

# **RAC Migration von 10gR2 auf 11gR2 über Data Guard mit Handicaps**

**Ralf Appelbaum  
TEAM GmbH  
Paderborn**

**Dr. Andreas Fleige  
Die Continentale  
Dortmund**

## **Schlüsselworte:**

Oracle Database, 10gR2, 11gR2, Data Guard, Rolling Upgrade, Migration, Real Application Clusters, RAC, Automatic Storage Management, ASM, physical Standby, logical Standby, Oracle Maximum Availability Architecture, MAA

## **Einleitung**

Die Herausforderung bei der Continentale Krankenversicherung a.G. bestand darin, einen Oracle 10g Rel.2 zwei Knoten „extended distance“ RAC (Real Application Clusters) mit möglichst geringer Downtime auf einen drei Knoten „extended distance“ RAC der Version 11g Rel. 2 zu migrieren. Im RAC sind vier Datenbanken für verschiedene Anwendungen konfiguriert. Die Datenbank-Software dreier Datenbanken wurden mit dem RAC auf 11g Rel.2 migriert. Eine Datenbank wurde in den neuen RAC übernommen, behielt aber den Softwarestand 10g Rel.2 bei. Mit der Migration verbunden war auch der Wechsel von OCFS2 (Oracle Cluster Filesystem) auf ASM (Automatic Storage Management) als Speicherort für Datenbankdateien und auch Clusterware und Votingfile sind nun im ASM abgelegt.

Mit Unterstützung durch die Firma TEAM wurde der neue RAC auf neuer Hardware aufgesetzt und zunächst die Gesamtkonfiguration separat auf ihre Ausfallsicherheit getestet. Die Migration der Datenbanken auf den neuen RAC erfolgte mittels Oracle Data Guard. Bei drei der Datenbanken kam dabei Logical Standby zum Einsatz. Somit war es entsprechend eines Rolling Upgrades möglich, diese vor dem Migrationstermin auf 11g Rel.2 zu aktualisieren und dennoch zu den produktiven Systemen synchron zu halten. Bei der vierten Datenbank wurde auf dem neuen RAC nur eine Physical Standby Datenbank unter Oracle 10g Rel.2 aufgesetzt. Zum Migrationszeitpunkt selber war dann „nur“ ein Failover auf die Standby Datenbanken im neuen RAC erforderlich.

Das Vorgehen wurde, soweit es ohne Einfluss auf die produktiven Datenbanken möglich war, im Projektverlauf von knapp einem  $\frac{3}{4}$  Jahr mehrfach im echten Umfeld erprobt und in Details weiter verfeinert.

Der Vortrag bzw. dieses Dokument beschreibt neben dem technischen Hintergrund des Migrationsvorgehens auch den Projektablauf und insbesondere die kleineren und größeren technischen Handicaps, die es bei Vorbereitung, Tests und Durchführung der Migration zu bewältigen gab.

## Ausgangssituation

Abgelöst werden sollte ein Oracle Datenbank Server (s. Abb. 1) bestehend aus

- 2 Knoten extended Distance RAC
- SuSE Linux Enterprise Server 10 (2.6.16) 64-bit
- Oracle Clusterware und Datenbank Software alle Version 10g Rel.2 (10.2.0.4.0)
- 4 Datenbanken
- Datendateien abgelegt in OCFS2 Devices

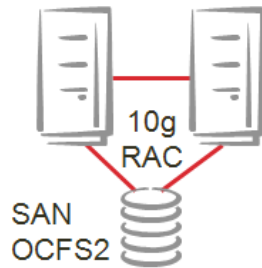


Abb. 1: Ursprungssystem, 10g RAC

Das Zielsystem (s. Abb. 2 und 4) besteht aus

- 2 extended Distance RAC á 3 Knoten
- Spiegelung der Datenbanken über DG zwischen den RAC
- Red Hat Enterprise Linux Server 5.6 (2.6.18) 64-bit
- Oracle Grid Infrastruktur 11g Rel.2
- 3 Datenbanken Oracle Database 11g Rel.2 (11.2.0.2.0)
- 1 Datenbank Oracle Database 10g Rel.2 (10.2.0.4.0)
- Datendateien abgelegt im ASM
  - jede DB in eigene Daten- und FRA-Diskgruppe
  - Spiegelung (normale Redundanz) über Volumes aus zwei SANs

