

Minimal Downtime Oracle 11g Upgrades

Martin Decker
ora-solutions.net

Schlüsselworte:

Upgrade, Minimal Downtime, Transportable Tablespaces, Transportable Database, Transient Logical Standby, SQL Apply, Rolling Upgrade

Einleitung

Nachdem der Premier Support für das Datenbank-Release 10gR2 im Juli 2010 endete und nur das erste Jahr des Extended Supports zusätzlich kostenfrei ist, stehen viele Kunden vor der Aufgabe des Upgrades von Oracle 10g nach 11gR1 oder 11gR2 bis spätestens Juli 2011. Dieser Vortrag zeigt Möglichkeiten, den Oracle-Upgrade mittels Logical Standby sowie mittels Transportable Tablespaces mit minimaler Downtime durchzuführen. Es wird aufgezeigt, welche Voraussetzungen für die beiden Alternativen jeweils benötigt werden sowie die potentiellen Probleme bei der Durchführung. Der Vortrag enthält zudem einen praktischen Teil, bei dem die beiden Varianten demonstriert werden.

Gründe für Upgrade

Die Gründe für den Upgrade auf Oracle 11g Release 1 oder 11g Release 2 sind unterschiedlich. Der Hauptgrund dürfte das Ende des Premier Support für die Oracle Version 10gR2 sein. Ein anderer Grund könnte sein, zusammen mit dem Releasewechsel einen Plattform-Wechsel durchzuführen. Durch das ausgezeichnete Preis-/Leistungsverhältnis nimmt die Verbreitung von Oracle Datenbank Installationen auf Linux x86-64 auf Basis von Intel Nehalem CPUs rasant zu. Des öfteren ist zudem eine Migration von HP-UX, Solaris oder AIX (Big Endian) auf 64bit-Linux (Little Endian) und der damit verbundenen Endian-Änderung notwendig.

Auf der anderen Seite sind natürlich einige Kunden auch an speziellen Features interessiert, die nur die aktuellen Releases bieten, z.B.:

- bessere Performance (z.B. Result Cache)
- SQL Plan Stabilität durch SQL Plan Management
- Real Application Testing (Database Replay bzw. SQL Performance Analyzer)
- Adaptive Cursor Sharing
- verbessertes DBMS_STATS Sampling

Upgrade-Methoden

Die Upgrade-Methoden unterscheiden sich stark in der Dauer der (geplanten) Downtime, der Komplexität sowie der Möglichkeit sowohl Cross-Plattform sowie Cross-Endianness zu erlauben.

Die folgende Tabelle liefert einen Überblick über die vorhandenen Methoden:

