

Verfügbarkeit und Service-Qualität sicherstellen

Patrick Schwanke, Quest Software GmbH

Eine Auszeit ist eine Auszeit, egal ob geplant oder ungeplant. Ungeplante Auszeiten lassen sich durch gute und schnelle Recovery-Mechanismen reduzieren. Geplante Auszeiten, wie sie bei Patches, Upgrades oder Migrationen entstehen, brauchen eine eigenständige Behandlung. In Oracle11g hat sich an dieser Stelle einiges getan, aber auch für ältere Versionen gibt es Hilfe. Die beste Verfügbarkeit nützt aber wenig, wenn die Antwortzeiten und damit die Servicequalität nicht stimmen.

Beginnend mit der Version 11g R2 sind Patchsets vollständige, eigenständige Installer – bei Patches und PSUs hat sich dagegen nichts geändert. Dies hat zwei wichtige Konsequenzen: Zum einen braucht bei Neuinstallationen nicht erst die Basisversion und dann das Patchset installiert zu werden, vielmehr reicht die direkte Installation des Patchsets. Dies gilt für alle Software-Komponenten, also insbesondere für die eigentliche Datenbank- sowie die Grid-Infrastructure-Software.

Zum anderen ist zum Einspielen eines Patchsets jetzt weniger Auszeit notwendig: Das Patchset wird im Hintergrund in ein neues Oracle-Home installiert. Erst nach Abschluss der Software-Installation müssen Datenbank-Instanz und Listener gestoppt werden. Genau wie in früheren Releases wird dann der Database Upgrade Assistant (DBUA) ausgeführt, um das Data-Dictionary-Upgrade durchzuführen. Nur dafür fällt also weiterhin eine Auszeit an, deren Länge vor allem von den installierten Komponenten abhängt (Oracle JVM, XML DB, Spatial, Text etc.). Als Konsequenz daraus lässt sich folgern:

- Datenbanken sollten nicht mit unnötigen Komponenten installiert werden. Insbesondere bei Benutzung des DBCA sind viele sogenannte „Standardkomponenten“ vorausgewählt. Auch die vorgefertigten Datenbanken des DBCA sind zwar beim Anlegen sehr schnell und bequem, enthalten aber sämtliche optionalen Komponenten und verur-

sachen somit bei jedem zukünftigen Patchset unnötig lange Auszeiten.

- Ab 11g R2 ändert sich mit jedem Einspielen eines Patchsets das Datenbank-Home beziehungsweise Grid-Infrastructure-Home. Der Autor schlägt daher vor, dass Oracle-Homes zukünftig die vierte Ziffer der Version enthalten, um ein konsistentes Namensschema zu bieten, wie `ORACLE_HOME=/u01/app/oracle/product/11.2.0.2/dbhome_1`.
- Das alte Home kann zwar nach dem Upgrade aller Datenbanken durch Aufruf des Deinstallers im Deinstall-Verzeichnis des zu löschenden Homes entfernt werden, temporär sind aber in jedem Fall altes und neues Homeparallel erforderlich. Die Installations-Mountpoints müssen also entsprechend dimensioniert sein.

Für das Grid-Infrastructure-Home ist die Out-Of-Place-Methode beim Einspielen von Patchsets die einzige Möglichkeit. Für Datenbank-Homes ist auch eine In-Place-Variante möglich, die aber von Oracle nicht empfohlen wird.

Auch das Betriebssystem will gepflegt sein

Das Einspielen von Betriebssystem-Updates oder -Patches sowie Hardware-Veränderungen erfordert häufig einen Neustart des Datenbank-Servers. In RAC-Umgebungen lässt sich dies schon länger umgehen, indem eine Rolling-Methodik benutzt, also jeder Knoten einzeln abgearbeitet wird. Zu

jedem Zeitpunkt stehen also alle Instanzen bis auf eine zur Verfügung.

Ist einem die RAC-Lizenz zu teuer und kann man auf die Skalierungsfähigkeiten eines RAC-Clusters verzichten, stellt das mit der Version 11g R2 eingeführte „RAC One Node“ eine lizenzkostengünstigere Alternative dar. Im Gegensatz zu einem vollen RAC kann eine RAC-One-Node-Datenbank zu einem Zeitpunkt immer nur eine einzige aktive Instanz haben. Lastverteilung ist damit offenbar nicht möglich, aber mithilfe der sogenannten „OMotion“-Technik kann diese Instanz von einem Cluster-Knoten auf einen anderen Knoten umgezogen werden – die Parallele zu VMwares „vMotion“ ist unübersehbar. Auf dem alten Knoten läuft die Instanz zunächst weiter, akzeptiert aber keine neuen Connections. Neue Connects gehen ausschließlich auf die neue Instanz. Nach einem einstellbaren Timeout (maximal 12 Stunden) wird die alte Instanz hart beendet (SHUTDOWN ABORT); spätestens jetzt müssen sich Anwendungen mit langlebigen Connects neu anmelden.