Bei der neuen Umgebung handelt es sich also um eine MAA (Maximum Availability Architecture) Umgebung. Die Intention dieser Umgebung ist bei der Continentale aber weniger die Sicherheit im Desasterfall (die zwei RAC stehen in denselben Rechenzentren), sondern folgende Möglichkeiten:

- zum Produktivsystem identische Testumgebung
- Datenbestand in Test-Datenbanken annähernd aktuell zu Produktiv-Datenbanken
- Softwareupdates (Patches und PSUs) über Data Guard Rolling Update/Upgrade
  - strategisches und erprobtes Vorgehen
  - möglichst geringe Downtime

## Auswahl eines Migrationspfads

Ein Upgrade von einer früheren Oracle Datenbank Version auf Oracle 11g Rel.2 kann erfolgen:

- als "in-place" Upgrade mittels Database Upgrade Assistant (DBUA) oder manuell über SQL-Skripte,
- über Export/Import bzw. Data Pump,
- als Rolling Upgrade mittels Data Guard SQL Apply.

Die zwei weiteren, selten genutzten Upgrade Möglichkeiten "Oracle Streams" und „Oracle Transportable Tablespaces“ wurden nicht berücksichtigt.

Ein Upgrade des bestehenden Systems von Oracle 10g Rel.2 auf 11g Rel.2 schied von vornherein aus, da die Änderungen zwischen den Oracle Versionen im Umfeld Clusterware bzw. Grid Infrastruktur zu gravierend sind. Darüber hinaus sollte das alte System als Fall-Back unangetastet bestehen bleiben, da die Continentale mit einem Upgrade bereits schlechte Erfahrungen gemacht hatte.

Die vier Datenbanken hatten Größen von 688 GB, 212 GB, 71 GB und 17 GB. Damit wäre auch ein Transfer auf den neuen RAC mittels Export/Import bzw. Data Pump noch an einem Wochenende möglich gewesen.

Weitere Kriterien zur Auswahl des Migrationsverfahrens waren jedoch:

- Es soll eine möglichst geringe Downtime aber mit ausreichend Pufferzeit erreicht werden.
- Die Migration soll vorab mehrfach getestet werden.
- Die Anwendungen sollen auf dem neuen System mit realistischem Datenbestand getestet werden können.
- Die zentrale Strategie für zukünftige Updates/Upgrades soll mit diesem Upgrade eingeführt und getestet werden.

⇒ Damit stand fest, die Migration erfolgt über Data Guard Rolling Upgrade!

Dass eine der vier Datenbanken auch in der neuen Umgebung unter der Version 10g Rel.2 weiter laufen muss und nicht auf Oracle 11g Rel.2 aktualisiert wird, hängt mit dem OWB (Oracle Warehouse Builder) zusammen. Dessen Kernfunktionalität ist seit 10g Rel.2 Bestandteil der Datenbank und somit hätte ein Upgrade dieser Datenbank auch ein Upgrade der Anwendung auf den neuen OWB bedeutet, was mit sehr hohem Aufwand verbunden ist.

## **Eingesetzte Konfigurationen für Hochverfügbarkeit**

Im Folgenden werden kurz die technischen Grundlagen der im Projekt eingesetzten Konfigurationen für Hochverfügbarkeit eines Oracle Datenbanksystems dargestellt. Diese sind:

- Oracle RAC bzw. Extended Distance RAC
- Oracle Data Guard
- Maximum Availability Architecture (MAA)

### **Oracle Extended Distance RAC**

Die Datendateien eines Oracle Datenbank-Systems sind in der Regel über ein Storage mittels RAID (Redundant Array of Independent Disks) gegen Ausfall gesichert. Über einen einfachen Oracle Real Application Clusters (RAC) kann darüber hinaus die Laufzeitumgebung eines Datenbank-Systems, also die Instanz, gegen einen Serverausfall geschützt werden. Mehrere Instanzen ein und der selben Datenbank laufen dabei verteilt auf jeweils eigenen Servern (Knoten). Neben der Ausfallsicherheit bietet ein RAC zusätzlich die Möglichkeit, das Datenbank-System horizontal, also durch Ergänzen weiterer Knoten, zu skalieren.

⇒ Was aber ist mit der Absicherung gegen ein Desaster, z.B. bei Ausfall eines Rechenzentrums?