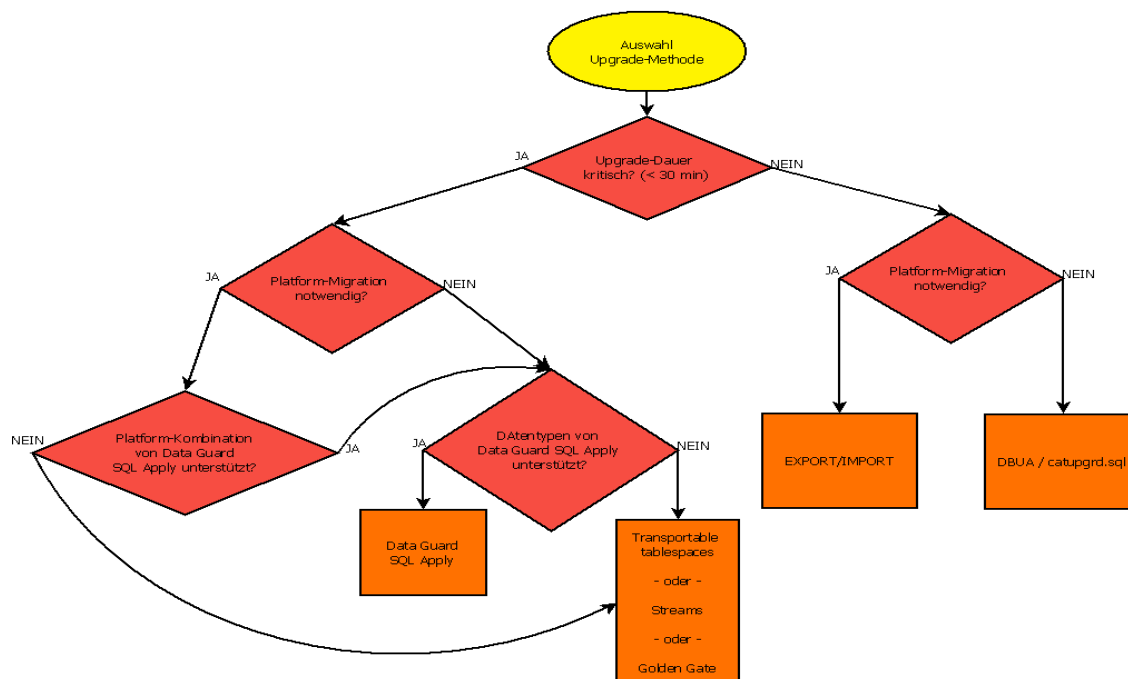
Upgrade-Methode	Dauer	Komplexität	Cross-Plattform	Cross-Endian
DBUA (Oracle recommended)	>30 min, unabhängig von Größe	sehr einfach	Nein	Nein
CLI - @catupgrd	>30 min, unabhängig von Größe	einfach	Nein	Nein
Export/Import	>30 min abhängig von Größe	mittel	Ja	Ja
Oracle Streams (in Zukunft desupported)	wenige Minuten	komplex	Ja	Ja
Oracle Golden Gate (Lizenzkosten)	wenige Minuten	komplex	Ja	Ja
Transportable Database (TDB) - kein Upgrade, nur Plattform-Migration	wenige Minuten	mittel	Ja	Nein
Transportable Tablespaces (TTS/xTTS)	wenn selbe Plattform/Endian wenige, Minuten, sonst abhängig von DB Größe	komplex	Ja	Ja
Data Guard Logical Standby SQL Apply	wenige Minuten	komplex	teilweise (MOS 414043.1)	Nein

Die von Oracle Corp. empfohlene Upgrade-Methode ist mittels **Database Upgrade Assistant (DBUA)**. Diese GUI-basierte Upgrade-Methode ist sehr einfach, es finden extensive Prüfungen statt und der Weg soll auch den unerfahrenen DBA ans Ziel bringen. Auf einem Oracle VM System wurde eine Datenbank mit minimal-Optionen in unter 17 Minuten und eine Datenbank mit Komplett-Optionen innerhalb von 50 Minuten von 10.2.0.4 auf 11.2.0.1.2 aktualisiert.

Immer mehr Systeme müssen 24x7 verfügbar sein und die seltenen Wartungsfenster müssen so kurz wie möglich sein. Wir konzentrieren uns deshalb im weiteren Verlauf auf die beiden Methoden "**Transportable Tablespaces (TTS)**" sowie "**Data Guard Logical Standby SQL Apply**", weil dadurch der Upgrade mit der geringstmöglichen Ausfallzeit ermöglicht wird.

Die Entscheidung, ob der Upgrade mittels Transportable Tablespaces bzw. SQL Apply durchgeführt werden soll, liegt meist an der Notwendigkeit einer Plattform-Migration. Während beim Upgrade mittels "Logical Standby SQL Apply" nur sehr begrenzte Plattform-Migrationen möglich sind, z.B. Windows nach Linux und umgekehrt, sind die Möglichkeiten bei Transportable Tablespaces zahlreicher. Der Upgrade via Transportable Tablespaces kann auch speziell dann in Frage kommen, wenn die Datenbank Datentypen verwendet, die von Data Guard SQL Apply nicht unterstützt werden.

