Die Herausforderung beim Versicherungsverbund Die Continentale bestand darin, einen Oracle 10*g* R2 Zwei-Knoten-„Extended-Distance"-RAC mit möglichst geringer Downtime auf einen 11*g* R2 Drei-Knoten-„Extended-Distance"-RAC zu migrieren. Die Migration der Datenbanken erfolgte als Rolling Upgrade über Data Guard mittels Logical Standby. Der Artikel beschreibt neben dem technischen Hintergrund auch den Projektablauf und insbesondere die kleineren und größeren Handicaps, die es zu bewältigen gab.

RAC-Migration von 10*g* R2 auf 11*g* R2 über Data Guard mit Handicaps

Dr. Andreas Fleige, Versicherungsverbund Die Continentale, und Ralf Appelbaum, TEAM GmbH

Ausgangssituation: Abgelöst werden sollte ein Oracle-Datenbank-Server mit folgenden Bestandteilen:

- Zwei Knoten „Extended Distance" RAC
- SuSE Linux Enterprise Server 10 (2.6.16) 64-bit
- Oracle Clusterware und Datenbank-Software, alle Version 10*g* R2 (10.2.0.4.0)
- Vier Datenbanken
- Daten-Dateien abgelegt in OCFS2-Devices

Das Zielsystem besteht aus diesen Komponenten:

- Zwei „Extended Distance" RAC á drei Knoten (siehe Abbildung 1)
- Spiegelung der Datenbanken über Data Guard zwischen den RACs
- Red Hat Enterprise Linux Server 5.6 (2.6.18) 64-bit
- Oracle Grid-Infrastruktur in der Version 11*g* R2
- Drei Datenbanken 11*g* R2 (11.2.0.2.0)
- Eine Datenbank 10*g* R2 (10.2.0.4.0)
- Daten-Dateien abgelegt im ASM
  - Jede Datenbank in eigener Daten- und FRA-Diskgruppe
  - Spiegelung (normale Redundanz) über Volumes aus zwei SANs

Bei der neuen Umgebung handelt es sich also um eine Maximum-Availability-Architecture-Umgebung (MAA, siehe Abbildung 2). Die Intention dieser Umgebung ist bei der Continentale aber weniger die Sicherheit im Desaster-Fall (die zwei RACs stehen in denselben Rechenzentren), sondern das Bestehen folgender Möglichkeiten:

- Aufbauen einer zum Produktiv-System identischen Test-Umgebung
- Den Datenbestand in Test-Datenbanken annähernd aktuell im Vergleich zu denen in Produktiv-Datenbanken halten
- Software-Updates (Patches und PSUs) über Data Guard Rolling Update/Upgrade einspielen:
  - Als zukünftiges strategisches und erprobtes Vorgehen
  - Mit möglichst geringer Downtime

Auswahl eines Migrationspfads

Ein Upgrade von einer früheren Oracle-Datenbank-Version auf 11*g* R2 kann auf folgenden Wegen stattfinden:

- Als „in-place"-Upgrade mittels Database Upgrade Assistant (DBUA) oder manuell über SQL-Skripte
- Über Export/Import beziehungsweise Data Pump
- Als Rolling Upgrade mittels Data Guard SQL Apply

Zwei weitere, selten genutzte Upgrade-Möglichkeiten über Oracle Streams und über Oracle Transportable Tablespaces wurden nicht berücksichtigt.

Ein Upgrade des bestehenden Systems von Oracle 10*g* R2 auf 11*g* R2 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 beziehungsweise Data Pump an einem Wochenende möglich gewesen. Weitere Kriterien zur Auswahl des Migrationsverfahrens waren jedoch:

- Eine möglichst geringe Downtime, aber mit ausreichend Pufferzeit zu erreichen
- Die Migration vorab mehrfach zu testen
- Die Anwendungen auf dem neuen System mit realistischem Datenbestand zu testen
- Die zentrale Strategie, zukünftige Updates/Upgrades mit diesem Upgrade einzuführen und zu testen

Damit stand fest, dass die Migration über Data Guard Rolling Upgrade erfolgen musste.

