



Christian Trieb
Leiter der Datenbank
Community

Liebe Mitglieder der Deutschen Oracle-Anwendergruppe,
liebe Leserinnen und Leser,

das Jahr beginnt mit Erwartungen an die neue Version der Oracle-Datenbank. Wann wird sie kommen? Wahrscheinlich dieses Jahr. Aber noch wichtiger ist, was die Version Neues bringen wird. Darüber zu spekulieren, ist nicht Bestandteil dieser Ausgabe der DOAG News, wir beschäftigen uns diesmal damit, wie die Datenbank optimal genutzt werden kann.

So wird das Thema „RAC und Hochverfügbarkeit“ auch mithilfe von Virtualisierung beschrieben. Außerdem wird die Migration von Oracle-Datenbanken unter verschiedenen Gesichtspunkten dargestellt. Performance-Aspekte der Datenbank und die unterschiedlichen Optionen kommen ebenfalls zur Sprache.

Aber auch für Datenbank-Entwickler bietet diese Ausgabe einige Artikel, etwa zu den Dauerbrennern „Apex“ und „Oracle Forms“. Zudem spielt das Thema „Data Warehouse“ eine Rolle und wird aus Datenbank-Sicht betrachtet.

Alle diese Artikel beziehen sich auf das aktuelle Release der Oracle-Datenbank. Was die neue Version bringen wird, erfahren Sie in einer späteren Ausgabe.

In diesem Sinne wünsche ich Ihnen viel Spaß und viele neue Informationen und Anregungen beim Lesen der aktuellen DOAG News.

Ihr



Schneller zum

Wesentlichen!

Einfach, verständlich, vollständig: Mit HUNKLER machen Sie Business Intelligence vom Start weg richtig.

- Integrierte, optimal abgestimmte Komplettlösungen für jeden Bedarf
- Zielgruppengenaue Reportvorlagen
- Robuste Technologiebasis (z. B. Oracle BI Server, Oracle Data Integrator)
- Stark verkürzte Projektzeiten
- Flexibel, skalierbar, investitionssicher
- Spezielle Lösung für SAP R/3
- Kooperation mit SAP-Spezialist NewFrontiers (www.newfrontiers.com)

ORACLE Platinum Partner

Partner von
NewFrontiers
10 Years!

Best Solutions Based on Oracle
HUNKLER
GmbH & Co. KG

Hauptsitz Karlsruhe
Geschäftsstelle Bodensee

Bannwaldallee 32
Fritz-Reichle-Ring 2

76185 Karlsruhe
78315 Radolfzell

Tel. 0721-490 16-0
Tel. 07732-939 14-00

Fax 0721-490 16-29
Fax 07732-939 14-04

info@hunkler.de
www.hunkler.de

- 3 Editorial
- 5 Spotlight
- 6 Pressespiegel
- 7 „Das ist immer eine Abwägung zwischen Chancen und Risiken ...“
Interview mit Klaus Vitts,
Leiter Datenbank-Team Bundesagentur
für Arbeit

Datenbank

- 10 Hochverfügbar durch Virtualisierung –
(wie) geht das?
Patrick Schwanke
- 14 Hora, die 10. – Kontinuität und Innovation
Günter Unbescheid
- 18 Realtime Data Warehouse: Auslagern
und Wiederherstellen operativer Daten
Steffen Kantorek
- 21 RAC-Migration von 10g R2 auf 11g R2
über Data Guard mit Handicaps
Dr. Andreas Fleige und Ralf Appelbaum
- 28 Oracle Database Appliance –
Überblick und erste Praxis-Erfahrungen
Hinrich Tobaben
- 31 Designing for Performance
Christian Antognini
- 34 Migrationsmethoden –
von ASM bis Zeichensatz
Patrick Schwanke

Entwicklung

- 38 Erfahrungen bei der Migration
von Forms/Reports 6i nach
Forms/Reports 11.1.2 und 12
Frank Christian Hoffmann
- 43 Edition-Based Redefinition
Christian Antognini

Best Practice

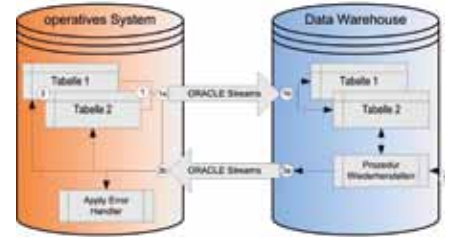
- 50 Vorsicht bei parallelen Abfragen
eines Oracle RAC auf
Multithreading-Chipsätzen
Piotr Sajda
- 56 Effizientes und einfaches Staging
in „Near Real Time“-Data-
Warehouses
Mathias Zarick und Karol Hajdu

Tipps und Tricks

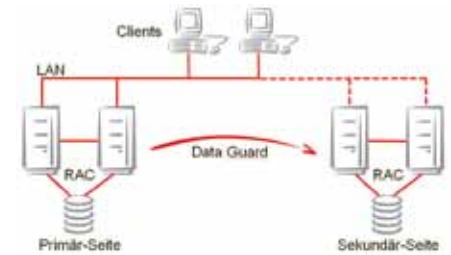
- 61 Heute: Generisches List-Item,
Teil 2
Gerd Volberg

Aus der DOAG

- 33 Vorschau
- 42 Wir begrüßen unsere neuen Mitglieder
- 54 Inserentenverzeichnis
- 61 Impressum
- 62 Die DOAG veranstaltet die größte
Oracle-Konferenz in Europa
- 64 Neuigkeiten aus dem Verein
- 66 DOAG-Termine



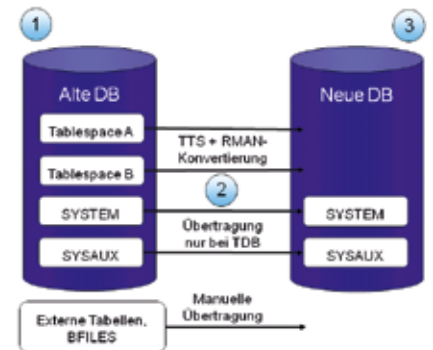
Realtime Data Warehouse: Auslagern und Wiederherstellen operativer Daten, Seite 18



RAC-Migration von 10g R2 auf 11g R2 über Data Guard mit Handicaps, Seite 21



Neu: Oracle Database Appliance – Überblick und erste Praxis-Erfahrungen, Seite 28



Migrationsmethoden – von ASM bis Zeichensatz, Seite 34



Dienstag, 15. November 2011

Die DOAG 2011 Konferenz + Ausstellung stellt in Verbindung mit dem Enterprise-Manager-Launch einen neuen Teilnehmer-Rekord auf. Auch seitens der Aussteller gibt es viele positive Kommentare (siehe Seite 62).

Dienstag, 15. November 2011

Die Mitgliederversammlung wählt turnusgemäß den neuen Vorstand. Er setzt sich zusammen aus Dr. Dietmar Neugebauer (Vorsitzender), Franz Hüll (Stellvertreter), Michael Paege (Stellvertreter), Ralf Kölling und Fried Saacke. Die fünf Vorstände bilden zusammen mit den vier Leitern der Communities das DOAG-Leitungsteam.

Mittwoch, 16. November 2011

Mylène Diacquenod, Online-Redakteurin der DOAG, führt im Rahmen der DOAG 2011 Konferenz + Ausstellung mehr als fünfzig Interviews mit wichtigen Personen aus der Oracle-Community. Die Aufzeichnungen sind unter „DOAG.tv“ auf den DOAG-Webseiten zu sehen.

Mittwoch, 16. November 2011

Zahlreiche Referenten aus den USA sprechen die DOAG-Verantwortlichen darauf an, dass sie von dem am Tag zuvor im Rahmen der DOAG 2011 Konferenz + Ausstellung stattgefundenen Speaker-Event begeistert waren. Die DOAG pflegt damit ihre guten Beziehungen zu den Oracle Headquarters.

Freitag, 18. November 2011

Die Mitgliederversammlung des Interessenverbands der Java User Groups e.V. (iJUG) hat ebenfalls einen neuen Vorstand. Sowohl der Vorsitzende Fried Saacke (DOAG), als auch seine beiden Stellvertreter Tobias Frech (Java User Group Stuttgart e.V.) und Stefan Koospal (Sun User Group Deutschland e.V.) sind einstimmig gewählt.

Freitag, 9. Dezember 2011

In seiner ersten konstituierenden Sitzung legte das DOAG-Leitungsteam, bestehend aus den neu gewählten Vorständen und den Leitern der vier Communities, die Zuständigkeiten neu fest. Zudem werden traditionell erste Schritte definiert, um die Ziele für die Legislaturperiode zu erreichen (siehe Seite 64).

Freitag, 23. Dezember 2011

Am letzten Arbeitstag des Jahres in der Geschäftsstelle blickt Fried Saacke, DOAG-Vorstand und Geschäftsführer, auf ein wirtschaftlich erfolgreiches Jahr zurück und bedankt sich beim Team für die hervorragende Arbeit.

Sonntag, 1. Januar 2012

Die DOAG wünscht allen Interessenten und Mitgliedern ein gutes und erfolgreiches neues Jahr.

Freitag, 13. Januar 2012

Die Datenbank Community startet das neue Jahr mit einer neuen Veranstaltungsreihe: Webinare, an denen DOAG-Mitglieder kostenlos teilnehmen können. Das erste Webinar zum Thema „Automatic Memory Management“ ist bereits nach kurzer Zeit ausgebucht.

Dienstag, 31. Januar 2012

Die Bestellfrist für die Vortrags-DVD der DOAG 2011 Konferenz + Ausstellung endet. In der DOAG-Geschäftsstelle gehen mehr als 300 Anforderungen ein.



Die DOAG in der Presse

ix vom 1. November 2011

DOAG verändert ihre Struktur

Die Deutsche Oracle-Anwendergruppe (DOAG) passt ihre Struktur den neuen Geschäftsfeldern von Oracle an. Statt der bisherigen monolithischen Organisation mit ihren Special Interest Groups gibt es nun vier „Communities“. Sie betreuen die Bereiche Datenbank, Development und Data Warehouse, Infrastruktur und Middleware sowie Business Solutions. Zur Begründung für die Umstellung schreibt die DOAG: „Durch eine themengerechte Adressierung der Mitglieder verbessert die DOAG das Networking und den gezielten Erfahrungsaustausch innerhalb der Communities.“ Als Folge soll der DOAG-Vorstand auf der nächsten Mitgliederversammlung von acht auf fünf Mitglieder schrumpfen – viele seiner Aufgaben übernehmen in Zukunft die Communities direkt.

ix vom 1. Januar 2012

Deutsche Oracle-Anwendergruppe wächst stark

Auf der 24. Konferenz der Deutschen Oracle-Anwendergruppe (DOAG) trafen diesmal schon am ersten Tag rund 1500 Besucher ein – laut Veranstalter ein einmalig hoher Wert. Auch die Vereinigung selbst meldete starkes Wachstum: Mit mehr als 2000 Firmenmitgliedschaften habe sich diese Mitgliedergruppe gegenüber dem Vorjahr mehr als versechsfacht.

Das früher virulente Thema Lizenzkosten, vor allem im Zusammenhang mit virtualisierten Umgebungen, spiel-

te diesmal offiziell keine Rolle. Allerdings nicht, weil Oracle auf die Kunden zugegangen wäre. Vielmehr sieht das Unternehmen, so die DOAG, keinen Spielraum für Änderungen. „Oracle will den Markt in Richtung Exadata und Appliances gestalten, nicht in Richtung Virtualisierung“, hieß es zu den Hintergründen.

Nach Schätzung des DOAG-Vorstands interessieren sich immer noch gut die Hälfte der Teilnehmer in erster Linie für Oracles Datenbank. Für die andere Hälfte gab es in diesem Jahr deutlich ausgebaute Tracks zu Java und MySQL, die durch die Übernahme von Sun zu Oracle kamen. Beide spielten allerdings auf dem Oracle-Stand wie schon im Vorjahr gar keine Rolle, einzig Solaris fand dort als alte Sun-Technik ein Plätzchen.

Mit dem bislang von Oracle stiefmütterlich behandelten NoSQL beschäftigte sich ebenfalls ein Vortrag. Er konzentrierte sich auf die kürzlich von dem Unternehmen präsentierte Weiterentwicklung der BerkeleyDB zur „NoSQL-Datenbank“ sowie Hadoop. Bei Besuchern hinterließ die Präsentation Irritationen: Einige gewannen den Eindruck, bei NoSQL-Datenbanken könnten einzelne Datensätze verloren gehen. Das ließ NoSQL-Kenner vermuten, Oracle habe womöglich gar kein Interesse an der alternativen Technik.

Infomarkt von 14. Dezember 2011

DOAG Jahrestagung

Was Effizienz und Innovation betrifft, führt an Oracle kein Weg vorbei. Die operative Gewinnmarge des Softwarehauses lag im letzten Geschäfts-

jahr (1. Juni 2010 bis 31. Mai 2011) auch nach der Übernahme des Hardware-Produzenten Sun Microsystems stabil bei 34 Prozent. Zudem hat das Unternehmen zuletzt eine neue Generation von „Fusion“-Anwendungen vorgestellt, die alte und neue Softwarewelten vereinen soll. In Deutschland jedoch spielt das Unternehmen im Markt für Software-Anwendungen nur eine Statistenrolle. „Oracle ist hierzulande zwar eine starke Marke“, sagt Dietmar Neugebauer, der Vorstandsvorsitzende der Deutschen Oracle Anwendergruppe e. V. (DOAG). „Aber jenseits der Datenbanken sind die Oracle-Produkte in Deutschland kaum bekannt.“

Das vor allem durch Übernahmen getriebene Wachstum von Oracle macht auch der DOAG zu schaffen. Das Softwarehaus, einst als Datenbankspezialist gestartet, hat sich in den letzten zehn Jahren zu einem Fullservice-Anbieter entwickelt, der mehr als 1.000 Software- und Hardware-Produkte in seinem Schaufenster liegen hat.

Dabei steigt das Interesse auf Kundenseite. So kamen zur 24. DOAG-Jahrestagung, die vom 15. bis 17. November 2011 in Nürnberg stattfand, rund 2.000 Besucher. Das waren nach Angaben der Veranstalter zwar nicht viel mehr als im Vorjahr, dafür aber blieben die einzelnen Teilnehmer länger. Zudem wurde in der fränkischen Metropole das 5.000. DOAG-Mitglied begrüßt. Dennoch sieht Neugebauer Schwächen in der Außendarstellung. „Es gibt eine sehr große Zahl von Anwendern, die die DOAG noch gar nicht entdeckt haben.“

Weitere Pressestimmen lesen Sie unter <http://www.doag.org/presse/spiegel>



Fried Saacke (links) im Gespräch mit Klaus Vitt

Die Bundesagentur für Arbeit setzt in ihrer IT-Strategie 2015 auf die Oracle Fusion Middleware. Klaus Vitt, CIO der Bundesagentur für Arbeit, erläutert Fried Saacke, DOAG-Vorstand und Geschäftsführer, und Wolfgang Taschner, Chefredakteur der DOAG News, die Hintergründe.

„Das ist immer eine Abwägung zwischen Chancen und Risiken ...“

Was sind bei der Bundesagentur für Arbeit die größten Herausforderungen an die IT?

Vitt: Das sind in erster Linie die geschäftlichen Anforderungen, die wir umsetzen müssen. Wir unterscheiden hier zwischen den Anforderungen des Gesetzgebers, die exakt definiert zu einem bestimmten Termin realisiert sein müssen, und den internen Projekten. Der dritte Block ist die Implementierung unserer IT-Ziel-Landschaft in Richtung serviceorientierter Architektur. Die größte Herausforderung besteht nun darin, unsere begrenzten personellen und finanziellen Mittel so einzusetzen, dass wir alle drei Ziele erreichen.

Wie lösen Sie das in der Praxis?

Vitt: Für mich stehen an oberster Stelle die größtmögliche Transparenz sowie die Mitsprache unserer internen Kunden bei der Priorisierung der Arbeiten. Es gibt ein ausgefeiltes Anforderungsmanagement, das dem Kunden genau

zeigt, welchen Aufwand jede einzelne Anforderung verursacht. Er kann nun selbstständig seine Anforderungen im Rahmen der ihm zur Verfügung stehenden Ressourcen priorisieren. Bei Projekten über fünfhunderttausend Euro legt eine Entscheidungsrunde fest, welche Projekte in welcher Reihenfolge und Umfang realisiert werden. Über ein Portfolio-Management erkennen wir, wenn sich Projekte verzögern und wie sich das auf andere Projekte auswirkt.

Welche Rolle spielt dabei die serviceorientierte Architektur?

Vitt: Wir setzen nicht auf SOA, weil es ein Hype ist, sondern weil wir darin ganz konkrete geschäftliche Vorteile sehen. Mithilfe von einer SOA möchten wir von den Silo-Verfahren wegkommen, um dann rollenbasierte Oberflächen für Anwendergruppen zu schaffen. Geschäftsprozesse werden damit optimal unterstützt. Dazu ein Beispiel: Die Mitarbeiter in unseren

ServiceCentern benötigen derzeit bis zu fünfzehn IT-Anwendungen, um alle eingehenden Fragen beantworten zu können. Künftig soll dieser Mitarbeiter nur noch eine Oberfläche haben, in der die einzelnen Services angeboten sind. Das heißt, wir trennen die Oberflächen von der Business-Logik, sodass wir jederzeit für die unterschiedlichen Anwendergruppen separate Oberflächen und die einzelnen Services gezielt bereitstellen können.