Das folgende Entscheidungsdiagramm soll hier helfen:



Transportable Tablespaces (TTS)

Transportable Tablespaces setzt auf das Prinzip, dass ein Tablespace mit allen enthaltenen Segmenten von einer Datenbank in eine andere "eingesteckt", (engl. plugged in) werden kann. Die Ziel-Datenbank kann dabei schon auf einem höheren Release stehen. Verständlicherweise müssen allerdings auch alle applikationsrelevanten Objekte, z.B. Functions, Procedures, Packages, Sequences, Types, etc., "transportiert" werden, die kein Segment in dem transportierten Tablespace besitzen, sondern ihre Information im Data Dictionary speichern.

Voraussetzung für TTS ist, dass sowohl Source Database als auch Target Database mit dem selben Character-Set konfiguriert sind. Zusätzliche Vorsicht ist bei Verwendung von BFILES sowie external tables geboten. Diese werden grundsätzlich bei TT nicht migriert. Ebenso werden Objekte in SYSTEM/SYSAUX Tablespaces sowie SYS/SYSTEM Objects nicht automatisch transportiert.

Das folgende Beispiel beschreibt die notwendigen Schritte beim Einsatz von Transportable Tablespaces für den Upgrade einer 10.2.0.4 Datenbank auf 11.2.0.1.2 unter Linux x86-64 ohne Plattform Migration. Ein "in-place" Upgrade, d.h. der Upgrade der Original-Datafiles eliminiert die Notwendigkeit Datafiles zu kopieren, wird mangels Fallback-Möglichkeit allerdings nicht empfohlen. In der Praxis lässt sich dies durch die Verwendung einer Physical Standby Datenbank auf der Zielseite lösen, weil dadurch das Kopieren der Datenfiles innerhalb der Downtime vermieden wird.

Im Laufe des Workflows werden 3 Data Pump Exports/Imports durchgeführt:

- Data Pump Import via NETWORK_LINK für System-Metadaten (Users, Roles, Role Grants, Profiles)
- Full-Struct Data Pump Export/Import (z.B. Functions, Procedures, Packages, Sequences, Types)
- Data Pump Export mit TRANSPORTABLE_TABLESPACES clause und Data Pump Import mit TRANSPORT_DATAFILES clause

Vorgehensweise:

Ausgangsbasis	
	<ul style="list-style-type: none">• Quell-Host: oel5n1• Quell-Datenbank: O10R204M• Ziel-Host: oel5n2• Physical Standby-Datenbank: O10R2DG• Datenbankname nach Upgrade: O11R2M
Vorbereitung	
	<ul style="list-style-type: none">• Erstellung einer Physical Standby Datenbank mit gleichem Oracle Release wie Source Database• Installation und Patching der 11.2 Oracle Binaries auf dem Zielsystem• Ausführung von <code>utlu112.sql</code> zur Prüfung der Upgrade-Voraussetzungen• Berücksichtigung von Objekten in SYS/SYSTEM Schema bzw. SYSTEM/SYSAUX Tablespaces• Erstellen (DBCA) und Vorbereiten der Zieldatenbank (Database Link, Directories)• Durchführung der Transportable-Tablespace Checks• Importieren der System-Metadaten (Users, Roles, Role Grants, Profile) von der Source Database• Drop der zu transportierenden Tablespaces von der Target Database• Full-Structure-Export von Source Database (exkl. User, Roles, Role Grants, Profiles)
Downtime	
	<ul style="list-style-type: none">• Stoppen der Applikation• Read Only Setzen der zu transportierenden Tablespaces in Source Database• Erzeugen des Create Sequence Scripts• Sicherstellen, dass Physical Standby synchron ist• Stoppen von Media Recovery wenn synchron• Shutdown Physical Standby• Transport oder Umbenennen der zu transportierenden Tablespaces von Physical Standby DB zu Target DB• Data Pump Export mit <code>transport_tablespaces</code> Clause (ca. 15 sek)• Kopieren des Dump Files auf Target System• Data Pump Import mit <code>transport_datafiles</code> Clause (ca. 45 sek)• Read Write Setzen der transportierten Tablespaces in Target Database• Kopieren des Dump Files mit Full-Struct-Export auf Target System• Full-Struct-Import in Target Database (ca. 45 sek)• Grant von System Privileges auf Target Database