Hat man auf diese Weise alle aktiven Instanzen von einem Knoten weggeschoben (in VMware-Sprache: den Knoten „evakuiert“), kann man entsprechende Wartungsarbeiten durchführen, den Knoten durchstarten und anschließend per OMotion wieder mit Instanzen „bevölkern“ (siehe Abbildung 1).

Soll ein Knoten nie mehr als eine Datenbank beherbergen, benötigt man für einen RAC-One-Node-Cluster also

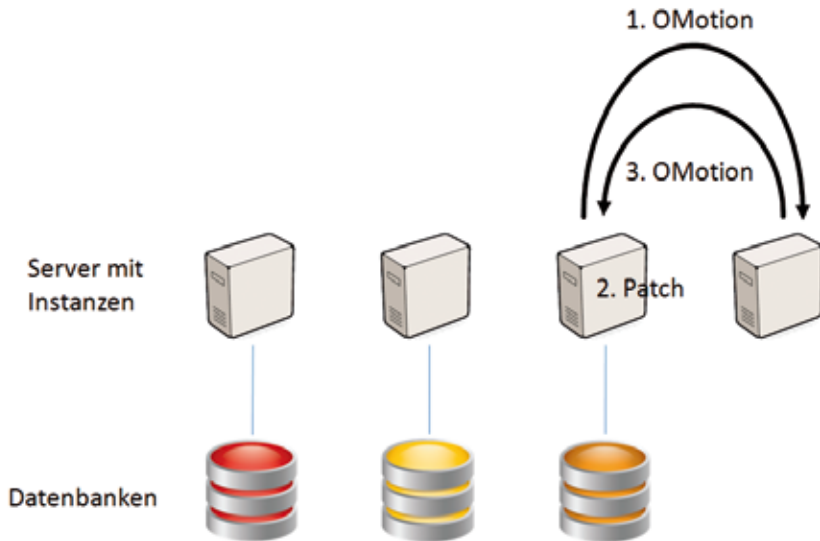


Abbildung 1: Betriebssystem-Upgrade oder -Patch in einem RAC-One-Node-Cluster

mindestens so viele Knoten wie beteiligte Datenbanken plus einen Ausweichknoten. Es ist aber wie bei einem normalen RAC auch möglich, Instanzen verschiedener Datenbanken auf demselben Knoten laufen zu lassen, was die benötigte Knotenzahl reduziert.

Auch harte Nüsse lassen sich knacken

Komplexere Umstellungen wie beispielsweise Server-, Plattform-, Storage-, Zeichensatzmigrationen oder Kombinationen hiervon sind eine Herausforderung, wenn es um die Reduktion von Auszeiten geht. Hier hat man es generell mit einem Trade-Off zwischen Auszeit und Risiko einerseits sowie Komplexität und Gesamtdauer der Migration andererseits zu tun: Simple Verfahren wie exp/imp, Datapump oder Transportable Tablespace benötigen entweder sehr lange Auszeiten oder haben diverse Einschränkungen.

Im Folgenden werden zwei Verfahren skizziert, die eine höhere Komplexität mit sich bringen, im Gegenzug aber für Auszeiten im Sekunden- bis Minutenbereich sowie minimales Risiko einer solchen Migration sorgen: „Rolling Upgrades“ und „Rolling Migrations“.

Während Rolling Upgrades ab der Version 10.1.0.3 und Enterprise Edition immerhin Versionswechsel sowie ab der Version 11g auch bestimmte

Plattformwechsel zwischen Windows und Linux zulassen (siehe My Oracle Support Doc ID 1085687.1), können Rolling Migrations für nahezu beliebige, auch kombinierte Upgrades und Migrationen eingesetzt werden. Quest Shareplex bietet diese Möglichkeit bereits seit über 10 Jahren an, unabhängig von den eingesetzten Datenbank-Versionen oder -Editionen.