Dass eine der vier Datenbanken auch in der neuen Umgebung weiter unter der Version 10*g* R2 laufen musste und nicht auf 11*g* R2 aktualisiert wurde, hängt mit dem Oracle Warehouse Builder (OWB) zusammen. Dessen Kernfunktionalität ist seit 10*g* R2 Bestandteil der Datenbank und somit hätte ein Upgrade dieser Datenbank auch ein Upgrade der Anwendung auf den neuen OWB bedeutet, was mit einem sehr hohem Aufwand verbunden ist.

Oracle Extended Distance RAC

Über einen Oracle Real Application Cluster (RAC) ist die Laufzeit-Umgebung eines Datenbank-Systems, also die Instanz, gegen einen Server-Ausfall geschützt. Die Continentale nutzt dies darüber hinaus in der Konfiguration des Extended Distance RAC, um der Gefahr eines Desasters zu begegnen (siehe Abbildung 1).

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

In der Grid-Infrastruktur 11*g* R2 werden auch die Dateien der Clusterware, also Cluster-Registry und Voting-Disk, im ASM gespiegelt abgelegt. Bei Ausfall der Netzwerk-Verbindung zwischen den Rechenzentren sehen die RAC-Knoten jeweils die lokale Kopie der Voting-Disk und es entsteht eine sogenannte „Split Brain"-Situation, das heißt, es ist nicht möglich zu entscheiden, welcher der aktive RAC-Teil ist. Um diese Situation zu vermeiden, muss eine dritte Kopie der Voting-Disk außerhalb der beiden SANs und auch außerhalb der beiden Rechenzentren eingerichtet sein. Bei der Continentale läuft dazu ein NFS-Server in einer Außenstelle, auf dem eine Datei abgelegt ist, die als dritte Spiegelplatte in der ASM-Diskgruppe fungiert.

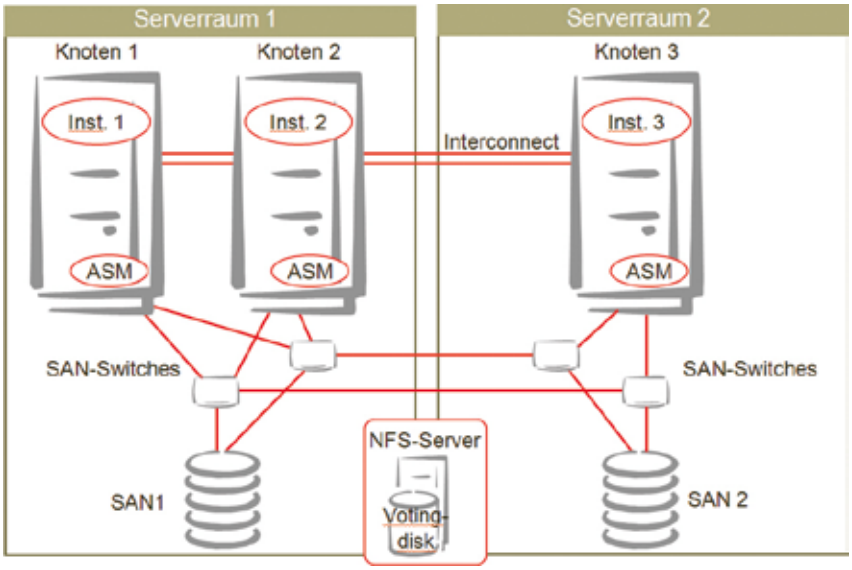

Abbildung 1: Extended Distance RAC

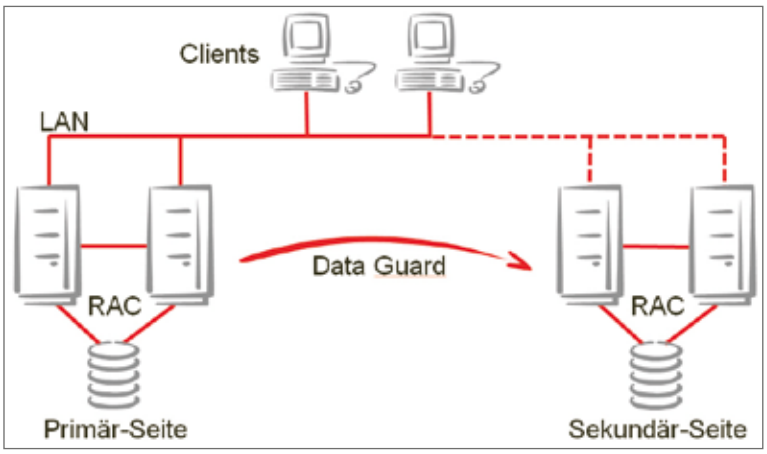

Abbildung 2: Maximum Availability Architecture (MAA)