Wie können Sie den daraus resultierenden Nutzen messen?

Vitt: Der geschäftliche Nutzen ist darin erkennbar, dass der Mitarbeiter weniger zwischen den einzelnen Applikationen wechseln muss. Zudem benötigt die einheitliche, durchgängige Oberfläche deutlich weniger Schulungsaufwand. Der Mitarbeiter kann sich gezielt auf die von ihm benötigten Funktionen konzentrieren und braucht die anderen nicht zu erlernen. Außerdem bekommt er die Möglich-

keit, beispielsweise einen Prozess mit einem Anrufer zu beginnen und ihn dann zur Fertigstellung an ein Back-Office weiterzuleiten.

Wie weit ist die Umstellung auf rollenbasierte Oberflächen fortgeschritten?

Vitt: Wir haben relativ früh damit begonnen, die geschilderte Grundidee in



**Zur Person:
Klaus Vitt**

Klaus Vitt ist seit 2006 Geschäftsführer (CIO-Funktion) der zentralen IT der Bundesagentur für Arbeit (BA) und zugleich Vorsitzender der Geschäftsführung des IT-Systemhauses der BA. Sein Bereich umfasst die strategische IT-Planung und -Steuerung, IT-Anforderungsmanagement und IT-Sicherheit als geschäftspolitische Führungsaufgaben. Hinzu kommt die bedarfsgerechte Umsetzung von informations- und kommunikationstechnischen Aufträgen der Zentrale der BA in eigener Betriebsverantwortung.

Klaus Vitt (Dipl.-Ing. FH; Dipl.-Math. Univ.) studierte Nachrichtentechnik an der Fachhochschule der Deutschen Bundespost. Anschließend schloss er sein Studium der Mathematik/Informatik an der Universität in Dortmund ab. Seine ersten Berufserfahrungen sammelte er bei dem Computer-Hersteller Sperry Univac als Fachlehrer für Betriebssysteme und Programmiersprachen sowie bei der Softwarefirma Applied Data Research als System-Engineer. Verantwortliche Positionen bei der Bertelsmann AG als Leiter der zentralen Datenverarbeitung bei Gruner & Jahr, Rechenzentrumsleiter in der zentralen Datenverarbeitung und als internationaler IT-Koordinator im Clubbereich schlossen sich an. 1996 bis 1999 war er Geschäftsführer für IT und Kundenbetreuung bei T-Mobile. Ab 1999 leitete er den Geschäftsbereich Customer Care/IT bei der Deutschen Telekom. Von 2001 bis 2006 war er Bereichsvorstand der T-Com für Informationsmanagement und Prozesse.

einem Prototyp umzusetzen und diesen in einem Modellversuch zu verifizieren. Die daraus resultierenden Erfahrungen sind sehr positiv. Derzeit sind wir dabei, Last- und Performance-Tests durchzuführen sowie die Oberflächen zu gestalten.

Welche Erfahrungen daraus können Sie an andere Unternehmen weitergeben?

Vitt: Wir haben es hier mit bestehenden IT-Anwendungen zu tun. Die neuen Oberflächen sind anwenderspezifisch und sollen nicht bei jeder Änderung neu gestaltet werden müssen. Wir entwickeln jetzt Gestaltungsmuster, um daraus sämtliche rollenbasierten Oberflächen zusammensetzen. Die Vorteile sind vielschichtig: Der Anwender hat immer das gleiche „Look & Feel“ und wir können aus den Gestaltungsmustern nahezu automatisiert die Oberflächen, Frameworks, Codes etc. generieren.

Sind die Oberflächen weitgehend selbst-erklärend, um den Schulungsaufwand zu reduzieren?

Vitt: Wir haben das in unserem Modellversuch erprobt. Die Mitarbeiter, die noch nicht so lange im Service-Center gearbeitet haben, sehen hier deutliche Vorteile.

Wie aufwändig ist die Umstellung auf SOA?

Vitt: Die Herausforderung bei der Implementierung einer serviceorientierten Architektur besteht darin, die bestehenden Anwendungen in Services zu zerlegen, obwohl wir diese gleichzeitig auch weiterentwickeln müssen. Außerdem müssen wir die alten Oberflächen auch nach der Zerlegung in Services zunächst noch weiterbetreiben. Wir stellen also im laufenden Betrieb auf SOA um.

Was sind dabei die Besonderheiten?

Vitt: Wir schaffen den Umstieg, indem wir diese Zerlegung in einzelne fachliche Services exakt planen. Außerdem müssen wir bereits bei der ersten rollenbasierten Oberfläche rund achtzig



Fotos: Wolfgang Taschner

Prozent der Geschäftsfälle in der entsprechenden Anwendergruppe abdecken, damit der Anwender nicht gleichzeitig mit der alten und der neuen Oberfläche arbeiten muss. Deshalb dauert es auch relativ lange, bis die erste Version einer rollenbasierten Oberfläche zur Verfügung steht. Das wird in etwa einem Jahr der Fall sein.

Setzen Sie Oracle-Produkte bei der SOA-Umstellung ein?

Vitt: Wir haben uns mit unterschiedlichen Plattformen beschäftigt und uns dann für die Oracle Fusion Middleware entschieden, weil sie eine Menge Funktionalitäten bietet, die wir sehr gut nutzen können. Hinzu kommt ADF als Entwicklungswerkzeug.

Wie sind hier Ihre Erfahrungen?

Vitt: Die Oracle Fusion Middleware besteht ja aus einer Reihe unterschiedlicher Komponenten. Das Zusammenspiel dieser Komponenten hat nicht immer funktioniert. Hinzu kamen viele Kinderkrankheiten, die wir erst zusammen mit dem Oracle Support in den Griff bekommen mussten. Außer-

dem haben wir sehr hohe Anforderungen an das Last- und Performance-Verhalten. Es ist uns jedoch gelungen, eine Architektur aufzubauen, die horizontal skalierbar ist.

In welchem Bereich müsste Oracle hier noch nachlegen?

Vitt: Es wäre wichtig, die Komponenten so zu gestalten, dass man nicht gezwungen ist, beim Versionswechsel einer einzelnen Komponente immer die anderen gleich mit upgraden zu müssen. Außerdem sollten beim Versionswechsel einer Komponente die Auswirkungen auf andere Komponenten dokumentiert sein.

Wie beurteilen Sie die Produkt-Strategie von Oracle?

Vitt: Ich halte die Produkt-Strategie für sehr visionär. Uns kommt es entgegen, wenn uns ein Hersteller eine homogene Middleware-Plattform zur Verfügung stellen kann.

Haben Sie kein Problem, von einem einzigen Hersteller abhängig zu sein?

Vitt: Je nachdem, wie intensiv die Hersteller-Komponenten in die Anwendungslandschaft integriert sind, besteht immer eine Abhängigkeit von diesem. Wir achten allerdings schon darauf, dass die Abhängigkeit nicht zu groß wird. Das ist aber immer eine Abwägung zwischen den Chancen und den Risiken. Auf der einen Seite macht man sich abhängig von einem Hersteller, und reduziert damit auf der anderen Seite den Aufwand für die System-Integration.

Welche Rolle spielen bei Ihnen die neuen Systeme von Oracle wie beispielsweise Exadata?

Vitt: Wir haben uns damit beschäftigt, sehen aber hier eine relativ hohe Abhängigkeit, weil ja die Hardware- und Betriebssystem-Komponenten ebenfalls von Oracle kommen. Da die Innovationszyklen der X86-Hardware extrem schnell sind, sind wir uns nicht sicher, ob Exadata der richtige Ansatz ist.

Auf einen Blick: Die Informationstechnik der Bundesagentur für Arbeit

Die IT der Bundesagentur für Arbeit gliedert sich mit 2.200 Mitarbeiterinnen und Mitarbeitern in die Bereiche „IT-Steuerung“, „IT-Systemhaus“ und „regionaler IT-Service“. Das IT-Systemhaus ist dabei der operative IT-Dienstleister der BA und sorgt dafür, dass alle BA-Mitarbeiter IT-unterstützt optimal arbeiten können. Das IT-Systemhaus gliedert sich in die Bereiche „Systementwicklung“, „Betrieb/Infrastruktur“ und den internen Service. Zum Bereich „Betrieb/Infrastruktur“ zählen auch die Netz-Infrastruktur und die Arbeitsplatzsysteme.

Die Informationstechnik der Bundesagentur für Arbeit vollbringt die komplette Rechenzentrumsleistung für 1.900 angebundene Liegenschaften. Mit 170.000 vernetzten PC-Arbeitsplätzen und der dazugehörigen Infrastruktur (12.400 Server, drei zentrale hochverfügbare Rechenzentren) betreibt die BA-Informationstechnik eine der größten IT-Landschaften Deutschlands und entwickelt diese ständig weiter.

Wie stellen Sie die Hochverfügbarkeit Ihrer Daten sicher?

Vitt: Wir setzen Real Applications Cluster ein und haben damit gute Erfahrungen gemacht. Die geschäftskritischen Anwendungen sind über zwei Standorte gespiegelt. Mittlerweile läuft das alles sehr stabil. Wir haben allerdings eine ganze Zeit gebraucht, um auf diesen Stand zu kommen.

Welchen Tipp können Sie bezüglich Hochverfügbarkeit an Ihre Kollegen weitergeben?

Vitt: Auf Kundenseite ist die Standardisierung der Systemkomponenten unerlässlich, sprich alle Datenbanksysteme müssen synchron auf dem gleichen Level hinsichtlich Version und Patches sein.

In welche Richtung wird sich Ihre IT in den kommenden Jahren entwickeln?

Vitt: Ziel ist die Bereitstellung einer leistungs- und zukunftsfähigen IT-Landschaft und -Infrastruktur zu marktüblichen Preisen. Die zentrale Herausforderung für den IT-Betrieb ist, eine hohe Verfügbarkeit und Performance der IT-Verfahren zu gewährleisten. Maßnahmen in diese Richtung sind Standardisierung, höhere Auslastung der Rechnerkapazitäten und Reduzierung des Energieeinsatzes sowie Innovation und Modernisierung durch Einführung neuer Technologien zur Effizienzsteigerung.

Wie sehen Sie den Stellenwert einer Anwendergruppe wie der DOAG?

Vitt: Für mich steht der Erfahrungsaustausch an erster Stelle. Ich wünsche es jedem Anwender, dass ihm negative Erfahrungen mit den Produkten erspart bleiben.



Wie kann eine virtuelle Umgebung besser verfügbar sein als eine physische Umgebung, wo doch mit der Virtualisierung auch zusätzliche Komplexität eingeführt wird? Der Artikel zeigt, dass virtuelle Umgebungen hier mehr zu bieten haben, als es zunächst scheint.

Hochverfügbar durch Virtualisierung – (wie) geht das?

Patrick Schwanke, CarajanDB GmbH

Angesichts der zahlreichen Technologien, die unter den Stichworten „Virtualisierung“ oder „Partitionierung“ laufen, konzentrieren wir uns hier auf Intel-basierte Virtualisierung mit VMware oder Oracle VM. Ziel ist die Reduzierung von Auszeiten bei Hardware-Wartungen, Hardware-Ausfall, Blue Screen oder Kernel Panic, Ausfall der Oracle-Instanz oder auch einer größeren Infrastruktur (Desaster-Fall). Im weiteren Sinne zählen auch der De-facto-Ausfall beziehungsweise die Nichterfüllung von SLAs durch schlechte Antwortzeiten hierzu, auch dabei können virtuelle Umgebungen helfen. Zum Schluss stellt der Autor Erfahrungen hinsichtlich Lizenz-Anforderungen für die unterschiedlichen Absicherungsszenarien vor.

Wartung der Hardware oder des Hypervisors

Techniken wie vMotion von VMware oder VMs Secure Live Migration von Oracle gehören heutzutage zum Standard-Repertoire virtueller Infrastrukturen. Damit können einzelne VMs ohne jede Auszeit auf einen anderen physischen Server migriert werden. Auf diese Weise kann man einen physischen Server „evakuieren“, das heißt alle VMs auf andere Server verlagern. Dann führt man die anstehenden Wartungsarbeiten durch, etwa den Einbau oder Austausch von Hardware oder auch ein Upgrade des Hypervisors. Anschließend startet man den Server durch und bevölkert ihn analog wieder mit VMs.

Im einfachsten Fall geschieht das Ent- und Bevölkern des Servers manuell. In Verbindung mit VMwares voll-

automatischem Dynamic Resource Scheduling (DRS) genügt es auch, für den betreffenden Server den sogenannten „Maintenance-Modus“ zu (de-)aktivieren. Die DRS-Logik führt dann automatisch alle notwendigen vMotion-Events durch.

Ausfall der Hardware oder des Hypervisors

Virtuelle Infrastrukturen bieten eine hochinteressante Alternative zum klassischen Failover-Cluster (active/passive-Cluster), nämlich den sogenannten „HA-Cluster“. Die Namensgebung ist hierbei einheitlich zwischen Oracle VM und VMware, wobei das letztgenannte Unternehmen diese Technik schon wesentlich länger anbietet.

Ein HA-Cluster ist eine Gruppe physischer Server, die wechselseitig füreinander einspringen können, wobei entsprechende Reservekapazitäten konfiguriert werden können und sollten. Beispielsweise sollte ein Fünf-Knoten-Cluster nicht zu mehr als 80 Prozent ausgelastet sein, da ansonsten der Ausfall eines Knotens vom Rest-Cluster nicht geschultert werden könnte.

Der Charme eines HA-Clusters liegt darin, dass mit minimalem Aufwand jede VM mit jeder darin laufenden Anwendung clusterfähig wird, ohne dass man sich mit Techniken wie MSCS, Linux-Cluster oder Oracle RAC One Node beschäftigen muss.

Der Nachteil gegenüber einem herkömmlichen Failover-Cluster ist, dass nach dem Ausfall des aktiven Knotens und der Erkennung dieser Tatsache die VM einfach auf einem anderen physischen Server gestartet wird; der passive Knoten muss also von Null auf gebootet werden. Somit handelt es sich um einen Kaltstart, die Auszeit liegt daher eher im Minuten- als im Sekundenbereich. Da die Knoten mithilfe von Heartbeats überwacht werden, erkennt der Mechanismus auch einen Ausfall des Hypervisors wie VMwares sogenannten „Purple Screen Of Death“ (PSOD, siehe Abbildung 1).

Windows- oder Linux-Ausfall innerhalb der VM

Ein VMware HA-Cluster kann auch überwachen, ob das Betriebssystem innerhalb der VM noch läuft (Stichwort

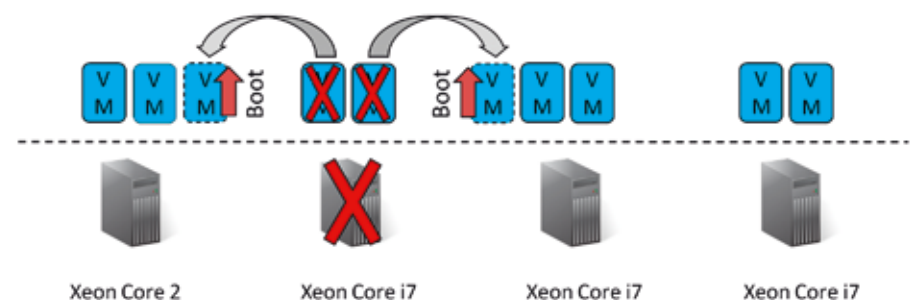


Abbildung 1: VMware HA-Cluster für Knoten- oder Hypervisor-Ausfall

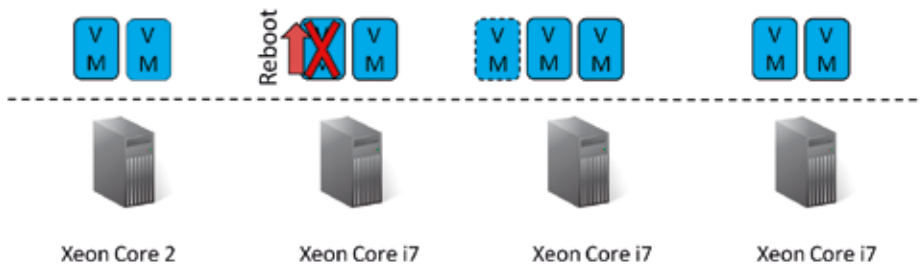


Abbildung 2: VMware HA-Cluster für den Ausfall des Gastbetriebssystems

„Virtual Machine Monitoring“). Im Falle eines Windows-Bluescreen oder eines Linux-Kernel-Panic löst der HA-Cluster einen Reboot der VM aus (siehe Abbildung 2). Nach mehreren erfolglosen Reboots kann man auch eine Alarm-Mail verschicken lassen. Auch hier sorgt ein Heartbeat dafür, solche Situationen zu erkennen, diesmal jedoch ein Heartbeat zwischen der HA-Clusterlogik und den VMware-Tools innerhalb der virtuellen Maschinen. Voraussetzung ist daher, dass – zumindest für die zu überwachenden VMs – die VMware-Tools installiert sind. Für Windows-VMs ist dies normalerweise kein Problem, für Linux-VMs wird ein GCC-Compiler benötigt, der aber ohnehin für die Installation der Oracle-Software benötigt wird.