In jedem Falle wird eine zweite, neue Datenbank aufgebaut: Bei Rolling Upgrade ist dies eine Logical-Standby-Datenbank, bei einer Rolling Migra-

tion wird die neue Datenbank von Null auf neu erstellt (z.B. mit dem Database Configuration Assistant DBCA) und anschließend per konsistentem Datapump oder exp/imp befüllt, wodurch gewissermaßen „en passant“ auch eine Reorganisation stattfindet

Anschließend werden laufende Änderungen von der bestehenden auf die neue Datenbank repliziert, bei Rolling Upgrade mittels Logical-Standby-Replikation, bei Rolling Migration durch Rekonstruktion der SQL-Kommandos und Transaktionen aus den Redolog-Daten und Nachspielen dieser Transaktionen auf der neuen Datenbank.

Bei Rolling Upgrade erfolgt nun das Upgrade der neuen Datenbank auf die gewünschte Version, bei Rolling Migration ist nichts mehr zu tun, da die Datenbank bereits mit der gewünschten Version erstellt wurde. Bis hierhin ist keinerlei Auszeit für die produktive Datenbank aufgetreten. Damit ist alles vorbereitet, und im Rahmen einer kurzen Auszeit können die Benutzer von der bestehenden auf die neue Datenbank umgebogen werden.

Während dieses Rollentausches von alter und neuer Datenbank lässt sich in beiden Szenarien die Replikationsrichtung drehen, sodass die alte Datenbank für einen gewissen Zeitraum weiterhin mit gepflegt wird. Bei unvor-

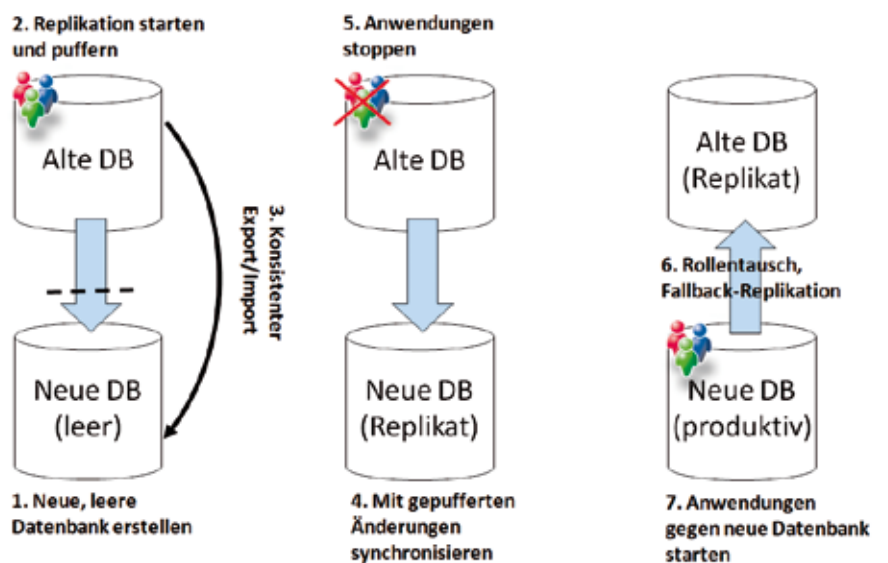


Abbildung 2: Ablauf Rolling Migration am Beispiel von Quest Shareplex

hergesehenen Problemen ist so ein erneuter Umzug der Benutzer auf die alte Datenbank möglich, ohne dass zwischenzeitliche Datenänderungen verlorengehen (siehe Abbildung 2).

Um das Migrationsrisiko weiter zu reduzieren, kann man vor der Umstellung auch die Replikation vorübergehend aussetzen (wobei die Änderungen nicht verlorengehen, sondern lediglich gepuffert werden) und auf der neuen Datenbank einen sogenannten „Restore-Point“ setzen:

```
SQL> CREATE RESTORE POINT vor_test;
```

Anschließend lassen sich beliebige Testläufe durchführen (funktionale Tests, Datenstichproben, Performance-Tests, Benchmarks etc.). Nach Abschluss der Tests fällt man mit dem Flashback-Datenbank-Kommando auf den Snapshot zurück und gibt die Replikation wieder frei, sodass die gepufferten Änderungen nachgefahren werden.

```
SQL> FLASHBACK DATABASE TO RESTORE POINT vor_test;
```

Damit das funktioniert, muss zumindest die neue Datenbank als Enterprise Edition lizenziert sein und – nur dort – das sogenannte „Flashback-Logging“ einmalig aktiviert werden:

```
SQL> STARTUP MOUNT
SQL> ALTER SYSTEM SET db_flashback_retention_target
                        = <max
                        testdauer in minuten>;
SQL> ALTER DATABASE ARCHIVELOG;
SQL> ALTER DATABASE FLASHBACK ON;
SQL> ALTER DATABASE OPEN;
```

Im Prinzip verfügbar ...