Maximum Availability Architecture (MAA)

Die Kombination von RAC und Data Guard bildet den Hauptbestandteil der sogenannten „Maximum Availability Architecture" (MAA) bei Oracle (siehe Abbildung 2). 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 Continentale (siehe Ausgangssituation).

Wie bereits zuvor erwähnt, soll Data Guard bei der Continentale jedoch nicht die Sicherheit im Desaster-Fall sicherstellen, sondern kommt zum Einsatz, um Rolling Upgrades durchzuführen. Hierbei werden die besonderen Eigenschaften einer logischen Standby-Datenbank genutzt, wodurch Rechner-Architektur, Betriebssystem und Oracle-Software-Version zwischen Primär- und Standby-System 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 Tabellen-Typen 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. Es wurde organisatorisch sichergestellt, dass die betroffenen Anwendungsteile ab dem Zeitpunkt der Umwandlung in eine Logical-Standby-Datenbank bis nach der Migration nicht genutzt wurden.

Migrationspfad mit Data Guard Rolling Upgrade

An sich bezeichnet „Rolling Update" beziehungsweise „Rolling Upgrade" eine Methode, um Software ohne Downtime aus Sicht des Anwenders

zu aktualisieren. Die Anwendung ist dabei durchgehend verfügbar. Aber: Einzelne Softwarekomponenten sind nacheinander nicht verfügbar.

Rolling Update beziehungsweise Rolling Upgrade ist also nur im Hochverfügbarkeits-Szenario möglich. Bei Oracle Data Guard ist ein Update beziehungsweise Upgrade nicht gänzlich ohne Ausfall möglich. Zumindest im kurzen Zeitraum eines Switchover beziehungsweise 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 11*g* R2 migrierten Datenbanken nach folgendem Vorgehen ab:

1. Der neue Drei-Knoten-Extended-Distance-Cluster mit Grid-Infrastruktur-Software Version 11*g* R2 wurde aufgesetzt.
2. Die Oracle-Datenbank-Software wurde sowohl in Version 10*g* R2 als auch 11g R2 auf diesem RAC installiert, ohne eine Datenbank anzulegen.
3. Die Physical-Standby-Datenbank mit 10*g* R2 wurde ohne Produktionsunterbrechung über Restore mittels Recovery Manager aufgebaut.
4. Die Physical-Standby-Datenbank wurde in eine Logical-Standby-Datenbank umgewandelt (siehe Abbildung 3).
5. Das SQL-Apply wurde ausgeschaltet.
6. Die Standby-Datenbank wurde von 10g R2 auf 11*g* R2 umgestellt.
7. Das SQL-Apply wurde wieder eingeschaltet. Der Instanz-Parameter „COMPATIBLE“ beider Datenbanken steht weiterhin noch identisch auf dem alten Versionsstand 10.2 (siehe Abbildung 4).
8. Zum Stichtag erfolgte ein bewusster Failover von der Primär- auf die Logical-Standby-Datenbank.
9. Die Data-Guard-Konfiguration zwischen altem und neuem RAC wurde abgebaut, bevor der neue RAC in Betrieb genommen wurde.
10. Ein Redo Apply von der neuen Primär-Datenbank mit höherer Version auf die alte Datenbank mit der alten Version ist nicht mehr möglich (siehe Abbildung 5).