Es ist sehr zu empfehlen, den Prozess ausgiebig zu testen, bevor das Produktionssystem angegangen wird. Dies ist sehr einfach möglich, in dem die Physical Standby Datenbank nach dem Setzen eines Guaranteed Restore Points read-write geöffnet wird und die Steps (z.B. Tablespace READ ONLY) statt auf dem Source-System auf dem Standby-System durchgeführt werden. Nach jedem iterativen Test kann das Standby System mittels Flashback wieder zurückgesetzt und mit der Primary Datenbank synchronisiert werden.

Hat man nach mehreren Iterationen genügend Vertrauen in den Workflow und das Target-System ausreichend getestet und dessen Funktion sichergestellt, kann der Upgrade auf dem Produktionssystem beginnen.

Data Guard Logical Standby SQL Apply

Beginnend mit der Oracle Version 10.1.0.3 ist es möglich, Rolling-Upgrades mittels "Logical Standby" durchzuführen. Es gibt allerdings Einschränkungen bzgl. Datentypen, die von Logical Standby SQL Apply unterstützt werden. Hilfreich hierfür sind die Queries:

```
SELECT DISTINCT OWNER, TABLE_NAME FROM DBA_LOGSTDBY_UNSUPPORTED;
SELECT OWNER FROM DBA_LOGSTDBY_SKIP WHERE STATEMENT_OPT = 'INTERNAL
SCHEMA';
```

Diese Queries zeigen, welche Datentypen derzeit in der Quell-Datenbank enthalten sind, die nicht von SQL Apply unterstützt werden sowie welche interne Schemas von Data Guard SQL Apply ignoriert werden.

Folgende Datentypen werden von Oracle Data Guard SQL Apply generell nicht unterstützt:

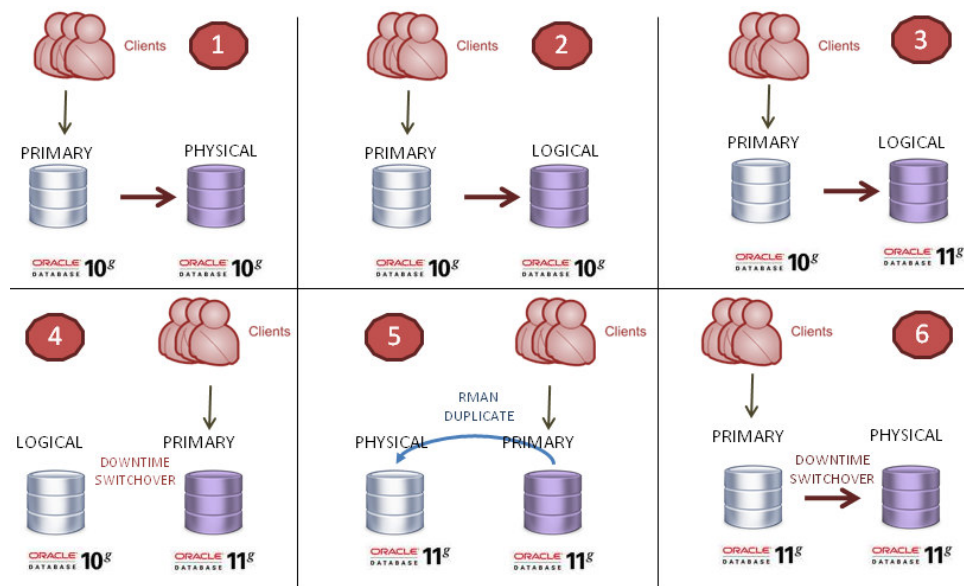
BFILE	UROWID	Encrypted columns
User-defined types	ROWID	XMLType
Multimedia data types (including Spatial, Image, and Context)		
Collections (including VARRAYS and nested tables)		

Um die Einschränkungen zu adressieren, hat man diese Möglichkeiten: a. Vermeidung von Änderungen an diesen Objekten während der kurzen Upgrade-Phase - oder - b. Aufzeichnung von Änderungen und manueller Export/Import dieser Objekte nach dem Upgrade - oder - c. Prüfung, ob die Datentypen durch Extended Datatype Support (EDS) unterstützt werden.