Bevor man diese Überwachungsfunktion aktiviert, sollte man peinlich genau prüfen, ob auch wirklich in allen betreffenden VMs die VMware-Tools installiert sind und die anderen VMs aus der Überwachung ausgenommen sind. Ansonsten führt das Nicht-Vorhandensein der VMware-Tools nämlich zu fehlenden Heartbeats, sodass die VM permanent durchgestartet würde. Ähnliches kann passieren, wenn die VM extrem ausgelastet ist und die VMware-Tools keine Prozessorzeit mehr bekommen. Lässt der Heartbeat länger auf sich warten als der eingestellte Timeout, wird die VM durchgestartet. Der Timeout sollte daher nicht zu gering gewählt werden. Ein zweistelliger Sekundenbetrag ist zu empfehlen.

Ausfall der Oracle-Instanz

Hinsichtlich eines Ausfalls der Oracle-Instanz oder des Listener kann keine der Virtualisierungslösungen einen

direkten Mehrwert erbringen. Genau wie in der physischen Welt ist hier eine Überwachung notwendig, beispielsweise mithilfe der Oracle-Clusterware. Seit der Datenbank-Version 11g R2 kann diese nicht nur für den Aufbau von RAC-Datenbanken verwendet werden, sondern auch für Nicht-RAC-Datenbanken einen automatischen Neustart von Listener, Datenbank- oder ASM-Instanz veranlassen (Stichwort „Oracle Restart“). Wichtig zu wissen ist, dass ab der Datenbank-Version 11.2.0.2 auch RAC-Instanzen in virtuellen Maschinen unterstützt werden (siehe [1]). Unter [2] findet sich eine Beschreibung, wie ein RAC-Cluster auf Basis von VMware aufgesetzt wird.

Disaster Recovery

Auch wenn das Thema „Disaster Recovery“ nicht Schwerpunkt dieses Artikels ist, seien hier kurz einige Lösungen vorgestellt. VMware selbst bietet mit dem Site Recovery Manager (vSRM) und dem seit vSRM5 enthaltenen vSphere Replication eine Lösung, um virtuelle Maschinen an einen entfernten Standort zu replizieren.

Ähnliches liefern Drittanbieter-Lösungen wie vReplicator von Quest

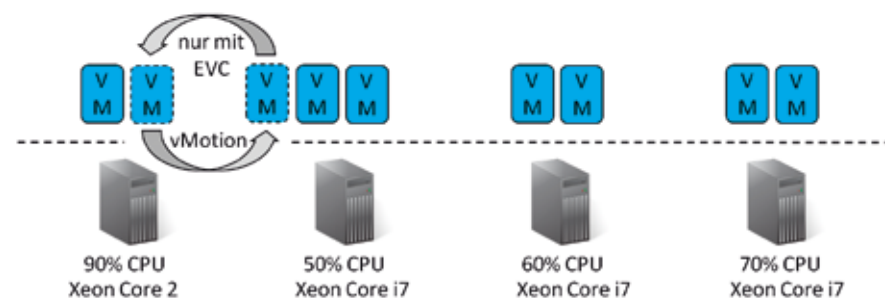


Abbildung 3: VMware DRS-Cluster für Lastverteilung

Software oder Backup & Replication von Veeam.

Auch VMware Fault Tolerance (FT) wirkt auf den ersten Blick interessant, da es eine Replikation einzelner VMs ohne spezielle Storage-Anforderungen leistet und Bestandteil der vSphere 4 Advanced Edition beziehungsweise vSphere5 Enterprise Edition ist. Hier sind jedoch zahlreiche Restriktionen zu beachten, die in [3] detailliert beschrieben sind. Insbesondere ist FT auf VMs mit einer einzigen vCPU beschränkt. Kritische Datenbanken sind dagegen oftmals mit mehr CPU-Kernen ausgestattet oder benötigen diese Option für die Zukunft. In diesen Fällen kommt FT nicht in Frage.

Grundsätzlich sollte man bei virtualisierten Oracle-Datenbanken Oracle-eigene DR-Szenarien auf Basis von Physical-Standby-Datenbanken prüfen, da diese sehr robust und einfach zu verwalten sind. Sei es per Data Guard für die Enterprise Edition oder per Skript beziehungsweise mithilfe von Werkzeugen wie Dbvisit Standby oder Libelle DBShadow für die Oracle Standard Edition (One).

Lastverteilung

Ein Aspekt, den man nicht unterschätzen sollte, ist die automatische Lastverteilung. Bei VMware läuft diese Funktion seit Längerem unter der Bezeichnung „Dynamic Resource Scheduling“ (DRS, siehe Abbildung 3). Oracle VM bietet dies in der neuesten Version 3.0 unter demselben Namen an. Dabei werden mehrere physische Server zu einem DRS-Cluster zusammengeschlossen. Im Oracle-Umfeld sollte man dies nicht als Alternative

zu RAC sehen, wohl aber als Ergänzung. Ein DRS-Cluster überwacht die Auslastung der physischen Server und verschiebt bei Ressourcen-Engpässen einzelne VMs per vMotion auf einen anderen physischen Server mit geringerer Auslastung.

DRS liefert zwar keinen zusätzlichen Schutz gegen Ausfall irgendwelcher Komponenten, hilft aber, schlechte Antwortzeiten durch überlastete Infrastruktur zu vermeiden. Sobald es also nicht nur um nominelle Verfügbarkeit geht, sondern auch um die Einhaltung von SLAs und die Zusicherung einer gewünschten Service-Qualität, ist DRS eine interessante Ergänzung.

Einige Hinweise aus den Erfahrungen des Autors:

- Ein DRS-Cluster sollte möglichst von vornherein im EVC-Modus (Enhanced vMotion Compatibility) erstellt und betrieben werden. Nachträglich lässt sich EVC nur noch mit einer Auszeit für alle VMs aktivieren. Wenn sich im Laufe der Zeit unterschiedliche CPU-Generationen in der Hardware ansammeln, werden DRS und vMotion ohne EVC zu einer Einbahnstraße, das heißt vMotion ist nur zu identischen oder neueren Prozessormodellen hin möglich. Der Weg zurück auf einen älteren Prozessor ist dann verwehrt, Server mit älteren Prozessoren verweisen zunehmend. Abbildung 3 zeigt dies.
- Da Oracle-RAC nicht nur Lastverteilung, sondern auch Hochverfügbarkeit realisieren soll, dürfen bei einem DRS-Cluster die RAC-VMs einer Datenbank nicht auf demselben physischen Server laufen, da dieser ansonsten einen Single Point of Failure darstellen würde. Hierfür setzt man eine sogenannte „Anti-Affinity-Regel“ ein, die verhindert, dass DRS die beiden Instanzen auf denselben physischen Server schiebt.
- DRS kann im voll- sowie im halbautomatischen Modus betrieben werden. Im letzteren Fall werden lediglich vMotion-Empfehlungen ausgesprochen, die ein VMware-Administrator per Knopfdruck umsetzen kann. Oftmals geht dadurch

aber unnötig Zeit verloren. Der Autor empfiehlt daher den vollautomatischen Modus. Die gelegentlich anzutreffende Befürchtung, dass dann ständig unkontrolliert VMs verschoben werden, kann er nicht bestätigen. Und selbst wenn dem so wäre, wird durch vMotion-Ereignisse keinerlei Auszeit verursacht. Außerdem kann man die vMotion-Häufigkeit durch den Aggressivitätslevel des DRS-Clusters beeinflussen.

- DRS-Clustering kann mit dem bereits beschriebenen HA-Clustering kombiniert werden. Dies ist auch die übliche Vorgehensweise in der Praxis; es wird mit kombinierten HA- und DRS-Clustern gearbeitet.

Lizenzierung

Alle Lizenz-Informationen werden nach bestem Wissen des Autors gegeben und spiegeln den Stand Dezember 2011 wider. Verbindliche Informationen finden sich jeweils bei den Herstellern selbst, entsprechende Links sind nachfolgend angegeben. Hinsichtlich der erforderlichen Lizenzierung sollte man zunächst entscheiden, welche VMware-Lizenzedition benötigt wird. Abbildung 4 gibt einen Überblick über die verschiedenen Editionen von vSphere v4 und v5 im Hinblick auf die hier besprochenen Clustering-Fähigkeiten. Als weitere Merkmale sind Obergrenzen hinsichtlich CPU- und Memory-Ausstattung der physischen Server und der VMs zu

beachten. Schließlich ist auch die sogenannte „Hot-Add-Funktion“ aufgeführt, die das Ändern der CPU- und Memory-Ausstattung einer VM im laufenden Betrieb ermöglicht, was bei Ressourcen-Engpässen innerhalb der VM für eine einfache, vertikale Skalierung sorgt.

Verbindliche Lizenzinformationen finden sich auf der VMware-Webseite, etwa unter [4]. Angesichts der aufgeführten Merkmale empfiehlt der Autor in den meisten Fällen die vSphere Enterprise Edition (siehe Abbildung 4).

Oracle unterteilt die verschiedenen Virtualisierungstechniken in Hard- und Soft-Partitioning, wobei die genaue Abgrenzung in [5] beschrieben ist. Dabei zählt VMware grundsätzlich als Soft-Partitioning, es müssen also Oracle-Datenbank-Lizenzen für den gesamten vSphere-Server erworben werden, auf dem eine VM mit einer Oracle-Datenbank läuft. Wird ein HA- und/oder DRS-Cluster benutzt, sind alle vSphere-Server des Clusters zu lizenzieren. Für größere Datenbank-Umgebungen lässt sich dieser Nachteil umgehen, indem ein Oracle-Only-Cluster (oder ein Oracle-Only-vSphere-Server) aufgebaut wird, auf dem ausschließlich Oracle-Datenbanken laufen. In diesem Falle wird sowieso die komplette Hardware benutzt und lizenziert. Je geringer dagegen der Anteil an Oracle-Software, desto höher der Lizenz-Overhead.

Oracle VM wird dagegen unter bestimmten Bedingungen, nämlich mit

	v4 Standard	v4 Advanced	v4 Enterprise	v4 Ent. Plus	v5 Standard	v5 Enterprise	v5 Ent. Plus
Max RAM pro Server	256GB	256GB	256GB	1TB	No limit	No limit	No limit
Max CPU-cores pro CPU-Sockel	6	12	6	12	No limit	No limit	No limit
Max vRAM pro CPU-Sockel	No limit	No limit	No limit	No limit	32GB	64GB	96GB
Max CPU-cores pro VM	4	4	4	8	8	8	32
vMotion und HA-Cluster	Ja	Ja	Ja	Ja	Ja	Ja	Ja
vCPU & vMem Hot-Add	Nein	Ja	Ja	Ja	Nein	Ja	Ja
DRS-Cluster	Nein	Nein	Ja	Ja	Nein	Ja	Ja

Abbildung 4: VMware-Lizenzeditionen

aktiviertem CPU-Pinning, als Hard-Partitioning gewertet (siehe [6]), sodass nur die tatsächlich benutzten physikalischen CPUs lizenziert werden müssen. Bisher ging dadurch allerdings die „Live Migration“-Fähigkeit, also das Pendant zu vMotion, verloren beziehungsweise es musste auf dem neuen physikalischen Server ein erneutes CPU-Pinning durchgeführt werden. Der Autor hat bisher keine Erfahrungen darüber, ob die in Oracle VM 3 enthaltenen HA- und DRS-Funktionen mit der CPU-Pinning-Anforderung verträglich, ob also HA und DRS in einer Hard-Partitioning-Umgebung möglich sind.

Fazit

Verschiedenste Arten geplanter und ungeplanter Auszeiten können mithilfe

der beschriebenen Virtualisierungstechniken umgangen oder zumindest reduziert werden. VMware hat nach wie vor einen technologischen Vorsprung vor Oracle VM. Oracle VM hat erst mit der neuesten Version 3 vergleichbare Techniken eingebaut, hier stehen praktische Erfahrungen noch aus. Lizenztechnisch lohnt sich der Einsatz von VMware jedoch nur, wenn reine Oracle-Umgebungen aufgebaut werden, also separate vSphere-Server oder -Cluster nur für Oracle-Datenbanken.

Referenzen

- [1] Support für Oracle-Produkte auf VMWare: MOS Doc-ID 249212.1
- [2] Aufbau eines RAC-Clusters auf Basis von VMware vSphere: <http://streetkiter.wordpress.com/2011/11/08/oracle-rac-11-2-0-3-auf-vmware-esxi/>
- [3] Restriktionen bei VMware Fault Tolerance: <http://communities.vmware.com/blogs/>

- vmroyale/2009/05/18/vmware-fault-tolerance-requirements-and-limitations
- [4] VMware-Lizenzinformationen: http://www.vmware.com/files/pdf/vsphere_pricing.pdf
- [5] Oracle Hard Partitioning und Soft Partitioning: <http://www.oracle.com/us/corporate/pricing/partitioning-070609.pdf>
- [6] Hard Partitioning mit Oracle VM: <http://www.oracle.com/technetwork/topics/virtualization/ovm-hardpart-167739.pdf>

Patrick Schwanke
CarajanDB GmbH
patrick.schwanke@carajandb.com



IT-Consulting	Schulungen	Software-Lösungen	Oracle Lizenzen
<ul style="list-style-type: none"> › Performance Tuning <ul style="list-style-type: none"> • Oracle Datenbank Tuning • Oracle SQL + PL/SQL Tuning › Real Application Clusters › Data Guard + Fail Safe › Datenbank Management <ul style="list-style-type: none"> • Konfiguration • Backup & Recovery • Migration und Upgrade › OEM Grid Control › Oracle Security › Services <ul style="list-style-type: none"> • Remote DBA Services • Telefon-/Remotesupport <p>Nutzen Sie unsere Kompetenz für Ihre Oracle Datenbanken.</p>	<ul style="list-style-type: none"> › Oracle SQL › Oracle PL/SQL › Oracle DBA › Oracle APEX › Backup & Recovery › RMAN › Neuerungen 10g/11g › Datenbank Tuning › Datenbank Monitoring › Datenbank Security Wir bieten Ihnen öffentliche Kurse sowie Inhouse-Schulungen. 	<ul style="list-style-type: none"> › Individualsoftware <ul style="list-style-type: none"> • .NET und Visual Basic • Java › Oracle APEX › PL/SQL Unser Ziel: Individuelle Softwareentwicklung mit Fokus auf Ihre Zufriedenheit. 	<ul style="list-style-type: none"> › Oracle Datenbanken <ul style="list-style-type: none"> • Standard Edition One • Standard Edition • Enterprise Edition • Personal Edition › Oracle Produkte <ul style="list-style-type: none"> • Enterprise Manager • Oracle Tools Optimale Lizenzierung durch individuelle Beratung.



Nach knapp zwei Jahren tritt die Berliner Firma KeepTool wieder mit einer neuen Hauptversion ihrer Verwaltungssoftware Hora an die Öffentlichkeit. Seit Ende November 2011 ist die Version 10 freigegeben und steht zum Download zur Verfügung. Der nachfolgende Artikel beschreibt nicht nur die wichtigsten der zahlreichen neuen Features, sondern gibt darüber hinaus auch einen generellen Einblick in die Arbeitsweise und den Funktionsumfang dieses Werkzeugs. Nach wie vor hilft Hora dabei, die tägliche Arbeit sowohl von DBAs als auch von Entwicklern nachhaltig zu optimieren.

Hora, die 10. – Kontinuität und Innovation

Günter Unbescheid, Database Consult GmbH

Auf den ersten Blick hat sich gegenüber der Vorgängerversion an der Oberfläche kaum etwas geändert. Dies ist durchaus als Kompliment zu verstehen, da der Nutzer sofort in der gewohnten Weise seiner Arbeit nachgehen kann und sich nicht neu orientieren muss. Bei genauerer Betrachtung fallen jedoch zahlreiche Neuerungen und Ergänzungen ins Auge, welche die neue Release-Nummer durchaus rechtfertigen. Hier hat man den Eindruck, dass sie allesamt aus der Praxis und mit Bedacht in das Werkzeug integriert wurden: Beispielsweise ist in diesem Zusammenhang „Feature Usage“ (wichtig für die Lizenzierung) zu nennen, die Mehrfachauswahl zur Generierung von DDL, die optimierte Ansicht von Rollen und ihnen zugeordneten Privilegien, die Übersicht von Tabellen- und Spaltenstatistiken, die

„Column Usage“ zur Ermittlung von Zugriffsmustern und die Pivot-Ansichten für spontane Datenanalysen – um an dieser Stelle nur einige wenige zu nennen.

Nach wie vor lassen sich die Vorzüge von Hora wie folgt auf den Punkt bringen:

- Hora ist ein klassisches Client-Werkzeug, das effizient und einfach installiert wird und mit minimaler Infrastruktur auskommt, also ohne Agenten, Repositories und Application-Server arbeitet.
- Der Benutzer kann sich wie gewohnt direkt in die verfügbaren Zielsysteme einwählen. Dies sind Oracle-Datenbanken der Versionen 8 bis 11.2, Single Instance oder RAC.
- Weitere Vorteile sind die schnelle, intuitive Navigation und die über-

sichtliche Darstellung aller maßgeblichen Funktionsbereiche der Datenbank aus den Kontexten „Verwaltung“, „Optimierung“ und „Entwicklung“. Das, was zusammengehört, wird über beliebig gestaffelte Tab-Reiter auch zusammen präsentiert und kann direkt über den Kontext angepasst werden.

- Es existieren umfangreiche Möglichkeiten, die grafische Oberfläche den eigenen Bedürfnissen anzupassen.
- Menügesteuerte Abfragen und die Darstellung von Tabellen-Inhalten erleichtern die Arbeit. Bei Aktivierung von Unicode lassen sich auch internationale Zeichen darstellen.