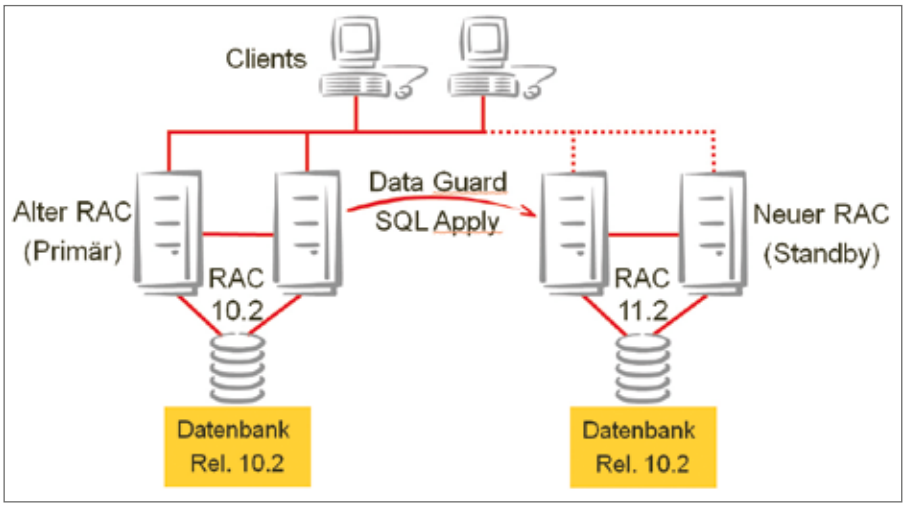

Abbildung 3: Data-Guard-Konfiguration vor dem Upgrade

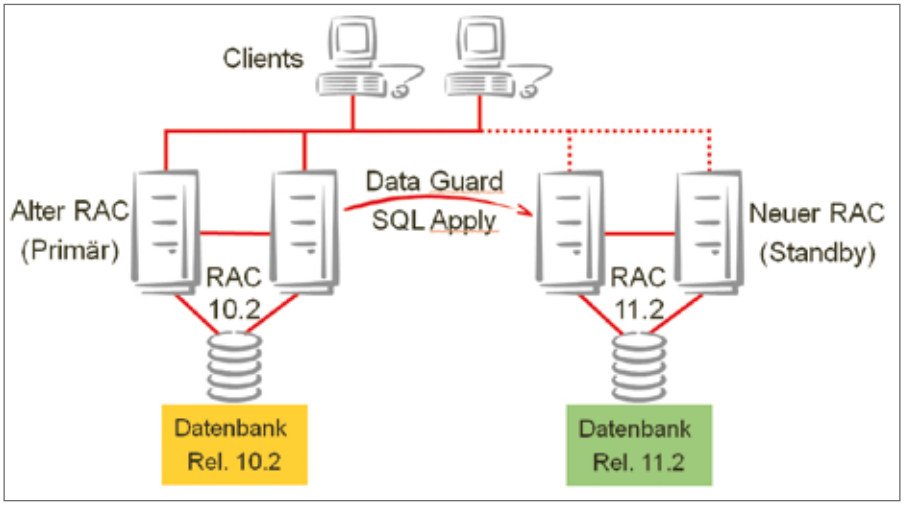

Abbildung 4: Data Guard mit gemischtem Release

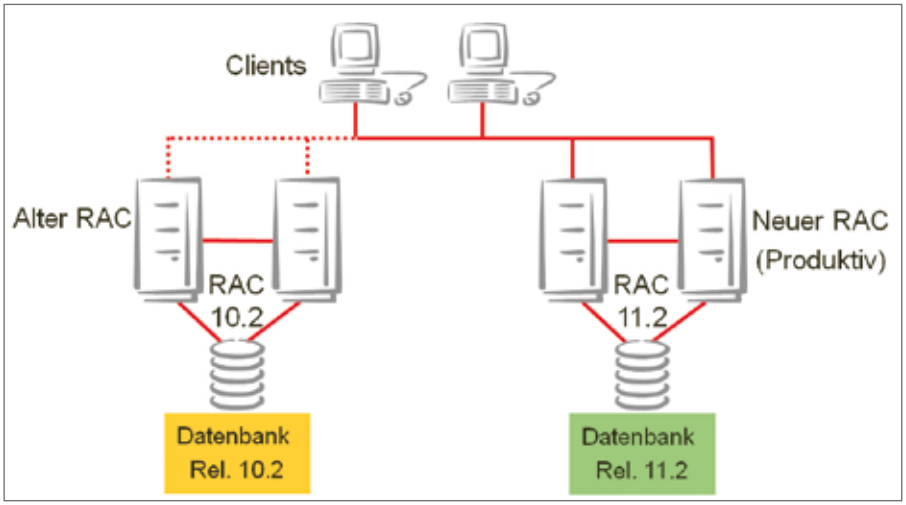

Abbildung 5: Data-Guard-Konfiguration nach dem Failover

Für die Migration der Datenbank, die nicht auf 11*g* R2 umgestellt wurde, sondern auch im neuen RAC mit dem

Softwarestand 10*g* R2 läuft, entfallen die Schritte 4 bis 7.