Bei Oracle kann diese Gefahr durch zwei Konfigurationen abgesichert werden:

- Extended Distance RAC
- Data Guard

Die Continentale nutzt Extended Distance RAC um der Gefahr eines Desasters zu begegnen.

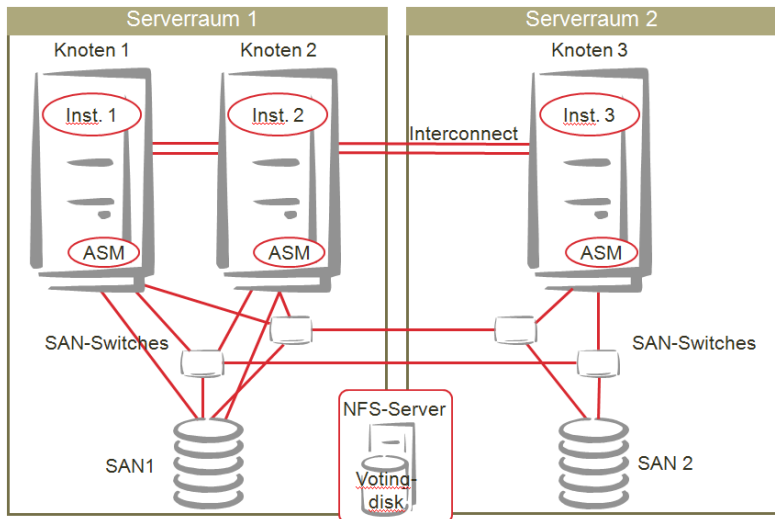


Abb. 2: Extended Distance RAC

Dazu werden beim neuen Zielsystem die drei Knoten des RAC am Standort Dortmund auf zwei Rechenzentren in verschiedenen Gebäuden verteilt. In jedem Rechenzentrum steht ein Storage Area Network (SAN). Die Datendateien der Datenbanken werden über Oracle ASM mittels normaler Redundanz auf die beiden SAN gespiegelt. Im ASM werden für jede Datenbank eine eigene Daten- und Fast-Recovery-Area(FRA)-Diskgruppe angelegt. Somit lässt sich die Kompatibilität der Diskgruppen an die Version der Datenbank anpassen.

In der Grid Infrastruktur 11g Rel.2 werden auch die Dateien der Clusterware, d.h. Clusterregistry und Votingdisk, im ASM gespiegelt abgelegt. Bei Ausfall der Netzwerkverbindung zwischen den Rechenzentren sehen die RAC Knoten jeweils die lokale Kopie der Votingdisk und es entsteht eine so genannte „Split Brain“ Situation, d.h. es ist nicht möglich zu entscheiden, welches der aktive RAC Teil ist. Um diese Situation zu vermeiden, muss eine dritte Kopie der Votingdisk außerhalb der beiden SAN und auch außerhalb der beiden Rechenzentren eingerichtet werden. Bei der Continentale läuft dazu ein NFS Server am Standort Köln, auf dem eine Datei abgelegt ist, die als dritte Spiegelplatte in der ASM Diskgruppe fungiert.

### Oracle Data Guard

Auch Data Guard kommt bei der Continentale zum Einsatz, dient aber vorwiegend dem Aufbau der Testumgebung und zukünftigen Rolling Updates/Upgrades.

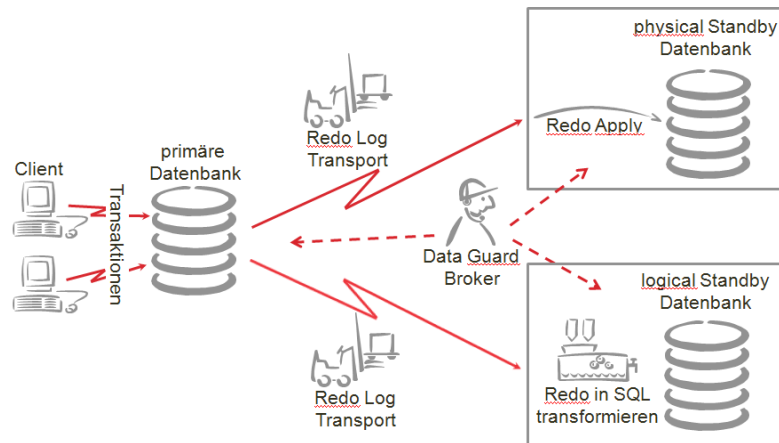


Abb. 3: Data Guard

Bei Data Guard werden die Transaktionen einer primären Datenbank, d.h. die Änderungen an den Daten, über die Redo Log Informationen auf ein oder mehrere physikalisch getrennt stehende, sekundäre Datenbanken übertragen und dort nachgefahren.

