbar, wenn die „platform\_id“ auf beiden Seiten identisch ist.

Auch diese Form der Migration sollte ausreichend getestet werden. Auf der Standby-Datenbank wird also zuerst geprüft, ob die Änderungen der Primär-Datenbank auch richtig verarbeitet werden. Um das auch mit SQL-Abfragen kontrollieren zu können, wird sie im Modus „read only“ geöffnet. In der Version 11g kann sie sogar im Modus „read write“ getestet werden. Erst nach diesem Test erfolgt das Switchover.

#### Fazit

Um die beste Variante für eine Migration mit oder ohne Upgrade zu finden, sind die verwendeten Hardware-Architekturen, Betriebssysteme und Datenbank-Versionen zu betrachten. Weiterhin ist die zur Verfü-

gung stehende Downtime wichtig, die ja im Produktiv-System immer möglichst kurz sein soll. So ist das Kopieren von Datendateien oft schneller als ein Import, bei dem alle Datenbank-Objekte neu erzeugt werden. Die Downtime kann entweder wenige Minuten oder auch mehrere Stunden betragen.

Alle Migrationen benötigen oft mehr Speicherplatz, als der Datenbankadministrator anfangs vermutet. Export-Dumpfiles, konvertierte Datendateien oder das Duplikat einer kompletten Datenbank erreichen heutzutage Größen im Giga- oder Tera-Byte-Bereich. Der Import und damit der Neuaufbau aller Objekte einer Anwendung in die neue Datenbank kann hingegen zu einer erheblichen Einsparung von Plattenplatz führen. Nicht selten werden durch diese Art der Reorganisation die Objekte um die Hälfte kleiner und damit die Datendateien anfangs nicht so groß wie in der alten Datenbank.

Immer ist zu berücksichtigen, dass auch ein Umschalten der Connect-Aufrufe der Anwender vorbereitet werden muss. Dafür stehen beispielsweise Net Service Names oder LDAP-Directories zur Verfügung, die entsprechend zu konfigurieren sind. Alle Angaben in diesem Artikel beziehen sich auf Oracle-Datenbanken der Versionen 10g und 11g. Werden ältere Versionen migriert, sind weitere Besonderheiten in den Dokumentationen zu beachten.

Um den Gesamtaufwand einer Migration abzuschätzen, dürfen für die neuen Datenbanken die Konfiguration der Backups und der Eintrag in die Betriebsdokumentation nicht unberücksichtigt bleiben. Auch das Abschalten der alten Datenbanken kostet Zeit. Neben den hier genannten Varianten einer Migration gibt es weitere Lösungen. So kann man – ähnlich wie bei Data Guard – die Datenbank duplizieren und eine Replikation der Datenänderungen einrichten. Beim Umschalten auf die neue Datenbank fällt auch dabei die Downtime sehr gering aus. Alle genannten Operationen erfordern ein entsprechendes Know-how. Eine Migration ist nicht „mal so schnell nebenbei erledigt“.

Andreas Reinhardt  
andreas.reinhardt@opitz-consulting.com