Projekt-Ablauf

Das Migrations-Projekt bei der Continentale begann mit einer Planungsphase, in der über das zukünftige Betriebssystem (Red Hat), die Oracle-Version (zunächst 11.2.0.1), die Anzahl der RAC-Knoten (drei), die Anzahl der CPU-Lizenzen (2+2+1) und über den Migrationspfad (Data Guard) entschieden wurde. Außerdem hat man das Storage-Management (ASM statt OCFS2), die Art des Mirroring (Normal Redundancy in ASM) und die Anzahl der Diskgruppen (zwei je Datenbank) festgelegt.

Die anschließende Testphase wurde im Sommer 2010 durch einen Rolling-Upgrade-Test mit Data Guard auf einem reinen „Spiel-System" auf PC-Basis ohne RAC eingeleitet. Parallel dazu konnten im Juli 2010 ein Drei-Knoten-„Referenz-Cluster" und dann im August auch das zukünftige Produktivsystem konfiguriert werden. Zur Konfiguration gehörten die Installation der Grid Infrastructure, der Datenbank-Software und der Backup-Software (NMO) sowie die Einrichtung der Diskgruppen. Anschließend wurde im September/Oktober mit einer Kopie einer produktiven Datenbank ein Rolling-Upgrade-Test auf das neue Referenz-System durchgeführt, wobei sich noch erhebliche technische Probleme offenbarten.

Das führte bei der Continentale zu einer Phase der Neubesinnung. Dabei wurde einerseits im Dezember 2010 die TEAM GmbH zur Unterstützung hinzugezogen und andererseits entschieden, auf die zwischenzeitlich erschienene Oracle-Version 11.2.0.2 umzuschwenken. Dazu musste insbesondere ein Upgrade der Grid Infrastructure durchgeführt werden, wobei durch eine Fehlbedienung der neue Referenz-Cluster so beschädigt wurde, dass er nicht ohne eine komplette Neuinstallation zu retten gewesen wäre.

Damit blieb nichts anderes übrig, als direkt in die Produktiv-Phase einzusteigen. Im Januar/Februar 2011 wurden dabei zu allen produktiven Datenbanken zunächst Physical- und dann Logical-Standby-Datenbanken auf dem neuen „Produktiv-Cluster" aufgebaut. Nach einem anschließenden Upgrade auf 11*g* (drei der vier Datenbanken) und einem Failover existierten damit Anfang März 2011 abgekoppelte Kopien der produktiven Datenbanken auf dem Zielsystem, wobei die produktiven Anwendungen noch weiterhin mit den alten Datenbanken arbeiteten. Mit diesen Kopien konnte man im März/April 2011 einige Tests auf dem zukünftigen „Produktiv-Cluster" durchführen. Zunächst erfolgte ein rein technischer „RAC-Stresstest" mit Ausschalten verschiedener RAC-Komponenten und anschließend ein stichpunktartiger Test der Anwendungen.

Anfang Mai 2011 begann dann die eigentliche heiße Phase der produktiven Migration mit dem Löschen der Kopien und dem Neuaufbau der Physical-Standby-Datenbanken zu den produktiven Datenbanken auf dem Zielsystem.

Für drei der vier Datenbanken wurde danach die Physical- in eine Logical-Standby-Datenbank umgewandelt und damit ein Upgrade auf Oracle 11.2.0.2 durchgeführt. Da anschließend der SQL-Apply weiterlief, konnte man die Aktionen am Umstellungstermin (Samstag, den 4. Juni 2011) im Wesentlichen auf einen Failover und eine Migration der Anwendungen beschränken. Dies ging verhältnismäßig reibungslos vonstatten, und auch die Anwendungen liefen danach relativ problemlos mit den neuen Datenbanken. Im Anschluss an die Migration wurden das defekte „Referenz-System" neu konfiguriert und dort Standby-Datenbanken zum neuen „Produktiv-System" aufgebaut.