⇒ Eine Standby Datenbank ist somit eine, ggf. zeitlich versetzte, transaktionskonsistente Kopie einer Produktionsdatenbank.

Es gibt 2 Typen von Standby Datenbanken:

- physikalische Standby Datenbank
- logische Standby Datenbank

Die **Eigenschaften der physikalischen Standby Datenbank** in Bezug zur primären Datenbank sind:

- identische Datenbankstrukturen auf Datenblockebene
- identische Datenbankschemata, Indizes usw.

Die Synchronisation erfolgt über das Recovern der Redo Log Informationen der primären Datenbank.

Die **Eigenschaften der logischen Standby Datenbank** in Bezug zur primären Datenbank sind:

- identische logische Informationen
- möglicherweise abweichende physikalische Struktur
- ggf. weitere Datenbankschemata

Die Synchronisation erfolgt über SQL-Statements, die aus Transformation der Redo Log Informationen der primären Datenbank gewonnen werden.

Besondere Eigenschaften einer logischen Standby Datenbank sind, dass Rechnerarchitektur, Betriebssystem und Oracle Software Version nicht identisch sein müssen. Damit wird die Migration auf ein anderes System, aber auch ein Upgrade von Betriebssystem und Oracle Software unterstützt, wie in diesem Projekt durchgeführt. Es gibt jedoch einige Einschränkungen in den Daten- und Tabellentypen und Befehlen der DDL und DML bei der Synchronisation der logischen Standby Datenbank.

⇒ Diese Beschränkungen mussten auch im Migrationsprojekt der Continentale bei einigen Anwendungsteilen berücksichtigt werden.

Eine zusätzliche Komponente ist der Data Guard Broker, dieser überwacht und steuert

- den Transport der Redo Log Informationen (Log Transport Services)
- das Nachfahren der Änderungen (Log Apply Services)
- den Rollenwechsel zwischen Primär und Sekundär Seite (Role Management Services)

Der Data Guard Broker wird bei der Continentale nicht eingesetzt.

## Maximum Availability Architecture (MAA)

Die Kombination von RAC und Data Guard bildet den Hauptbestandteil der so genannten Maximum Availability Architecture (MAA) bei Oracle.

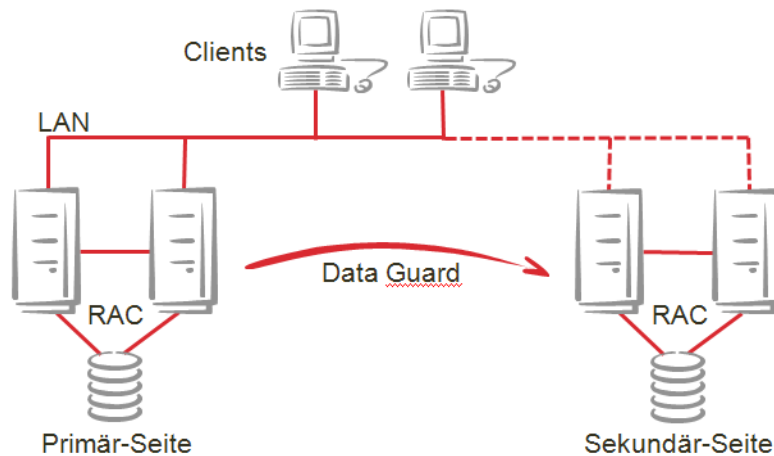


Abb. 4: Maximum Availability Architecture (MAA)

Dies ist die Konfiguration eines Datenbank-Systems, welche die maximale Verfügbarkeit bei geplanten und ungeplanten Ausfällen bietet und sie entspricht dem neuen Zielsystem bei der Continental (siehe Ausgangssituation).

## Migrationspfad: Data Guard Rolling Upgrade

An sich bezeichnet Rolling Update bzw. Rolling Upgrade eine Methode Software ohne Downtime aus Sicht des Anwenders zu aktualisieren.

- Die Anwendung ist durchgehend verfügbar!