Diese Charakteristik zeigt auch gleichzeitig die Grenzen von Hora auf: Log- und Trace-Dateien lassen sich beispielsweise mit Mitteln von Hora nicht darstellen und durchsuchen, es sei denn, man legt entsprechende externe Tabellen an. Start- und Stopp-Befehle sowie Betriebssystem-Jobs müssen separat gesteuert werden.

Systemvoraussetzungen

Hora wurde in der Entwicklungsumgebung Delphi erstellt und ist aus diesem Grunde prädestiniert für Windows-Client-Systeme. Es werden dementsprechend die 32-bit-Versionen von XP (SP3), Vista (SP2) und Windows 7 offiziell unterstützt. Hora ist jedoch auch unter Windows 7 (64-bit) sowie unter einer „floating license“ auf virtuellen Systemen wie MS Virtual PC und VMware einsetzbar.

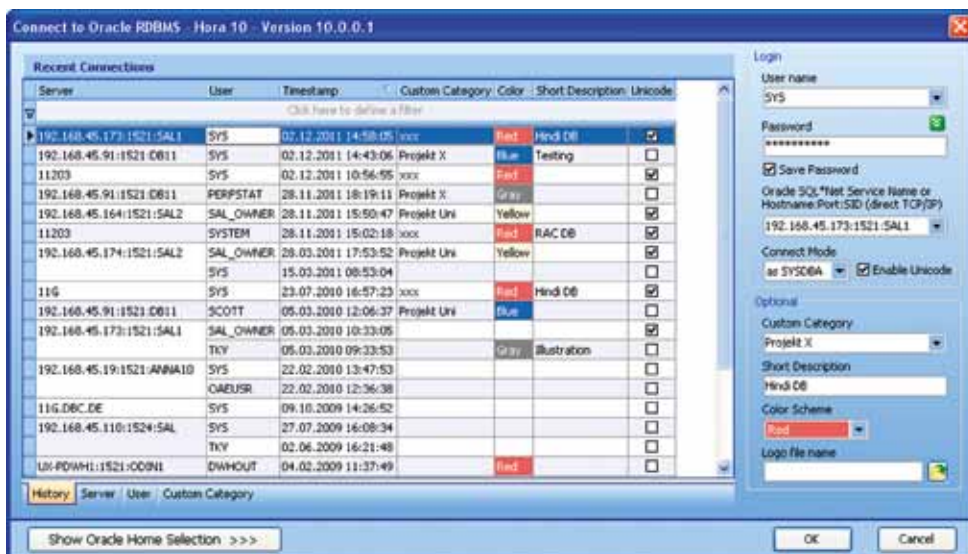


Abbildung 1: Das Einwahlmennü mit unterschiedlichen Gruppierungsmöglichkeiten

Die Verbindung zum Zielsystem kann direkt über TCP/IP oder einen Oracle-Client erfolgen, der dann natürlich kompatibel zum jeweiligen Zielsystem sein sollte. Bei Einbindung der Oracle-Client-Software bieten sich erweiterte Möglichkeiten, wie beispielsweise die Nutzung von Kerberos-Tickets zur Authentifizierung bei der Datenbank – sofern eine entsprechende Lizenzierung gewährleistet ist.

Einwahl in die Zielsysteme

Datenbank-Landschaften bestehen zum großen Teil aus Dutzenden, häufig auch Hunderten von Zielsystemen. Dementsprechend hat die effiziente Einwahl und Konfiguration der Ziele oberste Priorität. Hora bietet hier einen übersichtlichen Login-Dialog, der einmal getätigte Verbindungen – mit oder ohne Passwort – speichert sowie unterschiedliche Gruppierungs- und Sortierkriterien bietet, orientiert nach Servern, Benutzern oder der Einwahl-Historie. Darüber hinaus können Verbindungen kategorisiert, farblich markiert und mit Grafiken versehen werden. Parallele Verbindungen sind möglich, wobei jede Verbindung in einem separaten Hauptfenster gehalten wird (siehe Abbildung 1).

Navigation und „Look and Feel“

Der gewaltige Funktionsumfang aktueller Oracle-Versionen erzwingt effiziente und weitgehend intuitive Navigationsoberflächen. Hier nutzt Hora die bewährten Seitenleisten: Drei Hauptseiten (Standard, Additional und DBA) gruppieren thematisch das Seitenmenü. Eine vierte Hauptseite (Favorites) kann vom Benutzer frei mit Funktionskontexten belegt werden. Darüber hinaus lassen sich über „Settings“ bis zu zwei Einstiegsmasken sowie alle gängigen Präferenzen übersichtlich einstellen (siehe Abbildung 2).

Die Aktivierung eines Kontextes wie „Tablespaces“ öffnet rechts das entsprechende Hauptfenster, das seinerseits den Inhalt nochmals über Tab-Reiter gruppiert. Im Datenbereich der Fenster lassen sich nicht

nur Sortierungen und Filter angeben, sondern die angezeigten Daten gruppen- oder spaltenweise ein- oder ausschalten. Numerische Spalten werden unten auf der Seite automatisch summiert, numerische Einzelwerte mit einem Balkendiagramm hinterlegt. Bei manchen Seiten wie beispielsweise „Tables“ lassen sich die Darstellungen ein- oder mehrstufig durch Ziehen der Spalten-Überschriften gruppieren. So können Tabellen beispielsweise nach Tablespaces und Partitionen gruppiert angezeigt werden. Auf diese Weise lassen sich Informationen aufgabengerecht und übersichtlich darstellen. Einmal aktivierte Ansichten bleiben bis zum Ende der Session erhalten und sind entsprechend über Tab-Reiter navigierbar.

Unterschiedliche Benutzergemeinden und ihre Privilegien

Eingangs wurde bereits darauf hingewiesen: Hora wendet sich nicht nur an DBAs, sondern hat auch Entwicklern ausgereifte Funktionalitäten zu bieten. Die zentrale Frage an dieser Stelle lautet: „Wie geht Hora mit den in

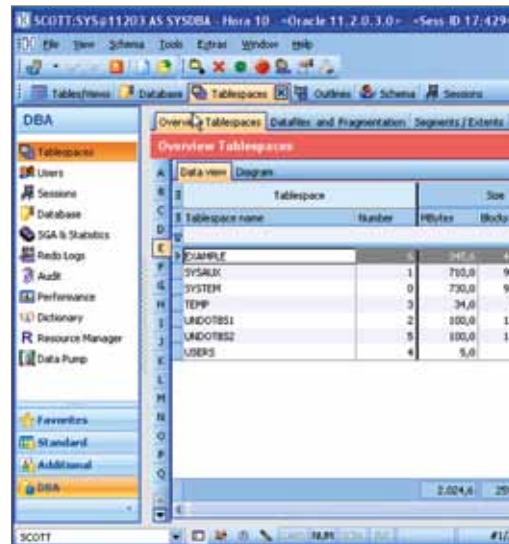


Abbildung 2: Die Navigationsbereiche – Einwahl unter „SYS“; „current schema“ steht auf „SCOTT“

der Regel höchst unterschiedlichen Privilegien-Profilen divergierender Anwendergruppen um?“ Hora bietet keine Möglichkeiten, ganze Menübäume vollständig aus- oder einzublenden, sondern orientiert sich stattdessen an den Privilegien, die der betreffende Benutzer in der Datenbank erhalten hat.

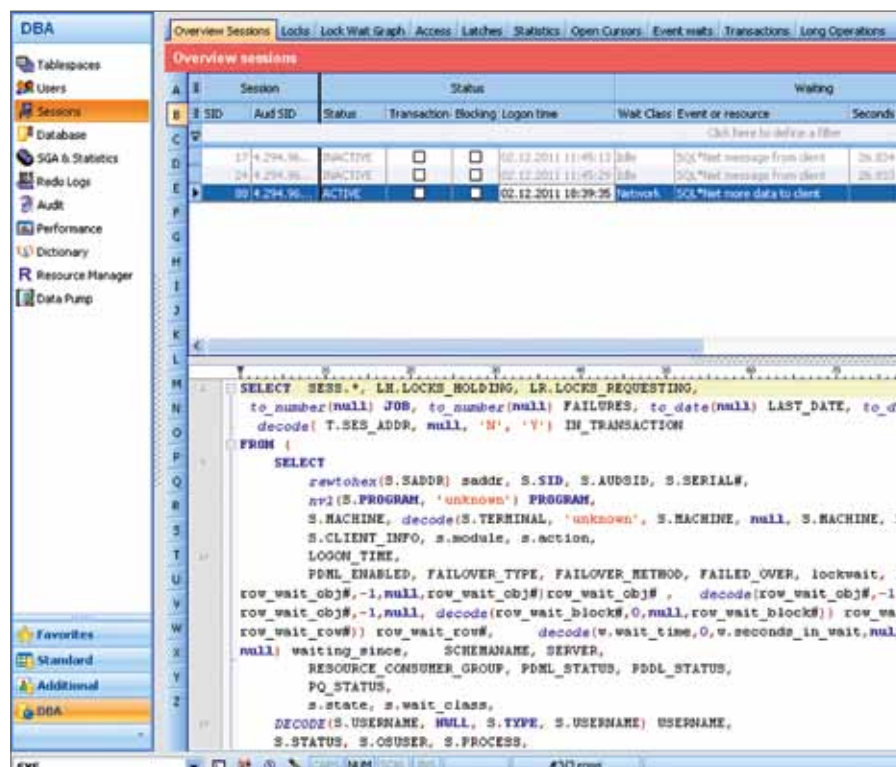


Abbildung 3: Die Session-Übersicht

Dementsprechend werden bestimmte Funktionsbereiche „ausgegraut“. Nicht privilegierte Aktionen, etwa bei direkter SQL-Eingabe, führen zu den üblichen ORA-Fehlern.

Anwender, die keine DBA-Rechte haben, trotzdem aber lesenden Zugriff auf zentrale Informationen erhalten dürfen, können mit der Rolle „select_catalog_role“ oder dem System-Privileg „select any dictionary“ ausgestattet werden. Zusätzlich bietet KeepTool die proprietäre Rolle „hora_user“ für diese Zwecke an.

DBA-Benutzern stehen sämtliche Funktionalitäten offen. Bei Schema-bezogenen Objekten wie Tabellen, Views und PL/SQL-Code kann das aktuelle Schema links unten auf dem Hauptfenster eingestellt sein. Auf diese Weise bekommt „system“ beispielsweise die Objekte von „scott“ zur Betrachtung und Bearbeitung angezeigt.

Typische Funktionsbereiche des DBA

Die Nutzungsstatistiken helfen bei der Prüfung der Lizenzvereinbarungen. Diese Übersicht ist über DBA -> Database -> Feature Usage zu erreichen und kann über „currently used“ die aktiven Einträge ausfiltern. Die Session-Übersicht (DBA -> Sessions) ist ein unverzichtbares Werkzeug beim Troubleshooting. Hier lassen sich nicht nur die geöffneten Cursor mit ihrem SQL anzeigen, sondern auch Locks, Statistiken und Wait Events darstellen. Über das Kontextmenü können Zugriffspläne (explain plan) ausgegeben werden (siehe Abbildung 3).

Das Thema „Schema-Security“ ist sehr übersichtlich dargestellt. Rollen und alle in ihrem Kontext gesetzten Privilegien können sowohl aus Sicht der Datenbank (DBA -> Database -> Roles) als auch aus Sicht des betreffenden

Benutzers (DBA -> Users -> Privileges) mit einem Blick erfasst werden (siehe Abbildung 4).

Der „Redo Log“-Bereich (DBA -> Redo Logs) zeigt nicht nur die Threads und init.ora-Parameter, sondern ermöglicht über Redo Log Changes auch eine Beurteilung der Logswitch-Frequenz, die zur Beurteilung der Dateigrößen unverzichtbar ist.

Die Performance-Ansichten geben einen Ad-hoc-Überblick über zentrale Ressourcen der Instanz und ermöglichen so eine schnelle, aber pauschale Zusammenfassung der wichtigsten Kennzahlen. Hier werden keine Historien geboten, sondern lediglich das aktuelle Geschehen dargestellt. Wer auf rückwärtige Daten und Reports des AWR zugreifen muss, ist auf die diversen SQL- und PL/SQL-Schnittstellen angewiesen, die jedoch komfortabel im SQL-Funktionsbereich in Form von Skripten vorbereitet und ausgeführt werden können.

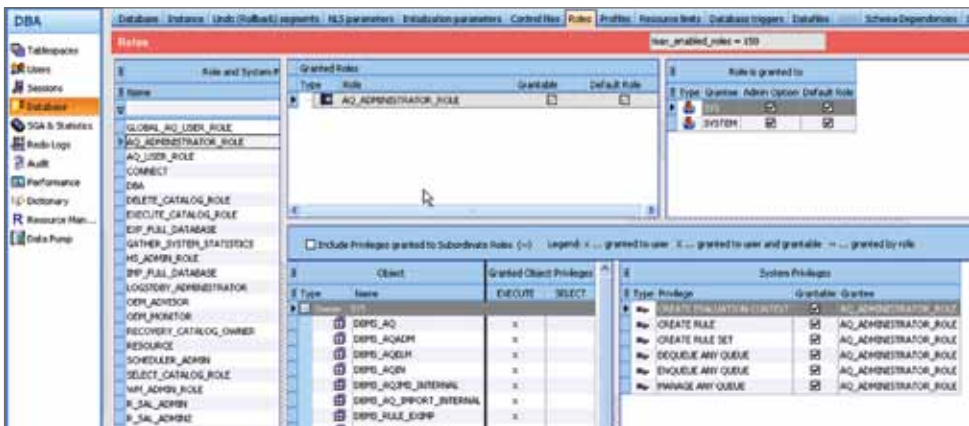


Abbildung 4: Die Rollen-Anzeige aus Sicht der Datenbank

Typische Funktionsbereiche des Entwicklers

Tabellen und Views lassen sich mit allen Kontext-Informationen, also auch Triggern und Indizes, über Standard -> Tables/Views auflisten. Änderungen sind direkt über Kontextmenüs möglich (siehe Abbildung 5).

Grundlegend überarbeitet wurde in diesem Zusammenhang auch der Statistik-Bereich, der nun Details zu den Histogrammen der betreffenden Spalten anzeigt.

Von großem Nutzen bei der Tuning-Arbeit ist die ebenfalls neue „Column Usage“ (Standard -> Tables/Views -> Columns), die Statistiken zur Nutzung von Attributen und Operatoren in Where-Klauseln ausgibt. Diese Ansicht ist ein unverzichtbares Hilfsmittel für das Design von Index-Strukturen (siehe Abbildung 6).

Master-Detail-Ansichten geben auf Knopfdruck eine Übersicht über Tabellenbeziehungen. Die Größe der Abbildung lässt sich über einen integrierten Schieberegler stufenlos anpassen (siehe Abbildung 7).

Wer SQL- oder PL/SQL-Code schreibt, wird durch die mittlerweile

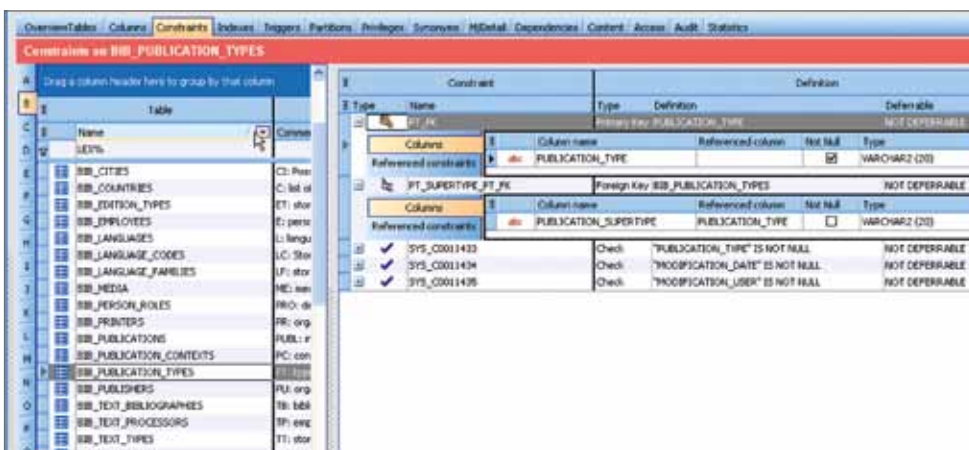


Abbildung 5: Der Tabellenbereich, hier mit der Darstellung von Constraints

üblichen Hilfsmittel unterstützt: Ein „Object Browser“ hält Objekt-Strukturen bereit, „Pretty Print“ formatiert den Code und bietet einfache und übersichtliche Formatoptionen an, ein grafischer „Query Builder“ hilft beim Aufbau der SQL-Kommandos, Code Snippets bieten Codier-Vorlagen und ein DDL-Generator erzeugt Create-Code bestehender Objekte.

Viele Oracle-Umgebungen horten ein historisch gewachsenes, oft sehr umfangreiches Reservoir an SQL-Skripten, die für den täglichen Betrieb von großer Wichtigkeit sind. In diesem Umfeld bietet der SQL-Bereich (Standard → SQL) diverse Möglichkeiten. SQL-Code kann aus einem „SQL Scratchpad“, über einen integrierten File-Explorer oder per FTP angezeigt und ausgeführt werden. Falls erforderlich, etwa bei speziellen SQL*Plus-Befehlen, kann für die Ausführung auch SQLPLUS genutzt werden.