Für die Continentale entstand bei diesem Projekt ein Aufwand von etwa 350 Mann-Tagen (im Wesentlichen verteilt auf vier Personen) und elf Tagen Consultant-Leistungen. Dabei ist in dem hohen Personal-Aufwand ein erheblicher Anteil an Know-how-Aufbau enthalten, da insbesondere ASM und Data Guard vorher fast gänzlich unbekannt waren.

Technische Handicaps

Im Zuge der Tests und der produktiven Migration trat eine Reihe von Problemen auf, die im Hinblick auf mögliche Nachahmer erwähnt werden sollen. Zunächst musste man leider feststellen, dass es schwierig war, eine geeignete Dokumentation für das gewählte Vorgehen zu finden. Insbesondere in der Online-Dokumentation von Oracle gibt es kein zusammenhängendes Konzept für dieses Upgrade-Szenario. Die Konzepte, die man findet, arbeiten meistens mit „Switchover" anstelle von „Failover". Es war allerdings eine relativ schmerzhafte Erfahrung, dass bei diesem Konzept ein „Switchover" zu einem Versionsabweichungs-Fehler führte.

Ebenso schmerzhaft war die Erfahrung, dass nach der Installation der Grid Infrastructure 11*g* R2 und der Datenbank-Software zunächst für eine 10g-Datenbank gar nichts funktionierte. So brach etwa ein „STARTUP NOMOUT" mit „ORA-29702" ab. Die Ursache dafür war, dass die Cluster-Knoten nach der Installation „UNPINNED" waren (siehe „olsnodes -t -n"). Dies ließ sich durch „crsctl pin css -n <node1> <node2> <node3>" als „root" beheben (siehe MOS-Notes 948456.1 und 946332.1). Der Hintergrund für dieses Verhalten ist das neue Konzept der „Server Pools" in 11*g* R2. Dabei sind die DB-Instanzen bei „policy managed" nicht mehr fest an die Cluster-Knoten gebunden. Für 10*g*-Datenbanken wird aber „administrator managed" benötigt.

Ein weiteres Problem nach der Installation bildeten „hängende", inaktive „oraagent.bin"-Prozesse in ASM (und später auch in der Datenbank), die zum Überlauf der maximalen Anzahl von Prozessen führten (siehe Bugs 10299006 und 11877079). Als Workaround wurden diese Prozesse in einem „crontab"-Job täglich gekillt.

Ferner wurde zunächst vergessen, „Huge Pages" einzurichten. Da auch der neue Parameter „memory_target" nicht genutzt werden sollte, ergaben sich dadurch im Test unakzeptabel lange Anmeldezeiten an die Datenbank.

Beim Aufbau der Physical-Standby-Datenbanken entstanden einige Schwierigkeiten durch Besonderheiten bei Parametern, die beispielsweise im ASM-Umfeld anders gesetzt werden müssen als in der Online-Dokumentation zu Data Guard beschrieben:

- „control_files“ sollte auf die Diskgruppen gesetzt sein, um Oracle-Managed-Files zu erhalten (zum Beispiel „=’+DATA’,’FRA’“).
- „db_create_file_dest“ sollte auf die Data-Diskgruppe gesetzt sein (zum Beispiel „=’+DATA’“).
- „db_recovery_file_dest“ sollte auf die FRA-Diskgruppe gesetzt sein (zum Beispiel „=’+FRA’“). Durch diese beiden „_dest“-Parameter bekommt man jeweils zwei Member pro Logfile-Gruppe bei einem „add logfile“-Kommando (ohne Pfad-Angabe).
- „log_archive_dest_n“ sollte „use_db_recovery_file_dest“ nutzen (was den Parameter „db_recovery_file_dest“ voraussetzt). Erst dadurch werden die Archivelogs in der FRA registriert.
- In RMAN sollte auf der Standby-Seite „archivelog deletion policy to applied on standby“ konfiguriert sein, um ein automatisches Löschen von übertragenen (und registrierten) Archivelogs zu erreichen.
- „local_listener“ kann man weglassen und „remote_listener“ sollte auf „<scan-Adresse>:1521“ gesetzt sein. Wenn man stattdessen diese Parameter aus der primären 10*g*-Datenbank unverändert übernimmt, kennt der Scan-Listener nur die lokalen Services.