Aber

- Einzelne Softwarekomponenten sind nacheinander nicht verfügbar.

Rolling Update bzw. Rolling Upgrade ist also nur im Hochverfügbarkeitsszenario möglich. Bei Oracle Data Guard ist ein Update bzw. Upgrade nicht gänzlich ohne Ausfall möglich. Zumindest im kurzen Zeitraum eines Switchover bzw. Failover von der Primär- auf die Sekundär-Seite ist die Datenbank für die Anwendung nicht verfügbar. Dennoch bezeichnet man auch dieses als Rolling Upgrade.

Im Groben lief das Rolling Upgrade über Data Guard für die drei nach 11g Rel.2 migrierten Datenbanken nach folgendem Vorgehen ab:

1. Der neue drei Knoten Extended Distance Cluster mit Grid Infrastruktur Software Version 11g Rel.2 wurde aufgesetzt.
2. Die Oracle Datenbank Software wurde sowohl in Version 10g Rel.2 als auch 11g Rel.2 auf diesem RAC installiert ohne eine Datenbank anzulegen.
3. Die Physical Standby Datenbank mit 10g Rel. 2 wurde ohne Produktionsunterbrechung über Restore mittels Recovery Manager aufgebaut.
4. Die Physical Standby Datenbank wurde in eine Logical Standby Datenbank umgewandelt.

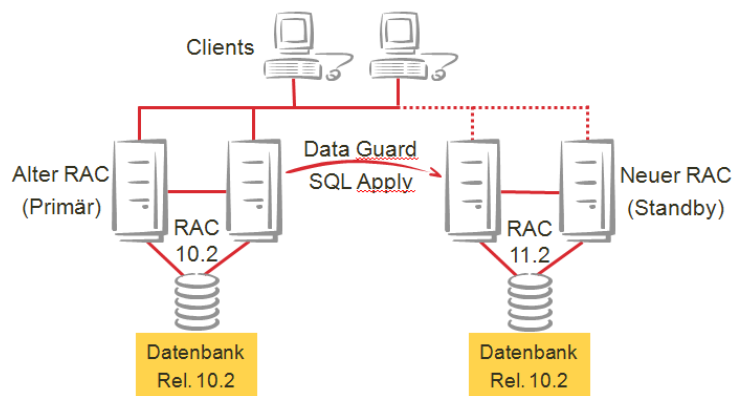


Abb. 5: Data Guard Konfiguration vor Upgrade

5. Das SQL Apply wurde ausgeschaltet.
6. Die Standby Datenbank wurde von 10g Rel.2 auf 11g Rel.2 upgegraded.

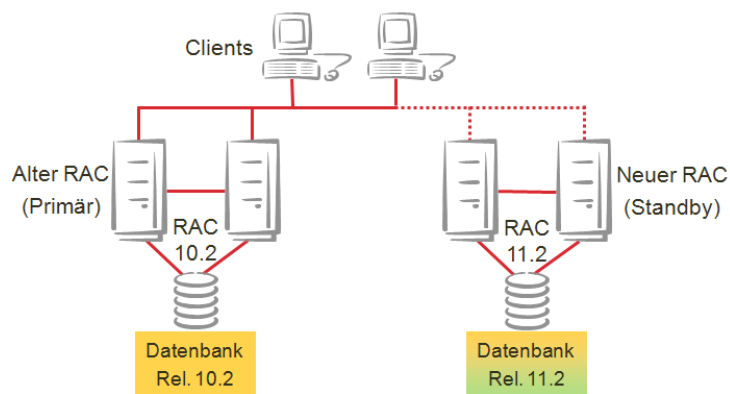


Abb. 6: Data Guard Konfiguration während Upgrade

7. Das SQL Apply wurde wieder eingeschaltet.
- Der Instanz Parameter COMPATIBLE beider Datenbanken stehen weiterhin noch identisch auf dem alten Versionsstand 10.2.