Fazit

KeepTool ist mit der neuen Version 10 seines Flaggschiffs Hora der Devise treu geblieben, DBAs und Entwicklern, aber auch Power-Usern ein solides und überschaubares Hilfsmittel für die tägliche Arbeit mit Oracle-Datenbanken in die Hand zu geben. Die neuen Features wurden maßvoll und überlegt in die vorhandene Struktur des Werkzeugs integriert. Hora ist damit nach

Type	Name	Type	Definition	Not Null	Deferrable
Primary Key	PUBLICATION_TYPE_PK	Primary Key	PUBLICATION_TYPE	NOT NULL	NOT DEFERRABLE
Foreign Key	PUB_SUPERTYPE_FT_FK	Foreign Key	PUBLICATION_SUPER TYPE		NOT DEFERRABLE
Check	SYS_C0011433	Check	'PUBLICATION_TYPE' IS NOT NULL		NOT DEFERRABLE
Check	SYS_C0011434	Check	'PUBLICATION_DATE' IS NOT NULL		NOT DEFERRABLE
Check	SYS_C0011435	Check	'PUBLICATION_USER' IS NOT NULL		NOT DEFERRABLE

Abbildung 6: Statistiken zur Nutzung von Spalten und Operatoren in Where-Klauseln

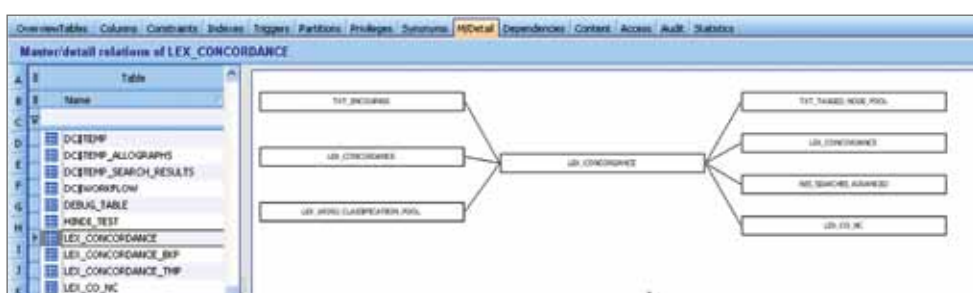


Abbildung 7: Die Master-Detail Ansicht

wie vor ein ausgereiftes und professionelles Instrument für den Einsatz in Oracle-Landschaften.

Dr. Günter Unbescheid
Database Consult GmbH
g.unbescheid@database-consult.de



Newsticker

Oracle veröffentlicht Ergebnisse des zweiten Geschäftsquartals 2012

Wie aus den veröffentlichten Zahlen hervorgeht, stieg der Umsatz im Vergleich zum Vorjahreszeitraum lediglich um zwei Prozent auf 8,8 Milliarden Dollar. Analysten hatten 9,2 Milliarden Dollar erwartet. Besonders Schlecht sieht das Ergebnis im Hardware-Geschäft aus: Der Umsatz schrumpfte in dem Bereich um 14 Prozent.

Obwohl das Software-Geschäft besser abschneidet und bei neuen Software-Lizenzen eine Umsatzsteigerung von zwei Prozent auf zwei Milliarden Dollar erzielt, verfehlte der Softwareriese eindeutig die Erwartungen der Analysten. Die Reaktionen von Oracle-Applikationskunden im deutschsprachigen Raum auf das unerwartet schwache Quartalsergebnis ihres strategischen Softwarelieferanten reichen von Gelassenheit bis hin zu ernster Besorgnis.

Trotz der ernüchternden Ergebnisse bleibt Larry Ellison zuversichtlich: Der Verkauf von Systemlösungen habe im zweiten Quartal stark zugenommen, so der Oracle-Chef in einer Pressemitteilung. Das Geschäft mit Produkten wie Exadata und Exalogic laufe gut. Die SPARC SuperCluster werden nun ausgeliefert und im dritten Quartal sollen auch die Exalytics Systeme sowie die Oracle Big Data Appliance auf den Markt kommen. Im Software-Geschäft setzt die rechte Hand von Ellison, Mark Hurd, auf Produkte wie Fusion Cloud ERP ans Cloud CRM, um ein solides organisches Wachstum zu erzielen.

Ohne Abschreibungen hätte der Gewinn laut Oracle bei 54 Cent je Aktie gelegen, erwartet worden waren 57 Cent. Die Oracle-Aktien gaben nach der Veröffentlichung der Quartalsbilanz um bis zu 9 Prozent nach. Auch die Aktien von SAP und IBM brachen ein.

Hohes Datenaufkommen ist eine große Herausforderung für moderne IT-Systeme. Operativ nicht mehr verwendete Daten in ein Data Warehouse auszulagern ist eine notwendige Maßnahme, um eine stabile Performance im System zu gewährleisten. Zudem ermöglicht es, die Datenstruktur an das hohe Datenvolumen anzupassen.

Realtime Data Warehouse: Auslagern und Wiederherstellen operativer Daten

Steffen Kantorek, Berenberg Bank

Doch was kann man tun, wenn die ausgelagerten Informationen auch geraume Zeit später nochmals benötigt und bearbeitet werden müssen? Dieser Artikel beschreibt eine Lösung, um die Datenlast in einem operativen System gering zu halten, ohne dabei die Verfügbarkeit der Daten zu verlieren.

Es wird eine Anwendung mit einem durchschnittlichen Datenaufkommen von einer Million Datensätzen täglich betrachtet. Dabei werden ein Prozent, also zehntausend Datensätze, nachträglich geändert. 95 Prozent dieser Änderungen finden innerhalb der ersten sieben Tage statt. Es besteht jedoch in wenigen Fällen der Bedarf, auch später noch Änderungen an den Daten vorzunehmen.

Da die Daten aus dem operativen System in jedem Fall in das Data

Warehouse übertragen werden, bietet es sich an, dies als Ausgangspunkt für eine Wiederherstellung der Daten zu verwenden. Die Idee hierbei ist, die Vorteile beider Systeme auszunutzen: zum einen die Geschwindigkeit des operativen Systems zur Datenverarbeitung und zum anderen die beim Umgang mit großen Datenmengen günstige Architektur des Data Warehouse.

Die Herausforderung besteht darin, einen Prozess zu schaffen, der es ermöglicht, die ausgelagerten Daten wiederherzustellen. Jedoch steht zum Anfang der Datenverarbeitung nicht fest, welche der Daten nochmals bearbeitet werden müssen. Der Prozess muss somit die Wiederherstellung der Daten zu einem beliebigen Zeitpunkt gewährleisten.

Hierzu wurden zwei getrennte Systeme aufgebaut: das operativ laufende

System, das die Daten empfängt und verarbeitet, sowie das Data Warehouse, das als Speicherort für historische Daten dient und zu Auswertungszwecken benutzt wird. Da beide Systeme unabhängig voneinander arbeiten sollen, muss hier eine strikte System-Trennung vorliegen. Das bedeutet, dass ein Ausfall oder Performance-Problem eines Systems keine Auswirkungen auf das andere System haben darf.

Realtime-Auslagerung in das Data Warehouse

Bei der Auslagerung der Daten werden diese mittels Oracle Streams an das Data Warehouse übertragen. Dieser Prozess findet in Realtime statt. Hierzu werden die Daten auf dem operativen System durch einen Capture-Prozess

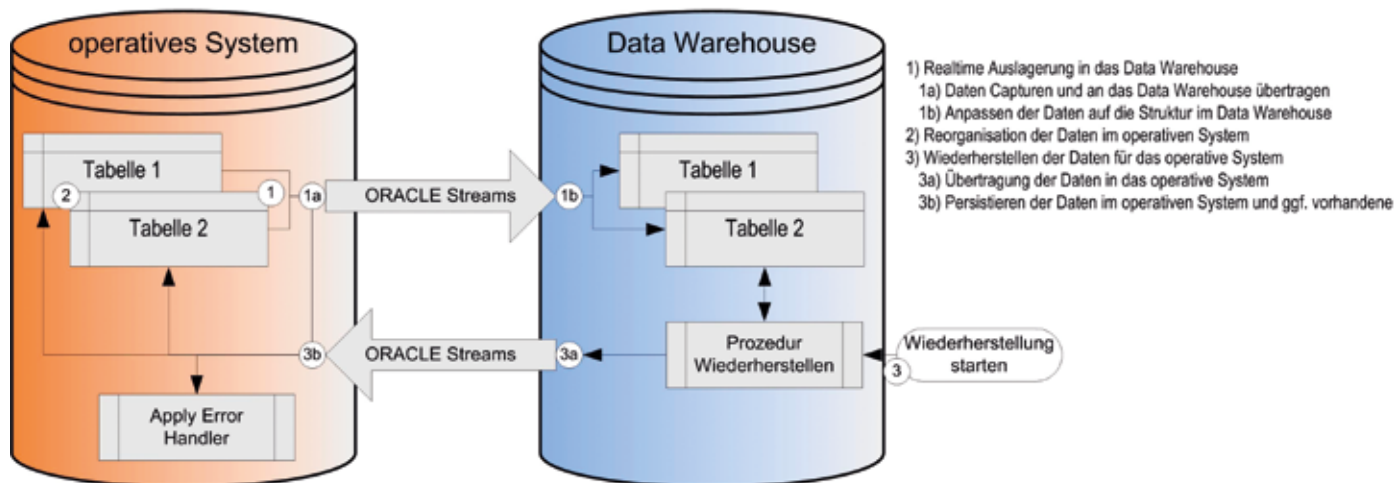


Abbildung 1: Schematischer Gesamtprozess, Auslagern und Wiederherstellen von Daten

in einen Logical Change Record (LCR) verpackt und per Queuing an das Data Warehouse übertragen. Oracle bietet verschiedene Möglichkeiten, um den Capture-Prozess aufzusetzen. Zum einen das klassische Capture über die Redo-Logs oder neu in Oracle 11g als synchrones Capture. Aber auch ein Capture über einen Trigger, der einen LCR erzeugt, ist denkbar.

Aufgrund der erweiterten Datenstruktur des Data Warehouse wird beim Apply des LCR die Datenstruktur angepasst (denormalisiert). Dafür befinden sich Trigger auf den Zieltabellen im Data Warehouse, die bei einer DML-Operation durch einen LCR starten. Im Trigger werden Foreign-Key-Beziehungen aufgelöst, um die Zugriffszeiten von Auswertungen auf historische Stammdaten zu minimieren. Hierfür sind die Firing Properties des Triggers mittels „DBMS_DDL“-Package angepasst. Ist das Merkmal „Firing Once“ auf „false“ gesetzt, startet der Trigger immer, auch beim Einarbeiten eines LCR. Durch die Realtime-Übertragung der Daten an das Data Warehouse befindet sich jeweils auf beiden Systemen der aktuelle Datenstand.

Reorganisieren der Daten im operativen System

Um die Performance im operativen System stabil zu halten, sind die Daten zu reorganisieren. Aufgrund der Realtime-Übertragung der Daten zum Data Warehouse kann dies zu einem frei wählbaren Zeitpunkt am Ende der Datenverarbeitung erfolgen.

Da der Großteil der nachträglichen Änderungen innerhalb der ersten sieben Tage nach Erstellung der Daten erfolgt, startet nach dieser Zeit auch der Reorganisations-Prozess. Alle verarbeiteten Daten, die somit älter als sieben Tage sind, werden aus dem operativen System entfernt. Dabei ist allerdings zu beachten, das Löschungen nicht im Capture-Prozess berücksichtigt und somit nicht zum Data Warehouse übertragen werden; dies geschieht durch die Definition einer Regel, die besagt, dass der Capture-Prozess keine „DELETE“-Operation aufnehmen soll.

```

SQL> DESC UMSATZ /*Data Warehouse*/
Name          Type          Nullable Default Comments
-----
UMSATZ_ID     NUMBER
KUNDE_ID      NUMBER        Y          Foreign Key auf hr.Kunde
BETRAG        NUMBER(14,2)  Y          Gesamtumsatz des Kunden
KUNDE_NAME    VARCHAR2(200) Y          Kundenname

SQL>
...
CREATE OR REPLACE PROCEDURE DATEN_WIEDERHERSTELLEN(IN_UMSATZ_ID IN NUMBER)
IS
  MY_MSG_HANDLE RAW(16);
  REC           hr.UMSATZ%ROWTYPE;
  LCR           SYS.LCR$_ROW_RECORD;
  ROW_LIST      SYS.LCR$_ROW_LIST := SYS.LCR$_ROW_LIST();
  MY_ENQ_OPT    SYS.DBMS_AQ.ENQUEUE_OPTIONS_T;
  MY_MSG_PROP   SYS.DBMS_AQ.MESSAGE_PROPERTIES_T;
BEGIN
  -- Datensatz selektieren
  SELECT *
  INTO rec
  FROM hr.UMSATZ u
  WHERE u.UMSATZ_id = IN_UMSATZ_ID;

  -- LCR aufbauen
  ROW_LIST := SYS.LCR$_ROW_LIST();
  ROW_LIST.EXTEND(3);
  ROW_LIST(1) := SYS.LCR$_ROW_UNIT(,UMSATZ_ID',SYS.ANYDATA.
  CONVERTNUMBER(rec.UMSATZ_ID), NULL, NULL, NULL);
  ROW_LIST(2) := SYS.LCR$_ROW_UNIT(,KUNDE_ID',SYS.ANYDATA.
  CONVERTNUMBER(rec.KUNDE_ID) , NULL, NULL, NULL);
  ROW_LIST(3) := SYS.LCR$_ROW_UNIT(,BETRAG' ,SYS.ANYDATA.
  CONVERTNUMBER(rec.BETRAG) , NULL, NULL, NULL);

  LCR := SYS.LCR$_ROW_RECORD.CONSTRUCT(source_database_name => SYS_
  CONTEXT(,USERENV', 'DB_NAME')
                                     , command_type      => ,INSERT'
                                     , object_owner       => ,HR'
                                     , object_name        => ,UMSATZ');

  LCR.SET_VALUES(,new', ROW_LIST);

  /*...
  Streams/ Queuing spezifische Werte setzen
  ... */

  -- Eisntellen in die Queue
  DBMS_AQ.Enqueue(Queue_name => ,HR.AQ_WAREHOUSE_OUTBOX'
                 , ENQUEUE_OPTIONS => MY_ENQ_OPT
                 , MESSAGE_PROPERTIES => MY_MSG_PROP
                 , PAYLOAD          => SYS.ANYDATA.CONVERTOBJECT(lcr)
                 , MSGID            => MY_MSG_HANDLE );
END;

```

Listing 1: Description der Umsatz-Tabelle auf dem Data Warehouse und Capture-Prozedur. Die Spalte „KUNDE_NAME“ wird nicht an das operative System übertragen.

Wiederherstellen der Daten für das operative System

Werden ausgelagerte Daten im operativen System wieder benötigt, wird dieser

Prozess manuell angestoßen. Die Daten aus dem Data Warehouse müssen nun in das operative System übertragen werden. Dieser manuelle Vorgang ändert die Daten im Data Warehouse nicht,

sodass kein Capture-Prozess die Daten aufnehmen und verarbeiten kann. Stattdessen wird eine Prozedur aufgerufen, die aus den Daten einen LCR erzeugt. Der Aufbau des LCR erfolgt in der Struktur des Zielsystems, es werden somit nur die Daten übermittelt, die auch dem operativen System zur Verfügung stehen (siehe Listing 1).

An dieser Stelle ist auch darauf zu achten, dass abhängige Daten auf dem operativen System in der korrekten Reihenfolge wiederhergestellt werden müssen, um Constraint-Verletzungen zu verhindern. Anschließend wird der LCR via Queuing an das operative System übermittelt, in dem die Daten persistiert sind. Eine erneute Übertragung der Daten in das Data Warehouse findet aufgrund definierter Regeln im Capture-Prozess nicht statt.

Um mögliche Fehler beim Wiederherstellen der Daten auf dem operativen System besser zu verwalten, kann man einen benutzerdefinierten ErrorHandler für den Apply-Prozess hinterlegen (siehe Listing 2). Das Wiederherstellen eines im operativen System bestehenden Datensatzes würde eine Primary-Key-Verletzung im operativen System verursachen und den LCR in eine Exception Queue legen, um dort erneut verarbeitet werden zu können. Da dies aber aufgrund der Daten-Redundanz nicht notwendig ist, kann man einen Error-Handler definieren, der solche Fehler protokolliert, jedoch ein Speichern des LCR in die Exception Queue verhindert.