Extended Datatype Support (EDS) steht ab 10.2.0.4 und 11.1.0.7 zur Verfügung und bietet die Möglichkeit, mit Hilfe von Triggern und Hilfstabellen, nicht unterstützte Datentypen wie z.B. user-defined data types in "native" data types zu konvertieren und auf dem Zielsystem wieder in die Original-Tabelle zurückzuführen.

"Cross-endian" Plattform-Migrationen werden von Data Guard SQL Apply nicht unterstützt und "same-endian" Migrationen nur teilweise. Einen guten Überblick bietet MOS Note 414043.1

Die folgende Graphik zeigt die Schritte im Überblick:



Ein Upgrade einer Oracle-Umgebung mit Primary Datenbank und Physical Standby Datenbank von Oracle 10.2.0.4 auf 11.2.0.1.2 wird in folgenden Schritten durchgeführt:

Prerequisite: Es ist zu empfehlen, sowohl auf Primary, als auch auf der Standby Seite Flashback Database zu aktivieren. Falls dies aus Performance-Gründen auf Primary nicht möglich sein sollte aber dennoch Flashback Database benutzt werden möchte, so kann später ein garantierter Restore Point nur im Mount-Mode gesetzt werden, was zusätzliche Downtime bedeutet. (ORA-38787 Creating the first guaranteed restore point requires mount mode when flashback)

Schritt 1: Ausgangslage

Zu Beginn besteht das Setup aus einer Primary Datenbank und einer Physical Standby Datenbank. Beide Datenbanken sind in der Version 10.2.0.4. Die Applikations-Benutzer sind noch mit der Primary-Datenbank verbunden und können uneingeschränkt arbeiten.

Schritt 2: Konvertieren der Physical Standby Datenbank zur Logical Standby Datenbank

Dann wird die Physical Standby Datenbank mittels
`SQL> ALTER DATABASE RECOVER TO LOGICAL STANDBY db_name;`
zu einer Logical Standby konvertiert. Die Logical Standby Datenbank wird nach der Konvertierung wieder in den Apply-Modus gesetzt sodaß Änderungen der Primary Datenbanken über SQL Apply auf die Logical Standby Datenbank angewendet (applied) werden. Zu beachten ist dabei, dass bei 10g die Logical Standby eine neue DBID und einen neuen DBNAME bekommt. Dies ist z.B. bei Grid Control bzw. RMAN catalog zu berücksichtigen. Bei 11g ist dies bei Verwendung von "transient logical standby" mittels "KEEP IDENTITY"-Claus nicht mehr erforderlich.

Schritt 3: Upgrade der Logical Standby

Nachdem auf dem Standby-Host die Oracle Binaries der Zielversion (Stand September 2010: 11.2.0.1.2) installiert wurden, wird die Logical Standby mittels DBUA oder catupgrd.sql "upgraded". Nach dem Upgrade wird der Apply Modus wieder aktiviert und die 11gR2 Logical Standby wird wieder von der Primary-Datenbank synchronisiert. An dieser Stelle kann nun die 11gR2 Logical Standby Datenbank getestet werden. Sind die Tests positiv, kann nach der Synchronisation der nächste Schritt durchgeführt werden.

Schritt 4: Switchover von Primary 10gR2 auf Logical Standby 11gR2

Dieser Schritt markiert den Beginn der Downtime. Bis hierher konnten die Applikations-Benutzer uneingeschränkt mit der Primary-Datenbank arbeiten. An dieser Stelle wird nun die Primary-Datenbank zur Logical Standby und die Logical Standby wird zur Primary-Datenbank. Die Benutzer können sich dann mit der neuen 11gR2 Primary-Datenbank verbinden und die Verfügbarkeit ist dadurch wieder hergestellt. Falls das System die Standby-Datenbank nur für den Upgrade-Prozess benutzt, ist der Workflow an dieser Stelle beendet. Falls jedoch das Ziel-Setup wieder aus einer Primary und einer Physical Standby bestehen soll, sind die folgenden Schritte notwendig.