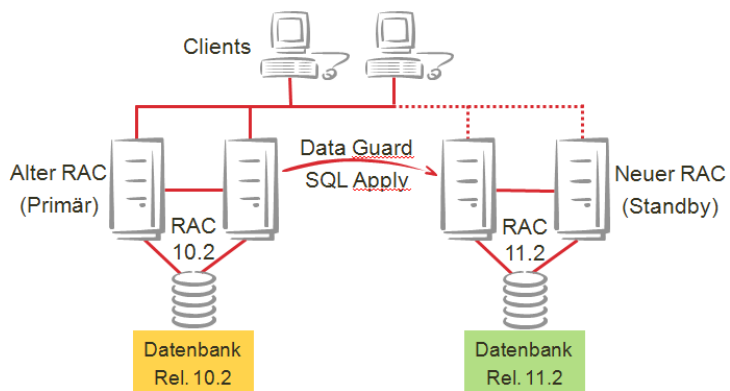


Abb. 7: Data Guard mit gemischtem Release

8. Zum Stichtag erfolgte ein bewusster Failover von der Primär-Datenbank auf die Logical Standby Datenbank.
9. Die Data Guard Konfiguration zwischen altem und neuem RAC wurde abgebaut bevor der neue RAC produktiv ging.
- Ein Redo Apply von der neuen Primärdatenbank mit höherer Version auf die alte Datenbank mit der alten Version ist nicht mehr möglich.

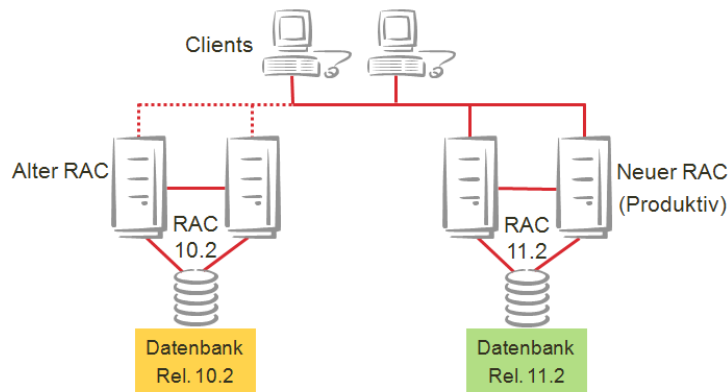


Abb. 8: Data Guard Konfiguration nach Failover

Für die Migration der Datenbank, die nicht auf 11g Rel.2 migriert wurde, sondern auch im neuen RAC mit dem Softwarestand 10g Rel.2 läuft, entfallen die Schritte 4 bis 7.

## Projektablauf

Das Migrations-Projekt bei der Continentale begann im Winter 2010 mit einer **Planungs- und Test-Phase:**

Febr. – Mai 2010:	Planung (Oracle-Version 11.2.0.1, 3 Knoten-RAC, ASM, Red Hat,...)
Juni – Aug. 2010:	Rolling-Upgrade-Tests auf einem "Spiel-System" ohne RAC
Juli 2010:	Installationen auf einem "Referenz-RAC" (Grid-Infrastruktur, DB-Software, ASM-Diskgruppen, Backup-Software)
Aug. 2010:	Installationen auf dem neue "Produktiv-RAC"
Sept. – Okt. 2010:	Rolling-Upgrade-Test auf "Referenz-RAC" mit Kopie einer Prod-DB
Dez. 2010:	Neuorientierung (TEAM, 11.2.0.2 statt 11.2.0.1, bei GI-Upgrade „Referenz-RAC“ zerstört)

Nachdem beim gescheiterten Upgrade-Versuch der Grid-Infrastruktur auf dem „Referenz-RAC“ das Referenz-System nicht mehr zu benutzen war, wurde entschieden, direkt in die **Produktiv-Phase** einzusteigen:

Jan. – Febr. 2011:	Rolling-Upgrade-Test mit allen Prod-DBs und dem neuen "Prod-RAC" (erst physical, dann logical Standby-DBs, dann 11g-Upgrade und Failover)
März 2011:	technische RAC-Streßtests auf dem neuen "Prod-RAC" (Kabel ziehen, ...)
März – Apr. 2011:	"Anwendungstests" mit den Prod-Kopien auf dem neuen "Prod-RAC"
Mai 2011:	Neuaufbau der Standby-DBs auf dem neuen "Prod-RAC", 11g-Upgrade
4.6.2011:	Failover auf das neue "Prod-RAC" inkl. Migration der Anwendungen

Insgesamt ergab sich bei der Migration folgender **Aufwand** an Hardware und Personal:

- Hardware: 3 "Spiel-PCs", 3-Knoten-RAC "PROD", 3-Knoten-RAC "REF"



- Personal intern: ca. 350 Mann-Tage (verteilt auf 4 Personen)
- Personal extern: 11 Tage Consultant-Leistung, 2 Tage Oracle Principal Sales Consultant

## Handicaps

Im Zuge der Tests und der produktiven Migration trat eine Reihe von Problemen auf. Zunächst gab es einige technische Probleme bei der Umsetzung der Rolling-Upgrade-Strategie.