Eine formelle Verfügbarkeit nützt wenig, wenn die Antwortzeiten schlecht sind. Oftmals ist „Service-Qualität“ der bessere Begriff, da er diesen Zusam-

menhang berücksichtigt. Sind grundlegende Tuning-Möglichkeiten ausgereizt, lassen sich bessere Antwortzeiten auch durch ein Scale-Out erreichen. Einschlägige Hilfe bieten hier Technologien wie RAC oder auch die Auslagerung bestimmter Arbeiten auf Schatten-Datenbanken, etwa Reporting-Jobs oder Echtzeitanalysen (siehe Abbildung 3).

Ein Topf, mehrere Deckel

RAC stellt bekanntlich den Zugriff auf eine einzige Datenbank durch mehrere Instanzen dar. Im Optimalfall findet eine gleichmäßige Lastverteilung über die Instanzen statt. Antwortzeiten und damit die Servicequalität können sich also mit RAC verbessern, sofern die Arbeit über mehrere Instanzen verteilt werden kann.

Dies ist desto eher der Fall, je mehr es sich um CPU- oder Memory-lastige Arbeiten handelt, insbesondere bei PL/SQL-intensiven Anwendungen.

pitss Die Oracle FORMS Spezialisten!

Sie sehen den Anforderungen von Morgen bereits heute ins Auge?

... Wir bieten Ihnen den Übergang in die Technologien von Übermorgen!

Services: Multi-Language, Source Control, Oracle / Non-Oracle Objects, Maintenance / Development, Application & Database Change, Zooming, Screen Design, Visual Design, Application Impact, Integration, Application Analyse, Application Engineering, Business Logic (BL), Data Access Layer (DAL), Migration APEX, Migration ADF / JEE, PL/SQL2.java, Upgrade Forms / Reports 3, 4, 5, 6, 10, 11g, Dynamic Translation, Dictionaries.

Technical Capabilities: Documentation, Specification, Dependenc., Null Objects, Unused Obj., Not Trigg., Not Tech. Rel., Red. Obj., Red. Source, Comparision, DA-Layer, BL-Layer, DBLA, WEBServ.

Migration Process: Analysis, Dead Code, Redundancy, BL 2 DB, Quality Assurance, Documentation, Iteration, Co-Existence of Technologies/ Partial Projects.

Products: Forms 11g, ADF, APEX, New App.

ORACLE Gold Partner | www.pitss.com

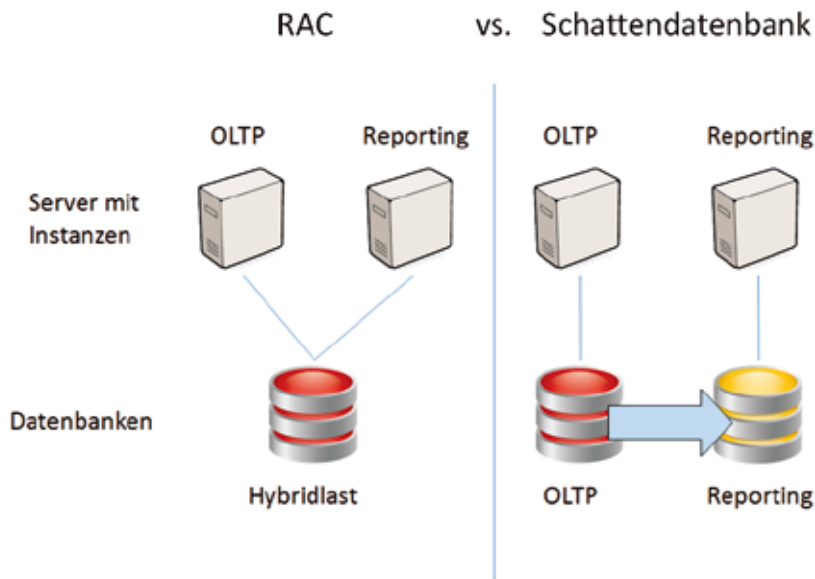


Abbildung 3: Lastverteilung mit RAC bzw. Schatten-Datenbanken

Handelt es sich um I/O-lastige Aufgaben wie beispielsweise umfangreiche Reporting- oder Batchläufe oder OLAP-Last, bringt RAC nur Vorteile, solange das Storage-System mit seiner I/O-Kapazität nicht zum Flaschenhals wird.