Auch mit korrekten Parametern gab es noch Probleme beim Aufbau der Physical-Standby-Datenbanken, etwa durch die unterschiedlichen Versionen von „srvctl“ in den verschiedenen Oracle-Versionen. Während man dies durch den sorgfältigen Umgang mit „$ORACLE_HOME“ vermeiden konnte, war es ein Verständnisfehler zu glauben, dass man etwas mit dem Auto-Backup des Controlfiles der Primär-Datenbank anfangen könnte. Stattdessen musste das Controlfile in RMAN mit „backup current controlfile for standby“ gesichert werden, damit auf der Standby-Seite das RMAN-Kommando „duplicate target database for standby“ funktionierte. Nach dem Backup-Kommando findet man im Alertlog Hinweise auf die notwendige (Gesamt-) Anzahl und Größe der Standby-Logfiles, sodass man sich das Nachvollziehen entsprechender Formeln ersparen kann. Übrigens kann man ab 11*g* das „duplicate“-Kommando auch ohne Backups, also direkt aus der Primär-Datenbank durchführen. Für eine 10*g*-Datenbank funktioniert das allerdings noch nicht. Die kritischste Situation beim Aufbau der Physical-Standby-Datenbanken war stets die Frage, ob der Log-Apply auch wirklich anlief. Manchmal war dazu erst ein Log-Switch in der Primär-Datenbank notwendig, den man mit „alter system archive log current“ erzwingen konnte. Beim ersten Start des Log-Apply konnten Fehlermeldungen mit „ORA-00313“ ignoriert werden, die sich auf noch fehlende Online-Logs bezogen. Die fehlenden Online-Logs wurden beim Start des Log-Apply automatisch angelegt. Dies gilt zumindest ab Oracle 10.2.0.4.

Auch bei der Umwandlung der Physical- in eine Logical-Standby-Datenbank war der Start des SQL-Apply der kritische Augenblick. Insbesondere der oben empfohlene Wert „use_db_recovery_file_dest“ im Parameter „log_archive_dest_n“ verhinderte den Start des SQL-Apply. Daher musste hier doch temporär bis nach dem 11*g*-Upgrade der komplette Pfad des entsprechenden ASM-Verzeichnisses eingetragen werden. Nach einem Stromausfall lief der SQL-Apply nicht an, es gab den Fehler „ORA-00332: archived log is too small - may be incompletely archived”. Dieses Problem ließ sich dadurch beheben, dass das betroffene Archivelog manuell von der Primär- auf die Standby-Seite kopiert wurde (sogar nur ins File-System, nicht ins ASM) und dann mit „register logfile manuell“ nachregistriert wurde.

Nach dem 11*g*-Upgrade auf der Standby-Seite gab es zunächst Verwirrung dadurch, dass die Alertlogs nicht