### Prinzipielle Probleme mit dem **Upgrade-Konzept**:

- Online-Doku: Es gibt kein zusammenhängendes Konzept für unser Upgrade-Szenario.
- Switchover: Der meistens empfohlene Switchover funktioniert hier nicht (=> Failover).

### Probleme **nach der Installation** der Grid-Infrastruktur und der Oracle-Software:

- `STARTUP NOMOUT`: bricht für 10g ab mit `ORA-29702` (vgl. MOS notes 948456.1, 946332.1)  
Ursache: die Knoten sind *unpinned* gem. `olsnodes -t -n`  
Lösung: als root `crsctl pin css -n <node1> <node2> <node3>`
- viele Prozesse: `oraagent.bin` bleiben *inactive* hängen (vgl. Bugs 10299006, 11877079)  
Lösung: täglich mit `crontab`-Job killen

### Besonderheiten bei **Parametern** für die physical Standby-DB:

- `control_files`: sollte erst nach `DUPLICATE FOR STANDBY` gesetzt werden (wg. OMF)
- `db_create_file_dest`: sollte auf die Data-Diskgruppe gesetzt werden, z.B. = ``+DATA``
- `db_recovery_file_dest`: sollte auf die FRA-Diskgruppe gesetzt werden, z.B. = ``+FRA``
- `log_archive_dest_n`: nutze `use_db_recovery_file_dest` (für Registrierung in FRA)
- in RMAN: `configure archivelog deletion policy to applied on standby`
- `local_listener`: weglassen
- `remote_listener`: = ``<scan-Adresse>:1521`` und `EZCONNECT` in `sqlnet.ora`

### Probleme beim Aufbau der **physical Standby-DB**:

- `srvctl`: beachte den 10g/11g-Versionsunterschied bei `ADD DATABASE`
- in Primary: `backup current controlfile for standby` (statt Autobackup)  
anschließend: Tipps zu `add standby logfile` im Alertlog (Größe, Gesamtanzahl)
- beim Log-Apply: `ORA-00313`-Fehler kann man ignorieren (ab 10.2.0.4)

### Probleme bei der Transformation in eine **logical Standby-DB**:

- SQL-Apply: läuft nicht an mit `use_db_recovery_file_dest` in `log_archive_dest_n`  
Lösung: ersetze `use_db_recovery_file_dest` durch den Pfad bis nach dem Upgrade
- SQL-Apply: läuft erst an nach `alter system archive log current in Primary`
- Evtl. Fehler: `ORA-00332: archived log is too small - may be incompletely archived`  
Lösung: Archivelog manuell nach Standby kopieren, manuell `register logfile`

### Probleme nach dem **11g-Upgrade** in der logical Standby-DB:

- `oraenv`: 10g/11g-Versionsunterschied; dadurch `$ORACLE_BASE` evtl. nicht gesetzt
- Alertlog: dadurch evtl. unerwartet verlagert (`diagnostic_dest` nutzt `$ORACLE_BASE`)
- SQL-Apply: hing bzw. lief extrem langsam  
Lösung: Anzahl der Applier-Prozesse und `max_sga` erhöht, z.B.:  
`DBMS_LOGSTDBY.APPLY_SET('MAX_SERVERS', '128')`

```
DBMS_LOGSTDBY.APPLY_SET ('APPLY_SERVERS', '64')
DBMS_LOGSTDBY.APPLY_SET ('MAX_SGA', '1024')
```

Probleme beim **Failover** auf die Standby-DB:

- laufende DB-Inst.: beim Failover Anzahl in Primary-DB: alle, in Standby-DB: 1
- Services: dürfen zur Sicherheit nicht in der alten und neuen DB gleichzeitig laufen
- DB-Objekte: DB-Links, Directories (unverändert) evtl. manuell anpassen

Unabhängig von der technischen Migration der Datenbanken ergaben sich auch organisatorische Probleme bei der Migration der produktiven Anwendungen auf diese neuen Datenbanken.