Schritt 5: Neuaufbau der Physical Standby

An dieser Stelle wird die ursprüngliche Primary Datenbank als Physical Standby neu aufgebaut. Dies kann mittels RMAN DUPLICATE FOR STANDBY vom Backup oder mittels FROM ACTIVE DATABASE von der laufenden Primary-Datenbank erfolgen.

Schritt 6: optional: Switchover auf ursprüngliche Primary-Datenbank

Optional kann mittels einer zweiten kurzen Downtime und einem Switchover wieder auf die ursprüngliche Primary-Datenbank gewechselt werden.

Ausblick

Die Technologie des Upgrades mittels Logical Standby wurde in der Version 11g noch etwas verbessert. So ist es nun möglich, die Konvertierung von Physical zu Logical Standby mit der Option "KEEP IDENTITY" (transient logical standby) durchzuführen, bei der die DBID und der DBNAME der Standby-Datenbank gleich bleibt. Dadurch ist es möglich, dass die ursprüngliche Primary-Datenbank nach dem Switchover von Primary zu Logical mittels Guaranteed Restore Point zum Zeitpunkt vor der Konvertierung zur Logical Standby zurückgesetzt werden kann. Anschließend kann wie gewohnt mittels Media Recovery synchronisiert werden und das RMAN Duplicate ist nicht mehr notwendig. Um die Handhabung noch stärker zu vereinfachen, hat Oracle das Script "physru.sql" entwickelt. Es bietet die Möglichkeit script-gesteuert den kompletten Upgrade-Prozess durchzuführen. Das Script kann unter MOS 949322.1 heruntergeladen werden.

Die hier behandelten Methoden gelten ebenfalls für das Full-Install Release 11.2.0.2, das für Linux laut MOS Note 742060.1 im Oktober 2010 freigegeben werden sollte.

Referenzen

- Oracle Lifetime Support:
<http://www.oracle.com/support/library/brochure/lifetimesupport-technology.pdf>
- MOS 601807.1 Upgrade Companion 11g Release 1
- MOS 785351.1 Upgrade Companion 11g Release 2
- Oracle Database Upgrade Guide 11g Release 2:
http://download.oracle.com/docs/cd/E11882_01/server.112/e10819/toc.htm
- MOS 837570.1 Complete Checklist for Manual Upgrades to 11g Release 2
- OTN Upgrade Page: <http://www.oracle.com/technetwork/database/upgrade/index-088044.html>
- MOS 880782.1 Support Status and Alerts for Oracle 11g Release 2 (11.2.0.X)
- MOS Note 414043.1: Role Transitions for Data Guard Configurations Using Mixed Oracle Binaries
- Oracle Database Maximum Availability Architecture (MAA) Whitepapers:
<http://www.oracle.com/technetwork/database/features/availability/oracle-database-maa-best-practices-155386.html>
- Oracle Data Guard Concepts and Administration 11gR2, Chapter 12: Using SQL Apply to Upgrade the Oracle Database
http://download.oracle.com/docs/cd/E11882_01/server.112/e17022/rollup.htm#BABGHIGF
- Oracle White Paper—Database Rolling Upgrades Made Easy by Using a Physical Standby Database: <http://www.oracle.com/technetwork/database/features/availability/maa-wp-11g-upgrades-made-easy-131972.pdf>

Kontaktadresse:

Martin Decker

Oracle Certified Master 10g
Oracle Certified Master 11g



ora-solutions.net
Franz-Fischer-Str. 7
D-81677 München

Telefon: +49 (0) 176 787 627 88
E-Mail martin.decker@ora-solutions.net
Internet: <http://www.ora-solutions.net>