Stößt das Storage-System an die Grenzen seiner I/O-Kapazität, ist die erste Idee natürlich eine Erweiterung der I/O-Kapazität durch Hinzunehmen neuer Platten, was insbesondere die Plattenhersteller freut. Zu bedenken ist aber, dass – solange die I/O-Kapazitäten nicht sehr großzügig überdimensioniert werden – jeder Reporting-Job die Antwortzeiten verschlechtert. Dies führt oftmals zu dem Phänomen, dass ein System zu gewissen Zeiten erträgliche Antwortzeiten liefert, jedoch zu bestimmten „Stoßzeiten“ wie Monats-, Quartals-, Geschäftsjahresende, Ferienzeit, Weihnachtsgeschäft oder bei adhoc angeforderten Berichten in die Knie geht.

Mehrere Töpfe

Endgültige Abhilfe schafft oftmals eine Schatten-Datenbank, die als unabhängiger Datentopf, auf separaten Platten liegend, für Reporting- oder Analysezwecke verwendet wird. Wichtige Kriterien für das weitere Vorgehen

sind dann, wie aktuell diese dedizierte Datenbank sein muss, ob sie für Reporting-Zwecke optimiert sein soll und – natürlich – wie viel das Ganze kosten darf.

Am einfachsten und kostengünstigsten ist eine regelmäßige Aktualisierung (etwa auf Tagesbasis) mittels Datenexport/-import, per Kloning aus einem Backup oder als Physical-Standby-Datenbank. Alle diese Techniken sind im Übrigen auch mit einer Oracle Standard Edition verwendbar.

Ist Datenaktualität entscheidend, muss auf andere Techniken ausgewichen werden, die mittels Replikation mehr oder weniger eine Echtzeitkopie bereitstellen. Dies kann eine Logical-Standby-Datenbank sein, die allerdings Oracle Enterprise Edition voraussetzt, sich in der Praxis als schwer handhabbar herausgestellt hat und einige Einschränkungen zum Beispiel hinsichtlich unterstützter Datentypen mitbringt (die View DBA_LOGSTDBY_UNsupported gibt hier genauere Auskunft). Seit der Version 11g R1 gibt es außerdem das sogenannte „Active Data Guard“. Dies entspricht einer Physical-Standby-Datenbank, die gleichzeitig repliziert und für Read-Only-Betrieb genutzt werden kann. Hier fallen lizenztechnisch die Kosten einer

Enterprise Edition plus der Active-Data-Guard-Option an.

Replikationstechniken wie Quest Shareplex bieten zum Preis zusätzlicher Lizenzkosten die größte Flexibilität. Die Schatten-Datenbank wird in Echtzeit gepflegt, kann für Reporting-Zwecke optimiert sein, beispielsweise durch spezielle Indizes, kann auf einer anderen (kostengünstigen) Plattform liegen und ist Read-Write geöffnet. Letzteres ist nicht zu vernachlässigen, da viele Reporting-Anwendungen eben doch nicht vollständig „read-only“ sind, sondern einige Meta-Informationen abspeichern müssen oder Zwischenergebnisse in Staging-Tabellen / temporären Tabellen ablegen. Shareplex arbeitet auch hier unabhängig von der Oracle-Version und Edition.

Fazit

Oracle 11g R2 wartet mit einigen Neuerungen auf, die geplante Auszeiten für das Einspielen von Betriebssystem-Patches oder Oracle-Patchsets spürbar reduzieren. Die neue Methodik der Out-Of-Place-Patchsets sowie „RAC One Node“ als kleiner Bruder des altbekannten RAC sind hier zu nennen. Für komplexere Umstellungen oder Migrationen der Datenbank stehen schon seit Längerem Werkzeuge bereit, um Auszeit und Risiko zu minimieren. Hierunter fallen sogenannte „Rolling-Upgrade-“ und „Rolling-Migrations-Verfahren“, die oben genauer erläutert wurden.

Nicht minder wichtig als die reine Verfügbarkeit ist aber auch die Service-Qualität. Neben Tuning sowie dem Aufstocken der Hardware sind hier als einschlägige Techniken RAC sowie Schatten-Datenbanken zu nennen, um eine bessere Lastverteilung und damit eine höhere Service-Qualität gewährleisten zu können.

Kontakt:

Patrick Schwanke
patrick.schwanke@quest.com