Probleme bei der Organisation von **Anwendungstests** mit Prod-Kopien auf dem neuen "Prod-RAC":

- Kompetenz: Gibt es eine „Organisationseinheit“ für zentrale Tests?
- Anwendungen: Gibt es zentrale Dokumentationen aller Oracle-Anwendungen?
- Datasources: Gibt es zentrale Dokumentationen aller Datasources (enthalten HOST)?
- Services: Gibt es Anwendungen mit DB-Name/SID (ändern!) statt log. Services?

Probleme mit **Datasources** bei der Migration der Anwendungen nach dem Failover:

- HOST-Problem: Änderung zum Migrationszeitpunkt in allen Datasources zu aufwändig  
Lösung: bei Migration die „alten“ VIP-Adressen im DNS auf neuen HOST setzen  
Applicationserver: müssen dafür durchgestartet werden wg. DNS-Cache, Connection-Pooling

## Ausfallzeiten

Der Failover von den alten auf die neuen Datenbanken und die Migration der Anwendungen am 4. Juni verliefen reibungslos. Der Gesamtausfall der Anwendungen von ca. 3 Stunden war zu verkraften und vorher so auch im Hause angekündigt worden. Insgesamt hatten wir folgende Ausfallzeiten der einzelnen Komponenten:

- **DB:** ca. ½ h (Failover, Log-Kontrolle, Dataguard-Parameter abbauen, ...)
- **Services:** ca. 1 ½ h (Stop/Start DB, Init-Parameter, Sonderbehandlung bei 10g, ...)
- **Anwendungen:** ca. 3 h (Stop/Start der Applicationserver)

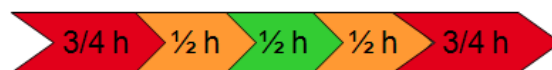


Abb. 9: Zeitstrahl Ausfallzeiten

## Quellen / Literatur

Oracle White Paper

- Database Rolling Upgrade Using Data Guard SQL Apply  
Oracle Database 11g and 10gR2  
<http://www.oracle.com/technetwork/database/features/availability/maa-wp-10gr2-rollingupgradebestprac-1-132006.pdf>
- Database Rolling Upgrade Using Transient Logical Standby:  
Oracle Data Guard 11g  
<http://www.oracle.com/technetwork/database/features/availability/maa-wp-11g-transientlogicalrollingu-1-131927.pdf>

- Upgrade Methods for Upgrading to Oracle Database 11g Release 2  
<http://www.oracle.com/technetwork/database/upgrade/11gr2-upgrade-methods-wp-2010-175757.pdf>

My Oracle Support (MOS) und Oracle Technology Network (OTN)

- MOS Dokument ID 300479.1  
10g Rolling Upgrades with Logical Standby
- MOS Dokument ID 785351.1  
Oracle 11gR2 Upgrade Companion
- MOS Dokument ID 338706.1  
Oracle Clusterware (formerly CRS) Rolling Upgrades
- OTN: SQL Apply Rolling Upgrade Best Practices

DOAG Konferenz Vorträge (2009)

- Inside 11g R2: *Minimal Downtime Upgrades - Mythen und Chancen*  
Mike Dietrich (Oracle)
- Inside 11.2: Datenbank-Upgrade - Tipps und Tricks und noch viel mehr  
Mike Dietrich (Oracle)
- Rolling Upgrade Verfahren mit der Oracle Datenbank 11g  
Ralf Appelbaum (TEAM)

**Kontaktadressen:**

**Ralf Appelbaum**

TEAM

Partner für Technologie und  
angewandte Methoden der  
Informationsverarbeitung GmbH  
Hermann-Löns-Str. 88  
D-33104 Paderborn

Telefon: +49 (0)5254 / 8008-0  
Fax: +49 (0)5254 / 8008-19  
E-Mail: [ra@team-pb.de](mailto:ra@team-pb.de)  
Internet: <http://www.team-pb.de>

**Dr. Andreas Fleige**

Abteilung IK3-DB

Die Continentale  
Continentale Krankenversicherung a.G.

Ruhrallee 92  
D-44139 Dortmund

Telefon: +49 (0)231 / 919-2808  
Fax: +49 (0)231 / 919-1267  
E-Mail: [andreas.fleige@continentale.de](mailto:andreas.fleige@continentale.de)  
Internet: <http://www.continentale.de/>