Fazit

Die vorgestellte Lösung zum Wiederherstellen von Daten wird bereits erfolgreich in der Berenberg Bank eingesetzt und stellt eine komfortable Möglichkeit dar, ausgelagerte Daten nachträglich zu bearbeiten. Darüber hinaus stellt Oracle mit Streams ein Werkzeug zur Verfügung, das die Replikation von Daten zwischen homogenen Systemen vereinfacht, auch bei abweichenden Datenstrukturen auf den Systemen. Es sollte jedoch auch die Datenmenge betrachtet werden, die zwischen den Systemen innerhalb einer Transaktion repliziert wird. So kann es bei großen Datenmen-

```
CREATE OR REPLACE PROCEDURE USER_DML_ERROR_HANDLER( message
IN ANYDATA
, error_stack_depth
IN NUMBER
, error_numbers
IN DBMS_UTILITY.NUMBER_ARRAY
, error_messages
IN DBMS_UTILITY.LNAME_ARRAY )
IS
lcr SYS.LCR$_ROW_RECORD;
v_info VARCHAR2(4000);
BEGIN
-- Zugriff auf den LCR
IF message.GetTypeName = ,SYS.LCR$_ROW_RECORD'
AND message.GETOBJECT(lcr) = DBMS_TYPES.SUCCESS
THEN
IF error_numbers(1) = 1 -- UNIQUE CONSTRAINT verletzung
AND lcr.GET_OBJECT_OWNER = ,HR'
AND lcr.GET_COMMAND_TYPE = ,INSERT'
AND lcr.GET_OBJECT_NAME = ,UMSATZ'
THEN
/*Informationen fuer die Log Tabelle sammeln*/
v_info := ,Error:'
||CHR(10)||'Tabelle= HR.UMSATZ'
||CHR(10)||'UMSATZ_ID= , ||lcr.GET_VALUE( value_
type =>'NEW'
, column_
name =>'UMSATZ_ID' ).ACCESSNUMBER;

/*...*/
END IF;
END IF;
/*...*/
EXCEPTION
WHEN OTHERS THEN
RAISE;
END;
...
BEGIN
DBMS_APPLY_ADM.SET_DML_HANDLER(OBJECT_NAME => ,HR.UMSATZ'
,OBJECT_TYPE => ,TABLE'
,OPERATION_NAME => ,INSERT'
,ERROR_HANDLER => TRUE
,USER_PROCEDURE => ,USER_DML_ERROR_
HANDLER'
,APPLY_NAME => ,APPLY_WAREHOUSE_
INBOX'
);
END;
```

Listing 2: Definition eines DML-Error-Handlers für Inserts auf die Tabelle „HR.UMSATZ“. Der Error-Handler wird bei fehlerhaften Insert-Operationen des Apply-Prozesses „APPLY_WAREHOUSE_INBOX“ aufgerufen

gen vorkommen, dass die Replikierung sehr lange dauert. Man sollte sich also bereits im Vorfeld überlegen, welche Daten man wirklich benötigt.

Steffen Kantorek
Berenberg Bank
steffen.kantorek@berenberg.de



Die Herausforderung beim Versicherungsverbund Die Continentale bestand darin, einen Oracle 10g R2 Zwei-Knoten-„Extended-Distance“-RAC mit möglichst geringer Downtime auf einen 11g 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 10g R2 auf 11g 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 10g 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 11g R2
- Drei Datenbanken 11g R2 (11.2.0.2.0)
- Eine Datenbank 10g 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 Disaster-Fall (die zwei RACs stehen in den-

selben 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 11g 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 10g R2 auf 11g 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 10g R2 laufen musste und nicht auf 11g R2 aktualisiert wurde, hängt mit dem Oracle Warehouse Builder (OWB) zusammen. Dessen Kernfunktionalität ist seit 10g 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 11g 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

eingrichtet 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.

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 Disaster-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.

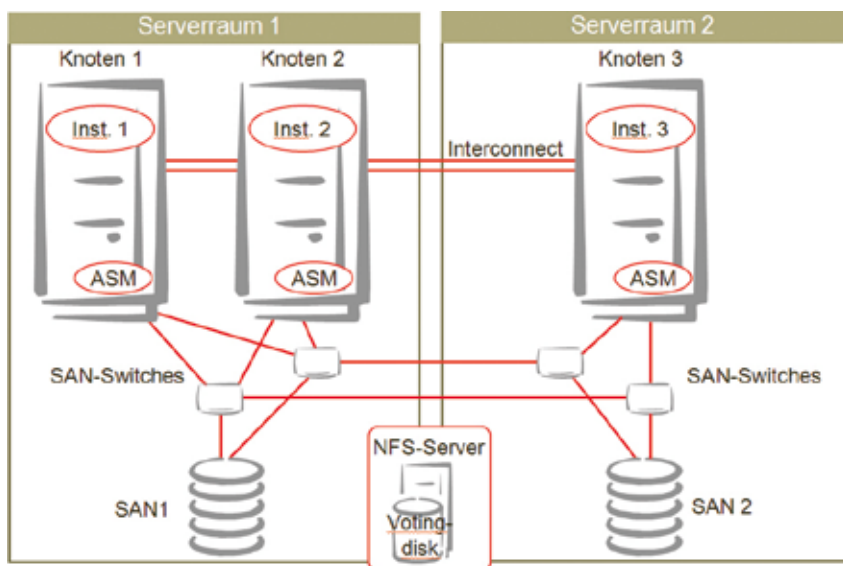


Abbildung 1: Extended Distance RAC

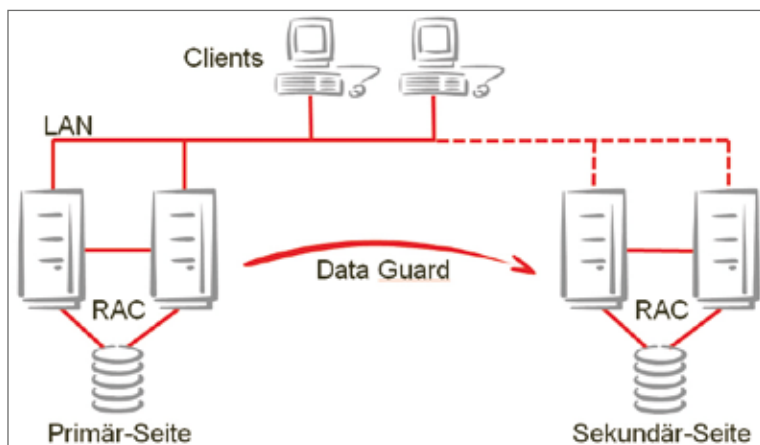


Abbildung 2: Maximum Availability Architecture (MAA)

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 11g R2 migrierten Datenbanken nach folgendem Vorgehen ab:

1. Der neue Drei-Knoten-Extended-Distance-Cluster mit Grid-Infrastruktur-Software Version 11g R2 wurde aufgesetzt.
2. Die Oracle-Datenbank-Software wurde sowohl in Version 10g R2 als auch 11g R2 auf diesem RAC installiert, ohne eine Datenbank anzulegen.
3. Die Physical-Standby-Datenbank mit 10g 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 11g 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 Ver-

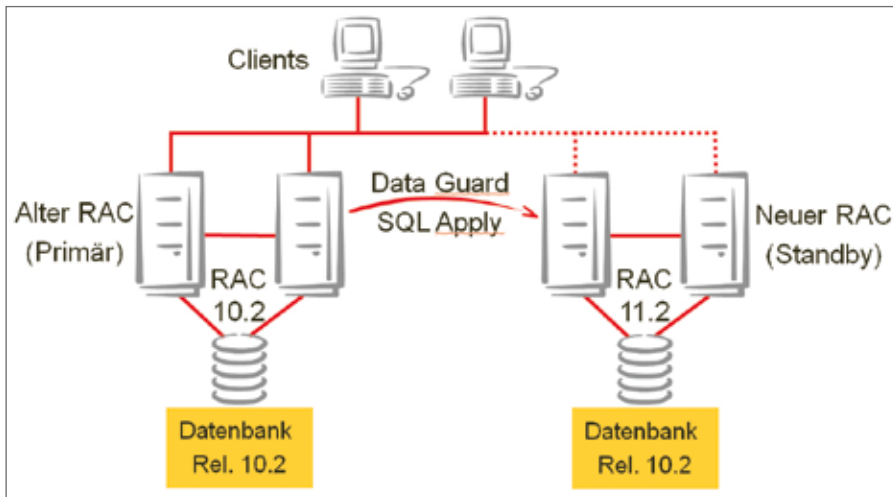


Abbildung 3: Data-Guard-Konfiguration vor dem Upgrade

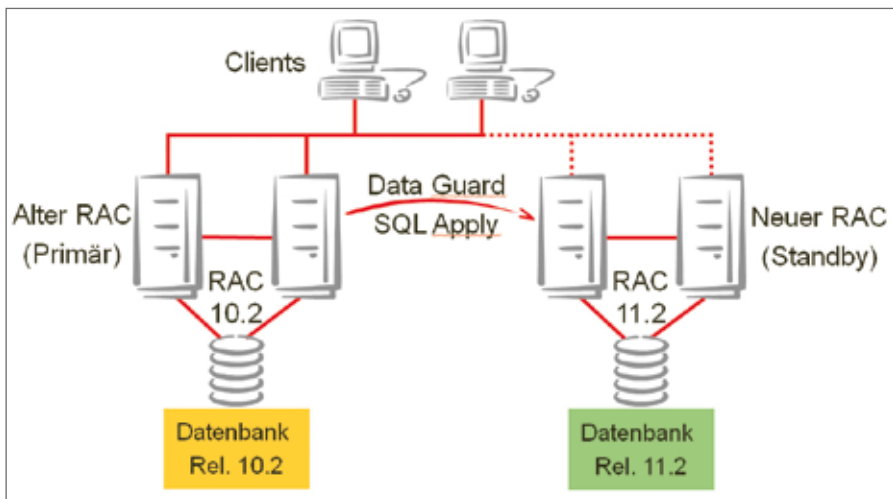


Abbildung 4: Data Guard mit gemischtem Release

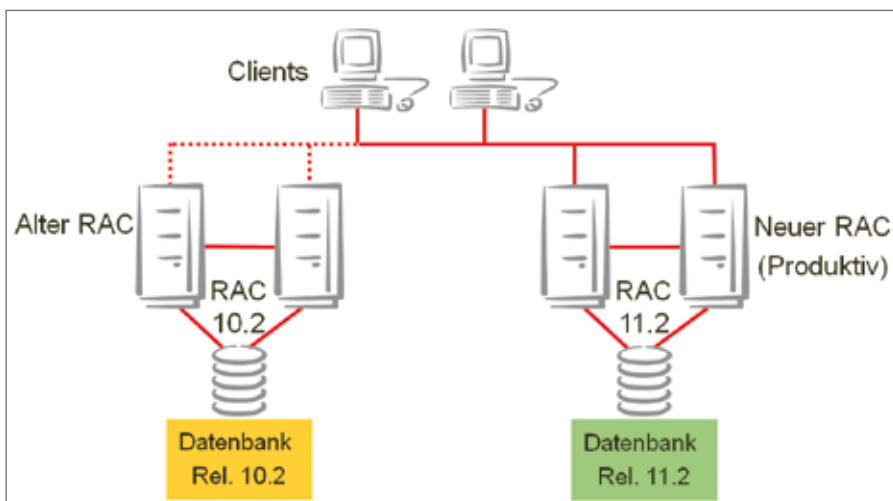


Abbildung 5: Data-Guard-Konfiguration nach dem Failover

sion auf die alte Datenbank mit der alten Version ist nicht mehr möglich (siehe Abbildung 5).

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

Softwarestand 10g 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 Neubestimmung. 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 11g (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-Applly 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 schmerzhaft Erfahrung, dass bei diesem Konzept ein „Switchover“ zu einem Versionsabweichungs-Fehler führte.

Ebenso schmerzhaft war die Erfahrung, dass nach der Installation der Grid Infrastructure 11g 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 11g R2. Dabei sind die DB-Instanzen bei „policy managed“ nicht mehr fest an die Cluster-Knoten gebunden. Für 10g-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 10g-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 11g das „duplicate“-Kommando auch ohne Backups, also direkt aus der Primär-Datenbank durchführen. Für eine 10g-Datenbank funktioniert das allerdings noch nicht. Die kritischste Situation beim Aufbau der Physical-Standby-Datenbanken war stets die Frage, ob der Log-Applly 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-Applly konnten Fehlermeldungen mit „ORA-00313“ ignoriert werden, die sich auf noch fehlende Online-Logs bezogen. Die fehlenden Online-Logs wurden beim Start des Log-Applly 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-Applly der kritische Augenblick. Insbesondere der oben empfohlene Wert „use_db_recovery_file_dest“ im Parameter „log_archive_dest_n“ verhinderte den Start des SQL-Applly. Daher musste hier doch temporär bis nach dem 11g-Upgrade der komplette Pfad des entsprechenden ASM-Verzeichnisses eingetragen werden. Nach einem Stromausfall lief der SQL-Applly 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 11g-Upgrade auf der Standby-Seite gab es zunächst Verwirrung dadurch, dass die Alertlogs nicht

Libelle BusinessShadow®

Unabhängig bezüglich

- Fehlerursache
- Entfernung
- Hardware / Architektur
- Komplexer Systeme

Schnelle Arbeitsaufnahme

- Mit konsistenten Daten
- Auf Knopfdruck
- Automatisiert
- ...

Hans-Joachim Krüger
Chief Technology Officer
Libelle AG

**Recovery ohne Restore.
Konsistent. Per Knopfdruck.**

Mit Libelle BusinessShadow®

Mehr erfahren:

www.libelle.com/business



ORACLE Gold Partner

Libelle

Libelle AG

Gewerestr. 42 • 70565 Stuttgart, Germany
T +49 711 / 78335-0 • F +49 711 / 78335-148
www.libelle.com • sales@libelle.com

Ausfallzeiten:

- **DB:** ca. ½ h
(Failover, Log-Kontrolle, "Dataguard abbauen", ...)
- **Services:** ca. 1 ½ h
(stop/start DB, init-Par., DBNEWID bei 10g, ...)
- **Anwendungen:** ca. 3 h
(Stop/Start der Applicationserver)



Abbildung 6: Zeitstrahl Ausfallzeiten

dort zu finden waren, wo sie vermutet wurden. Die Ursache war der Versionsunterschied der Prozedur „oraenv“ in 10g / 11g, 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 Ser-

vices 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.

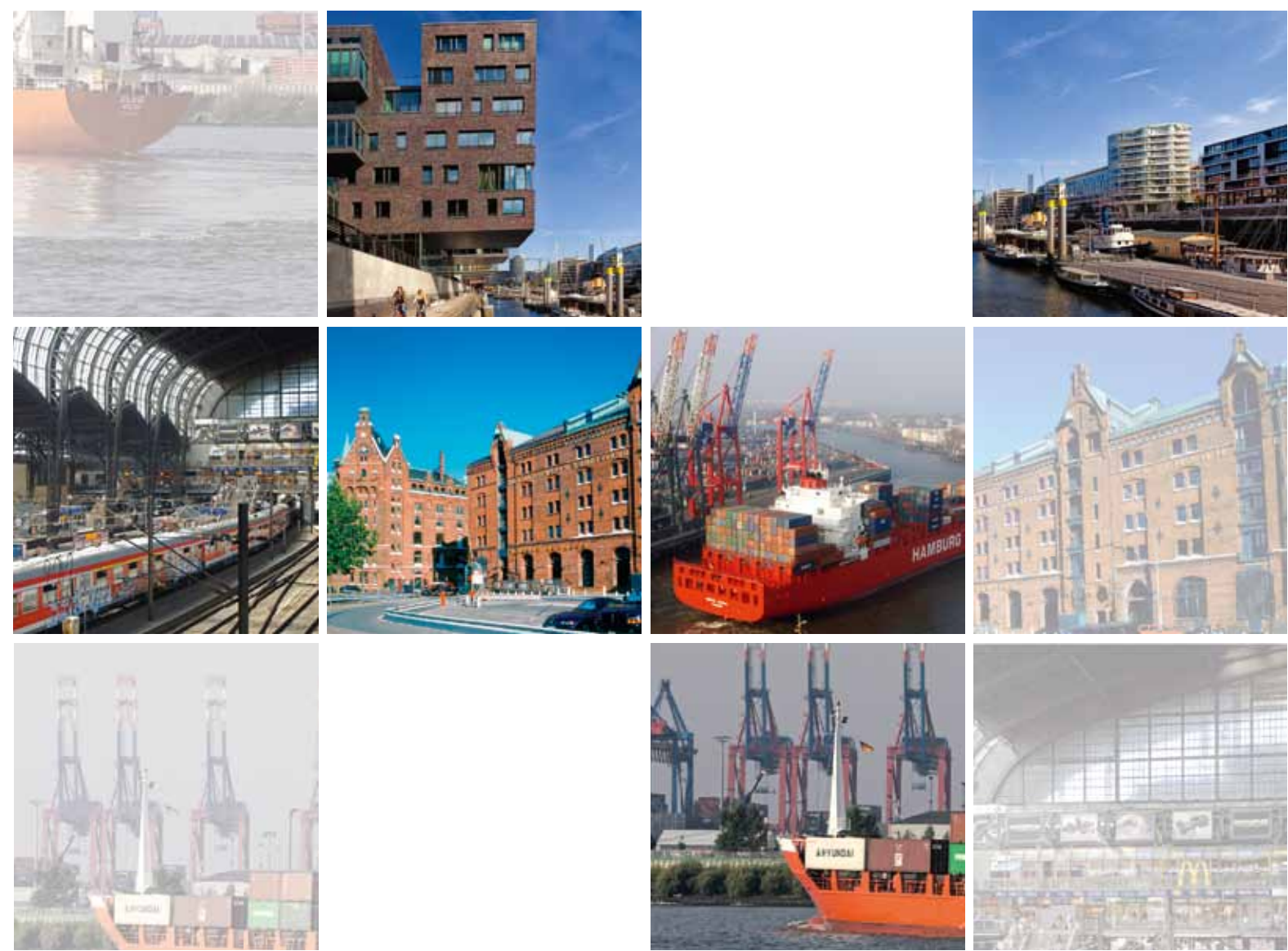
Schließlich ist erwähnenswert, dass nach Aussage der Kollegen die Migration einen nicht unerheblichen Performance-Gewinn gebracht hat. Dies ist allerdings bisher nicht im Einzelnen durchgemessen worden.