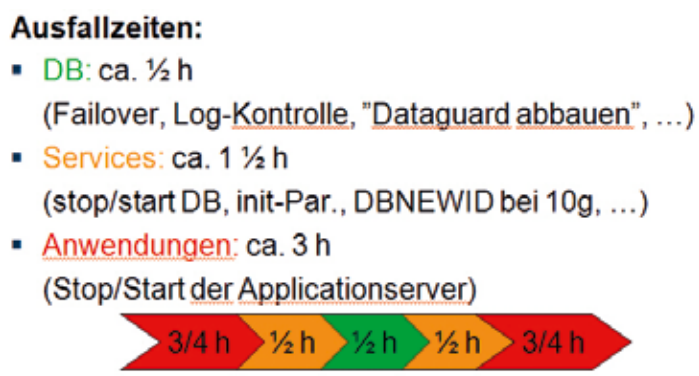

Abbildung 6: Zeitstrahl Ausfallzeiten

dort zu finden waren, wo sie vermutet wurden. Die Ursache war der Versionsunterschied der Prozedur „oraenv" in 10*g* / 11*g*, wodurch „$ORACLE_BASE" nicht gesetzt war und dadurch der neue Parameter „diagnostic_dest" einen unerwarteten Wert hatte. Etwas dramatischer war die Tatsache, dass der SQL-Apply nach dem Upgrade hing beziehungsweise Stunden für den Apply von wenigen Minuten Änderung in der Primär-Datenbank benötigte. Die Lösung war hier ein Erhöhen der Anzahl der Applier-Prozesse mittels „DBMS_LOGSTDBY.APPLY_SET(`MAX_SERVERS´,`128´)" und „DBMS_LOGSTDBY.APPLY_SET(`APPLY_SERVERS´,`64´)". Allerdings wiederholte sich das Problem schon kurz danach und die Anzahl der Applier-Prozesse musste sukzessive so lange erhöht werden, bis der aktuelle Zustand der Primär-Datenbank auf der Standby-Seite erreicht war.

Beim abschließenden Failover musste man beachten, dass für die Standby-Datenbank alle Instanzen bis auf eine heruntergefahren waren. Bei einem Test stellte sich heraus, dass die gleiche Maßnahme in der Primär-Datenbank für den Failover fatal war. Beim produktiven Failover musste natürlich sichergestellt sein, dass die Services, über die die Anwendungen auf die Datenbanken zugriffen, nicht gleichzeitig in den „alten" und den „neuen" Datenbanken zur Verfügung standen. Ferner mussten Datenbank-Links und Directories nach dem Failover noch manuell an die neuen Gegebenheiten angepasst werden.

Organisatorische Handicaps

Im Zusammenhang mit der Migration der Anwendungen traten nicht nur technische, sondern auch organisatorische Schwierigkeiten auf. So stellten sich bei der Organisation der stichpunktartigen Anwendungstests zunächst die Fragen, wer eigentlich für solche zentralen Tests verantwortlich war und ob es zentrale Dokumentationen aller Anwendungen gab, die die Oracle-Datenbanken benutzten. Eine weitere Schwierigkeit bestand darin, alle Data-Sources zu ermitteln, über die auf die Datenbanken zugegriffen wurde. Da die Data-Sources den „HOST"-Eintrag enthalten und dieser sich im Zuge der Migration änderte, hätten eigentlich alle Data-Sources zum Migrationszeitpunkt angepasst werden müssen. Dieses Vorgehen schied allerdings aus, da sich herausstellte, dass über hundert Data-Sources betroffen waren. Stattdessen wurden im DNS die „alten" VIP-Adressen beibehalten und zum Migrationszeitpunkt auf die neuen Rechner umgeleitet. Um den DNS-Cache beziehungsweise das Connection-Pooling auszuschalten, mussten dazu aber die Application-Server durchgestartet werden. Im Nachhinein konnten dann sukzessive alle „HOST"-Einträge angepasst werden. Einige wenige Data-Sources, die noch mit SID beziehungsweise DB-Namen konfiguriert waren, wurden noch vor dem Umstellungszeitpunkt auf einen „logischen" Service-Namen umgestellt, da sich die SID und der DB-Name bei der Migration änderten, während die Services unverändert übernommen werden konnten.

Failover

Insbesondere durch das Durchstarten der Application-Server und durch Sicherheitsmaßnahmen für die DB-Services gab es letztendlich doch einen relativ langen Ausfall der Anwendungen am Umstellungstermin, dem 4. Juni 2011. Während die Datenbanken durch die manuellen Tätigkeiten beim Failover nur für etwa eine halbe Stunde nicht zur Verfügung standen, fielen die Anwendungen insgesamt drei Stunden aus (siehe Abbildung 6). Dies war vorher jedoch auch so abgeschätzt und im Hause kommuniziert worden, sodass die Umstellung im geplanten Rahmen blieb.

Schließlich ist erwähnenswert, dass nach Aussage der Kollegen die Migration einen nicht unerheblichen Performance-Gewinn gebracht hat. Dies ist allerdings bisher nicht im Einzelnen durchgemessen worden.

Dr. Andreas Fleige
Versicherungsverbund
Die Continentale
andreas.fleige@continentale.de

Ralf Appelbaum
TEAM GmbH
ra@team-pb.de