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



Ralf Appelbaum
TEAM GmbH
ra@team-pb.de





DOAG 2012 Logistik + SCM

5. Juni 2012

DIE HIGHLIGHTS

- Hochkarätige Keynote-Speaker
- Spannende praxisnahe Vorträge von erfahrenen Referenten
- Innovative Systeme, Werkzeuge und Technologien werden präsentiert
- Parallele Workshops geben Einblicke in Produkte und Lösungen
- Mehr als 100 erwartete Teilnehmer
- Begleitende Ausstellung mit einer Vielzahl Anbieter
- Gelegenheit für intensive Diskussionen und Networking
- Spannendes Vorabendprogramm im Miniaturwunderland Hamburg
- Die Speicherstadt Hamburg als attraktiver Veranstaltungsstandort

In Kooperation mit



WEITERE INFOS

DOAG Deutsche ORACLE-Anwendergruppe e.V.
 Tempelhofer Weg 64 | D-12347 Berlin
 Tel: 0700-11362438 | Fax: 0700-11362439 | office@doag.org

<http://bs.doag.org/go/logistik2012>

Die Oracle Database Appliance oder kurz „ODA“ ist ein neues, ganzheitliches System aus Hard- und Software, das die Inbetriebnahme, die Wartung, den Betrieb und den Support hochverfügbarer Oracle-Datenbank-Lösungen extrem vereinfacht. Der Artikel beleuchtet die besonderen Eigenschaften und dokumentiert erste Praxiserfahrungen des sogenannten „One Button Install“.

Oracle Database Appliance – Überblick und erste Praxis-Erfahrungen

Hinrich Tobaben, ORACLE Deutschland B.V. & Co. KG

Im September 2011 wurde die ODA im Rahmen einer Web-Konferenz der Öffentlichkeit vorgestellt (siehe Abbildung 1). Neben der Exadata ist sie ein weiteres sogenanntes „Engineered System“, das speziell für den Einsatz der Oracle-Datenbank konzipiert wurde. Exadata und ODA adressieren jedoch unterschiedliche Anforderungen. Bei der Exadata liegt der Fokus eher auf maximaler Performance und sehr großen Installationen. Selbst die kleinste Ausbaustufe einer Exadata, das Quarter Rack, ist für kleinere Installationen, die diese extreme Leistung nicht benötigen, bereits überdimensioniert. ODA hingegen ist für Kunden geeignet, die zwar eine hochverfügbare Datenbank-Lösung suchen, aber nicht das Know-how haben, eine solche Lösung zu installieren und zu betreiben. Das Hauptziel ist die Reduktion der Komplexität eines hochverfügbaren Datenbank-Clusters hinsichtlich Installation und Betrieb. Aber auch an eine gute Performance wurde gedacht, da alle

System- und Datenbank-Parameter auf Best-Practice-Erfahrungen basieren und für die ODA-Hardware optimiert wurden. Durch neue lizenzrechtliche Möglichkeiten bietet die ODA insbesondere für kleine Datenbank-Umgebungen einen interessanten Ansatz, da eine Lizenzierung von 2 bis 24 Prozessor-Kernen (siehe unten) möglich ist.

Besondere technische Eigenschaften

Bei der Hardware für die ODA handelt es sich um einen sogenannten „Cluster in a Box“, in dem alle Komponenten redundant ausgelegt sind. In dem nur vier Höheneinheiten (knapp 18 cm) hohen Gehäuse sind zwei Intel-basierende Serverknoten untergebracht, die unabhängig voneinander arbeiten. Jeder dieser Knoten hat zwölf Kerne, aufgeteilt auf zwei Sockel, 96 GB Hauptspeicher, einen Service-Prozessor (ILOM) sowie zwei interne gespiegelte Bootplatten.

Als externe Netzwerk-Schnittstellen werden jeweils sechsmal 1 GbE

und zweimal 10 GbE zur Verfügung gestellt. Bis zu vier Bonds im Active-Backup-Modus sind möglich. Für die redundante Cluster-Kommunikation existieren pro Knoten noch zusätzlich zweimal 1-GbE-Schnittstellen, die intern verdrahtet sind und daher keine externen Kabel benötigen.

Alle gemeinsamen Platten des Clusters sind von vorn erreichbar und können leicht getauscht werden. Es handelt sich hierbei um zwanzig „dual ported“ SAS-Festplatten mit je 600 GB, die mit dem Automatic Storage Management (ASM) von Oracle dreifach gespiegelt sind. Hinzu kommen vier Solid-State-Disks (SSD) mit jeweils 73 GB, auf denen die Online Redologs der Datenbanken gespeichert sind. Auch dieser Bereich wird mit ASM dreifach gespiegelt. Die Netzteile und die Lüfter sind ebenfalls redundant ausgelegt und können im laufenden Betrieb getauscht werden. Abbildung 2 zeigt grob die Architektur der Hardware.

Wie bereits erwähnt, verfolgt Oracle mit der ODA das Hauptziel, die Komplexität eines Datenbank-Clusters zu reduzieren. Um dies zu erreichen, wurden unter anderem folgende Maßnahmen ergriffen beziehungsweise Einschränkungen festgelegt:

- Definition einer festen Hardware-Konfiguration (Cluster in a Box), die nicht verändert werden darf
- Vorinstallation einer vordefinierten Oracle-Linux-5.5-Version, in der keine Pakete entfernt oder zugefügt werden dürfen
- Vorinstallation eines speziell für ODA entwickelten Appliance



Abbildung 1: Oracle Database Appliance

Managers mit dazugehörigem Command Line Interface (oakcli), der die Installation des kompletten Systems innerhalb von zwei Stunden erledigt und den Betrieb wesentlich vereinfacht

- Definition einer vorgegebenen Version für die Grid- und Datenbank-Software (downloadbar als spezielles Bündel für die ODA über support.oracle.com)
- Definition von speziellen Bundle-Patches, die alle Patches (Firmware, Linux, Clusterware, ASM, Datenbank) für die ODA zusammenfassen
- Definition einer Whitelist für Agenten (etwa für Backup), die für die ODA qualifiziert wurden und installiert werden dürfen (zu finden auf der ODA-Seite im OTN)
- Vordefinition eines festen ASM-Storage-Layouts, basierend auf Best-Practice-Erfahrungen, das nicht geändert werden darf
- Vordefinition von System- und Datenbank-Parametern, basierend auf Best-Practice-Erfahrungen
- Vordefinition verschiedener Datenbank-Templates von „Very Small“ bis „Very Large“

„Pay-As-You-Grow“-Lizenzmodell

Für die ODA existiert ein einmaliges Lizenzmodell nach dem Prinzip „Pay-As-You-Grow“. Die Anzahl der Kerne, die zu lizenzieren sind, können sich am Bedarf orientieren. Als Voraussetzung für diese Art der Lizenzierung dient ein besonderes Verfahren zur Deaktivierung von Kernen (CPU Cores by Demand). Dabei ist Folgendes zu beachten:

- Alle 24 Kerne sind aktiviert, wenn die ODA ausgeliefert wird. Falls man das nicht einschränken möchte, muss man nichts unternehmen
- Wenn Kerne deaktiviert werden sollen, fordert man via MOS einen speziellen Key an. Dieser wird dann dem Kommando „oakcli apply core_configuration_key /tmp/myfile.txt“ übergeben. Die entsprechenden BIOS-Einstellungen werden vorgenommen und nach einem Reboot sind dann die Kerne deaktiviert, die nicht verwendet werden sollen.

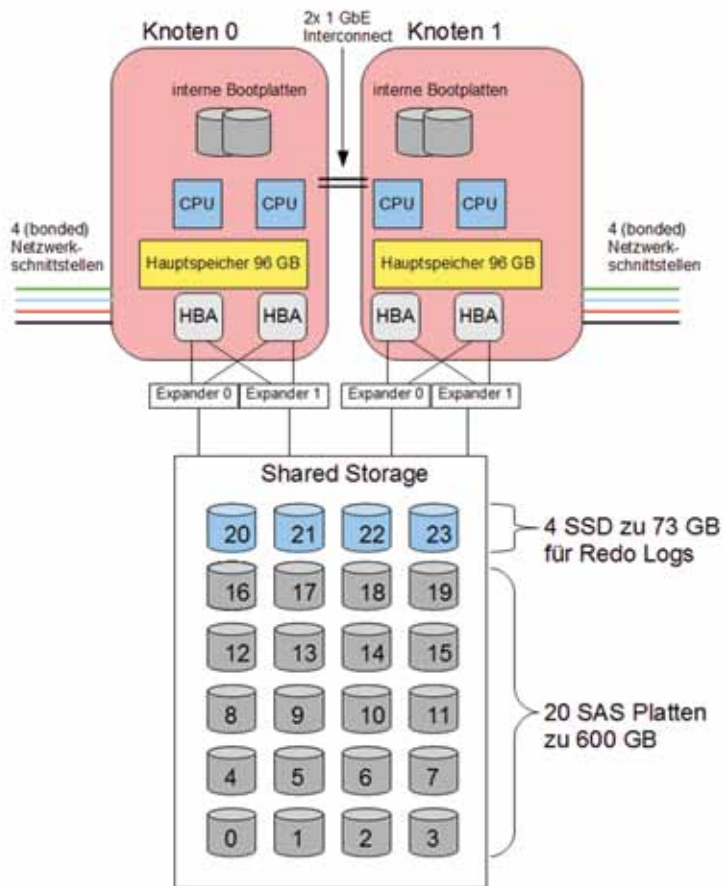


Abbildung 2: Architektur der Hardware

- Es müssen mindestens vier Kerne aktiviert sein (zwei Kerne pro Server-Knoten). Auch bei dieser minimalen Konfiguration hat die ODA Zugriff auf den gesamten Hauptspeicher und auf alle Speicherbereiche.
- Es können später zusätzliche Kerne aktiviert werden. Das Inkrement muss dann allerdings 4, 8, 12 oder 16 betragen.
- Die Anzahl der Kerne darf später nicht mehr reduziert werden, weil nur eine einmalige Deaktivierung erlaubt ist
- Die Anzahl der aktivierten Kerne ist pro Server-Knoten immer gleich
- Die Deaktivierung von Kernen auf BIOS-Ebene wird nur bei der ODA als Hard-Partitioning anerkannt
- Wenn für die Enterprise Edition und für RAC One Node nur ein Knoten benutzt werden soll und der andere nur als Failover-Knoten für weniger als zehn Tage im Jahr zum Einsatz kommt, braucht man nur die Hälfte der aktivierten Kerne zu lizenzieren (siehe Abbildung 3)

Erste Praxiserfahrungen

Der Autor möchte vorausschicken, dass er vorher noch nie einen RAC-Cluster aufgesetzt habe und auch von Clusterware und ASM so gut wie keine Vorkenntnisse besitze. Daher entspricht sein Wissensstand in etwa dem der möglichen ODA-Zielgruppe. Die erste ODA in Deutschland hatte im November 2011 einen Value Added

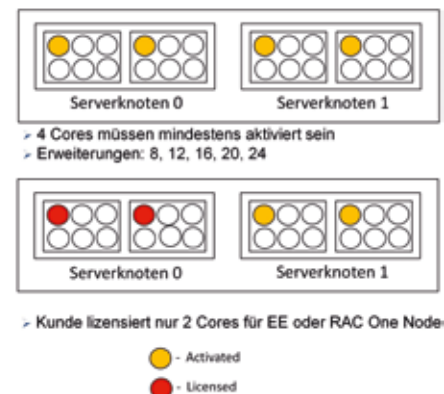


Abbildung 3: Lizenzbeispiel

Distributor (VAD) von Oracle bekommen. Der Autor hat sich zusammen mit einem Mitarbeiter des VAD dann praktisch mit der ODA beschäftigt.

Auch wenn es sich bei der ODA um eine Appliance handelt, sind die Vorbereitungen sorgfältig zu erledigen. Insbesondere wenn man die ODA in einem einfachen Testlabor aufbaut, sollte man daran denken, dass auch dort ein DNS-Server verfügbar ist, sonst geht in der aktuellen Version gar nichts und man verliert unnötige Zeit. Die Empfehlung des Autors lautet, die gesamte Netzwerk-Struktur im Vorfeld mit dem Offline Configurator zu testen. Dieser läuft als Java-Anwendung unter Windows, Linux sowie OSX und kann von der ODA-Seite im OTN heruntergeladen werden. Er empfiehlt sogar, die gesamte Konfiguration vorab mit dem Offline Configurator zu erstellen und dann als Datei abzuspeichern. Diese Datei kann man dann später auf den Serverknoten 0 transferieren und danach vor dem Deployment in den Appliance Manager importieren.

Bevor jedoch das Deployment gestartet werden kann, muss noch das sogenannte „End User Bundle“ (Grid- und Datenbank-Software) via MOS als Patch „12978712“ heruntergeladen und auf die ODA nach „/tmp“ trans-

feriert werden. Das Bundle kommt als Zip-Datei und wird mit dem Kommando „oakcli unpack -package /tmp/p12978712_21000_Linux-x86-64.zip“ auf der ODA ausgepackt und dabei an den richtigen Ort kopiert (siehe auch MOS-Note 1373617.1). Jetzt kann das Deployment gestartet werden und die eigentliche Installation des Datenbank-Clusters beginnen. Wenn alle Vorbereitungen korrekt erledigt worden sind, läuft das Deployment ohne weitere Rückfragen durch und dauert für einen RAC-Cluster etwa zwei Stunden. Dabei ist immer genau angezeigt, was gerade gemacht wird und wie viel Prozent schon erledigt sind. Nach Fertigstellung kann man sich ganz normal via „DB Control“ anmelden und die üblichen Einstellungen vornehmen sowie Daten importieren. Die Passwörter für root etc. lassen sich jetzt ändern.

Fazit

Zusammenfassend lässt sich sagen, dass der Autor mit nur sehr wenigen Vorkenntnissen in der Lage war, auf der ODA einen RAC-Cluster zu installieren. Das reine Deployment ist so etwas wie ein „Single Button Install“. Um die Installation vorzubereiten, sind jedoch einige wenige CLI Kommandos notwen-

dig. In seinen Augen ist das allerdings relativ einfach und zwischenzeitlich auch deutlich besser dokumentiert. Die Dokumentation wurde diesbezüglich nachgebessert und um ein Poster erweitert, auf dem man auf einer einzigen Seite die wichtigsten Einzelschritte der Installation findet. Zusätzlich gibt es jetzt auf der ODA-Seite im OTN noch sogenannte „Step-by-Step“-Videos, die ausführlich auf die Installation eingehen. Was bleibt, ist im Moment noch ein kleiner Schönheitsfehler, der aber nur dann auftritt, wenn man nicht mit einer US-Tastatur arbeitet. Zurzeit wird das Mapping einer deutschen Tastatur noch nicht erkannt beziehungsweise es wird nicht gefragt, welches Tastaturlayout man verwenden möchte. Dieser kleine Schönheitsfehler ist jedoch bereits gemeldet worden und sollte leicht zu korrigieren sein.

Weiterführende Informationen

ODA-Seite im OTN mit Offline Configurator, Third-Party-Agents-Liste, Step-by-Step-Videos: <http://www.oracle.com/technetwork/server-storage/engineered-systems/database-appliance/index.html>
 Dokumentation inklusive Setup-Poster: http://docs.oracle.com/cd/E22693_01/index.htm
 End User Bundle (Grid- und Datenbank-Software): <https://updates.oracle.com/download/12978712.html>
 My Oracle Support Note 1373617.1 via <https://support.oracle.com/>
 Informationen auf Deutsch: http://www.oracle.com/webfolder/technetwork/de/community/dbadmin/tips/dbappliance_fa/index.html
 Datenblatt: <http://www.oracle.com/us/dm/oda-datasheet-german-1376266-de.pdf>
 Whitepaper: <http://www.oracle.com/us/dm/111123-oda-whitepaper-german-1377922.pdf>
 Häufig gestellte Fragen: <http://www.oracle.com/us/dm/111123-oda-faq-german-1377916.pdf>

Hinrich Tobaben

ORACLE Deutschland B.V. & Co. KG
 hinrich.tobaben@oracle.com



Abbildung 4: Appliance Manager – System-Info

Viel zu oft beginnt die Optimierung einer Applikation zu spät – nämlich erst dann, wenn deren Entwicklung bereits abgeschlossen ist. Soweit kann es nur kommen, wenn der Performance eine geringere Bedeutung als anderen Applikations-Anforderungen beigemessen wird. Performance ist jedoch nicht eine Option; sie ist ein Schlüsselkriterium jeder Applikation. Ungenügende Performance wirkt sich nicht nur nachteilig auf die Akzeptanz der Applikation aus, sie führt normalerweise – aufgrund der geringeren Produktivität – auch zu einem niedrigeren Return of Investment. Zusätzlich verursacht schlechte Performance auch höhere Software-, Hardware- und Betriebskosten.

Designing for Performance

Christian Antognini, Trivadis AG

Im Folgenden werden die häufigsten datenbankrelevanten Design-Probleme beschrieben, die in vielen Fällen zu einer schlechten Performance führen. Die aufgeführten Praxistipps basieren auf der jahrelangen Tuning-Erfahrung des Autors. Sie sollen helfen, rechtzeitig notwendige Überlegungen in das Datenbank-Design einfließen zu lassen und so spätere, aufwändige Tuningmaßnahmen zugunsten einer optimalen Performance zu vermeiden.

Unzulänglichkeiten im logischen Datenbank-Design

Früher war es selbstverständlich, einen Datenarchitekt in das Entwicklungsprojekt einzubeziehen. Dieser war oft nicht nur für die Daten und das Datenbank-Design verantwortlich, sondern auch Teil des Teams für die Gesamtarchitektur und das Design der Applikation. Er hatte meist große Erfahrung mit Datenbanken und wusste genau, wie man Datenintegrität und Performance mittels geeigneten Designs sicherstellen kann. Heutzutage ist dies leider immer weniger der Fall. Viel zu oft gibt es in Projekten kein formales Datenbank-Design mehr. Die Applikationsentwickler erstellen das Client- oder das Middle-Tier-Design und plötzlich wird das Datenbank-Design von einem sogenannten „Persistent-Framework-Tool“ (im Java-Umfeld beispielsweise Hibernate oder iBatis) generiert. Die Datenbank wird lediglich als einfacher Daten-Container betrachtet.

Praxistipp: Die Bedeutung des logischen Datenbank-Designs darf nicht unterschätzt werden. Wird dieses nicht

korrekt entwickelt, können Performance-Probleme auftreten.

Implementation von generischen Tabellen

Jeder CIO träumt von Applikationen, die sich einfach und schnell an neue oder veränderte Anforderungen anpassen lassen. Der Schlüsselbegriff heißt „Flexibilität“. Solche Träume manifestieren sich oftmals in Form von Applikationen mit generischem Datenbank-Design. Funktionen wie beispielsweise das Hinzufügen neuer Attribute, sind dann nur noch eine Frage der Konfiguration. Die beiden folgenden Datenbank-Design-Modelle erlauben diese Flexibilität:

- **Entity-Attribute-Value-Modell (EAV)**
Wie der Name sagt, werden für jede Information mindestens drei Spalten benötigt: Entität, Attribut und Wert. Jede Kombination definiert den Wert eines spezifischen Attributs, das mit einer spezifischen Entität verknüpft ist.
- **XML**
Jede Tabelle hat nur wenige Spalten, wobei zwei Spalten immer vorhanden sind: ein Identifier und eine XML-Spalte, in der fast alles gespeichert ist. Es können auch zusätzliche Spalten für die Speicherung von Metadaten (zum Beispiel wer wann die letzte Modifikation vorgenommen hat) vorhanden sein.

Aus Sicht der Performance sind diese Designs äußerst problematisch. Der Grund liegt in der engen Beziehung von Flexi-

bilität zur Performance. Ist die Flexibilität maximal ausgeprägt, dann ist die Performance minimal (und umgekehrt).

Praxistipp: Ein flexibles Design bedeutet implizit auch suboptimale Performance. In gewissen Situationen mag suboptimal gut genug sein, aber in anderen Situationen kann dies gravierende Folgen haben. Daher sollten generische Tabellen nur zum Einsatz kommen, wenn damit auch die erforderliche Performance erreicht werden kann.

Verzicht auf Constraints

Constraints (Primärschlüssel, Unique-Keys, Fremdschlüssel, NOT NULL-Constraints und Check-Constraints) sind nicht nur fundamental, um die Datenintegrität zu gewährleisten, sie werden auch vom Optimizer für die Generierung der Ausführungspläne verwendet. Ohne Datenbank-Constraints ist der Optimizer nicht in der Lage, bestimmte Optimierungstechniken auszunutzen. Sind Constraints auf Applikationsebene definiert, führt dies zu zusätzlichem Code, der sowohl geschrieben, als auch getestet werden muss. Probleme können dabei auch im Bereich der Datenintegrität entstehen, weil die Daten auf Datenbank-Ebene jederzeit manuell geändert werden können.

Praxistipp: Es empfiehlt sich, alle Constraints auf Datenbank-Ebene zu definieren.

Unzulängliches physisches Datenbank-Design

Projekte, in denen das logische Design direkt auf das physische Design

abgebildet wird, ohne die Vorteile der Oracle-Datenbank-Eigenschaften zu nutzen, sind nicht ungewöhnlich. Das häufigste und offensichtlichste Beispiel ist die direkte Verknüpfung aller Beziehungen zu einer Heap-Tabelle. Aus Sicht der Performance ist dies nicht optimal. In gewissen Situationen gewährleisten Index-Organized-Tabellen (IOT), Index-Cluster oder Hash-Cluster eine bessere Performance.

Eine Oracle-Datenbank bietet wesentlich mehr Indexierungsmöglichkeiten als normale BTree- und Bitmap-Indizes. Je nach Situation sind komprimierte Indizes, Revers-Indizes, Function-Based-Indizes (FBI), linguistische Indizes oder Text-Indizes zur Verbesserung der Performance geeignet.

Für große Tabellen ist zudem der Einsatz von Partitionierung interessant. Den meisten Datenbank-Administratoren ist dies auch bewusst. Ein häufiges Problem ist jedoch, dass die Entwickler denken, die Partitionierung der Tabellen hätte keinen Einfluss auf das logische Datenbank-Design. Dies trifft nur teilweise zu. Aus diesem Grunde empfehlen wir, Partitionierung gleich von Anfang an einzuplanen.

Praxistipp: Für eine bestmögliche Performance sollte man das logische Design nicht direkt auf das physische Design abbilden.

Falsche Datentyp-Auswahl

Auf den ersten Blick scheint die Wahl des Datentyps keine schwierige Aufgabe zu sein. Trotzdem findet man immer mehr Systeme mit unzulänglichen Datentypen. Es gibt vier Hauptprobleme im Zusammenhang mit der Datentyp-Auswahl:

- *Falsche oder unzureichende Datenvalidierung*
Daten müssen innerhalb der Datenbank validiert werden können. Das heißt, es sollten beispielsweise keine numerischen Werte in einem „VARCHAR2“-Datentyp gespeichert sein, weil die Daten sonst extern zu validieren sind.
- *Informationsverlust*
Beim Konvertieren eines Originaldatentyps in einen Datenbank-

Datentyp können Informationen verloren gehen. Werden beispielsweise Datum und Zeit in einem „DATE“-Datentyp anstelle eines „TIMESTAMP WITH TIME ZONE“-Datentyps gespeichert, gehen die Sekundenbruchteile und die Zeitzonen-Information verloren.

- *Gewisse Dinge funktionieren nicht wie erwartet*
Operationen und Eigenschaften, für die die Reihenfolge der Daten wichtig sind, können zu unerwarteten Resultaten führen. Zwei typische Beispiele sind Range-partitionierte Tabellen und „ORDER BY“-Klauseln.
- *Optimizer-Anomalien*
Der Optimizer kann unter Umständen aufgrund einer falschen Datentyp-Auswahl keine gute Schätzung hervorbringen, was zu nicht optimalen Zugriffspfaden führt. Der Optimizer kann in einem solchen Fall seine Aufgabe nicht erfüllen, weil ihm die korrekte Information fehlt.

Praxistipp: Es gibt viele gute Gründe, die Datentypen korrekt und sorgfältig auszuwählen; man erspart sich damit viele Probleme.

Inkorrekte Verwendung von Bind-Variablen

Aus Sicht der Performance haben Bind-Variablen Vor- und Nachteile. Zu den Vorteilen zählt das Cursor-Sharing im Library-Cache und die damit verbundene Verhinderung von Parsing. Der Nachteil von Bind-Variablen in der „WHERE“-Klausel (und ausschließlich dort) ist, dass entscheidende Informationen dem Optimizer verborgen bleiben. Der Optimizer kann beispielsweise Literale besser verwenden als Bind-Variable. Dies ist speziell dann der Fall, wenn dieser prüfen muss, ob ein bestimmter Wert innerhalb oder außerhalb eines Wertebereichs liegt (größer oder kleiner als der größte oder der kleinste Attributwert), wenn ein Prädikat in der „WHERE“-Bedingung auf einem Wertebereich basiert (beispielsweise „HIREDATE > ‚2009-12-31‘“) oder wenn der Optimizer für die Optimierung Histogramme verwendet. Konse-

quenterweise sollte man Bind-Variable nur dann verwenden, wenn keine der drei erwähnten Bedingungen zutrifft. Ausgenommen davon sind SQL-Anweisungen, deren Parse-Zeit um Größenordnungen kleiner ist, als deren Ausführungszeit. In diesem Fall ist der Einsatz von Bind-Variablen nicht nur irrelevant für die Ausführungszeit, sondern es erhöht sich auch das Risiko von sehr ineffizienten Ausführungsplänen.

Praxistipp: Bind-Variable sollten auf keinen Fall zum Einsatz kommen, wenn der Optimizer maßgeblich von Histogrammen Gebrauch macht. Bind-Variable verringern das Sicherheitsrisiko durch SQL-Injection, weil die Syntax einer SQL-Anweisung über einen Bind-Variablenwert nicht geändert werden kann.

Fehlender Einsatz von Advanced-Datenbank-Features

Oracle verfügt mit seiner High-End-Datenbank-Engine über zahlreiche Advanced-Features. Darunter versteht man Datenbank-Funktionen und -Eigenschaften, die üblicherweise in anderen Datenbank-Produkten nicht verfügbar sind. Der Autor empfiehlt, die datenbankseitigen Möglichkeiten soweit wie möglich zu nutzen und bereits existierende Features nicht manuell nachzubilden. Dies bedeutet natürlich auch, die mit der jeweiligen Version neu eingeführten Features speziell zu beachten und sie nicht nur hinsichtlich ihrer Funktionalität, sondern auch bezüglich der Stabilität zu testen.

Ein verbreitetes Argument gegen den Einsatz der Advanced-Features sind die eingeschränkten Portierungsmöglichkeiten auf Datenbanken anderer Hersteller – und damit eine gewisse Abhängigkeit von Oracle. Auf der anderen Seite werden Unternehmen die Datenbank unter einer bestimmten Applikation nur in Ausnahmefällen austauschen, sondern gleich die ganze Applikation ersetzen.

Praxistipp: Die Entwicklung von datenbankunabhängigem Applikationscode empfiehlt sich nur dann, wenn dafür gute Gründe vorliegen. Auch wenn dies der Fall ist, ist das Thema „Flexibilität vs. Performance“ zu beachten.

Fehlende Verwendung von Stored-Procedures

Ein Spezialfall des oben erwähnten Punkts ist die Verwendung von PL/SQL-Stored-Procedures für die Verarbeitung großer Datenmengen. Das bekannteste Beispiel ist ein Extract-Transform-Load-Prozess (ETL). Erfolgt die Extrakt- und Ladephase in derselben Datenbank, ist es aus Performance-Sicht beinahe fahrlässig, in der Transformationsphase auf die Vorteile der SQL- und PL/SQL-Engine zu verzichten, welche die Quell- und Ziel-Daten verwaltet. Leider führt die Architektur diverser bekannter ETL-Werkzeuge genau zu diesem Verhalten. Das heißt, die Daten werden aus der Datenbank extrahiert (und oft auf einen anderen Server verschoben), transformiert und wieder in die Datenbank geladen.

Praxistipp: Für die optimale Performance empfehlen wir eine möglichst datennahe Verarbeitung mithilfe des Einsatzes von PL/SQL-Prozeduren.

Vorschau auf die nächste Ausgabe

Die Ausgabe 02/2012 hat das Schwerpunktthema:

„Apex“

Folgende Themen sind im Fokus:

- Grundlagen
- Ausblick
- Plug-ins
- Apex mobile
- Projekterfahrungen

Die Ausgabe 02/2012 erscheint am 2. April 2012.

Die weiteren Schwerpunktthemen und Termine stehen unter www.doag.org/go/doagnews

Ausführung von unnötigen Commits

Commit-Operationen führen zu Serialisierung. Der Grund dafür ist einfach: Es gibt einen einzigen Prozess, den Logwriter-Prozess, der Daten in die Redo-Log-Files schreibt. Weil eine serialisierte Operation nicht skaliert, sollte man sie aus Performance-Gründen so weit wie möglich verhindern oder zumindest minimieren. Teilweise ist dies durch das Zusammenfassen einzelner Transaktionen möglich – also beispielsweise das Laden von Datensätzen durch Batch-Jobs. Anstatt nach jedem Datensatz ein Commit auszuführen, ist es vorteilhafter, die Daten in Batches einzufügen und nach jedem Batch einen Commit-Befehl auszuführen.

Praxistipp: Transaktionen sollten nicht zu häufig einen Commit-Befehl enthalten. Das Zusammenfassen von Transaktionen ist daher sinnvoll und meistens auch effizienter.

Häufiges Öffnen und Schließen von Datenbank-Verbindungen

Das Erstellen einer Datenbank-Verbindung und Starten des damit verbundenen dedizierten Prozesses auf dem Datenbank-Server ist bezüglich der erforderlichen Ressourcen und der notwendigen Zeit nicht zu unterschätzen. Schlimmster Fall ist eine Web-Applikation, die für jede Anfrage eine Datenbank-Verbindung öffnet und gleich wieder abbaut. Dies ist natürlich alles andere als optimal. In einer solchen Situation bietet sich der Einsatz eines Connection-Pools an.

Praxistipp: Häufiger Auf- und Abbau von Datenbank-Verbindungen ist zu vermeiden.

Wichtig: Last-Tests

Der Sinn und Zweck von Integrations- und Acceptance-Tests ist die Verifizierung der funktionalen Anforderungen, der Performance sowie der Applikationsstabilität. Es kann nicht oft genug betont werden, dass Performance-Tests dieselbe Wichtigkeit haben wie funktionale Tests. Liefert die Applikation ungenügende Antwortzeiten, ist das ebenso schlecht,

als wenn die funktionalen Anforderungen nicht erfüllt werden. In beiden Fällen kann die Applikation unbrauchbar sein.

Auch wenn die Entwicklung, die Implementierung und die Durchführung von aussagekräftigen Integrations- und Acceptance-Tests keine einfache Aufgabe ist, darf man darauf nicht verzichten. Ohne Last-Tests hat man keine Gewähr, dass die Applikation der erwarteten Belastung standhält.

Praxistipp: Um die erwartete Performance zu verifizieren, sollte eine Applikation nie ohne Last-Tests entwickelt werden. Das Mantra sollte heißen: Was nicht getestet ist, wird nicht funktionieren.

Fazit

Leider wird häufig für das Design einer Applikation und Datenbank der Performance-Aspekt unterschätzt oder erst gar nicht berücksichtigt. Diese Praxis führt häufig zu unerwarteten Überraschungen. Der Autor hat in diesem kurzen Artikel die datenbankrelevanten Design-Probleme beschrieben, die in der Praxis am häufigsten auftreten. Leider sind diese Probleme nicht ohne einen größeren Aufwand nachträglich zu lösen. Es empfiehlt sich daher, die dargestellten Problempunkte rechtzeitig zu berücksichtigen.

Weitere Informationen

„Der Oracle DBA – Handbuch für die Administration der Oracle Database 11gR2“, Hanser Verlag (ISBN 978-3-446-42081-6).

Christian Antognini
Trivadis AG

christian.antognini@trivadis.com



Nahezu jede Datenbank zieht irgendwann einmal um oder muss überarbeitet werden – sei es durch einen Storage-, Plattform-, oder Server-Wechsel, vielleicht sogar in eine virtuelle Maschine hinein, oder gleich bei einem Umzug in ein anderes Rechenzentrum. Für geschäftskritische Datenbanken fallen manchmal Spezial-Anforderungen an wie die Umstellung auf Unicode-Zeichensatz, Auszeiten im Minuten-Bereich oder Fallback-Möglichkeiten. Dieser Artikel stellt praxiserprobte Migrationsmethoden mit ihren jeweiligen Stärken, Schwächen und Randbedingungen vor.

Migrationsmethoden – von ASM bis Zeichensatz

Patrick Schwanke, CarajanDB GmbH

Export/Import geht immer

Die Werkzeuge „exp/imp“ und „Data Pump“ decken nahezu beliebige Migrations-Szenarien ab, insbesondere alle Versionssprünge zwischen Oracle 6 und 11.2. Data Pump erlaubt Sprünge zwischen 10.1 und 11.2.

Möchte man hingegen von einer bestehenden 11g-Datenbank migrieren, ist „exp“ zwar noch verfügbar, aber nicht mehr unterstützt. Spätestens für 11.2 empfiehlt sich ohnehin Data Pump, da in der Enterprise Edition Objekte standardmäßig mit Deferred Segment Creation erstellt werden und dadurch Tabellen, die noch nie Daten enthielten, vom alten „exp“ ohne Fehlermeldung einfach ignoriert werden (siehe MOS Doc-ID 960216.1).

Das benutzte Werkzeug muss der Version der jeweiligen Datenbank entsprechen. Bei einer Migration von 9.2 nach 11.2 muss man also mit einem 9.2-Export-Werkzeug starten und die Daten mit einem 11.2-Import-Werkzeug importieren. Der Hintergrund ist, dass zwar das von exp/imp beziehungsweise Data Pump benutzte Dateiformat abwärtskompatibel ist, nicht aber die Werkzeuge selbst. Häufig ist es sinnvoll, einfach das jeweilige Werkzeug aus dem entsprechenden Oracle-Home des Datenbankservers aufzurufen.

Nachteilig ist in jedem Fall der relativ lange Zeitraum für Export, Übertragung und Import, der eine entsprechend lange Datenbank-Auszeit bedeutet. Beide Werkzeuge sind insbesondere für LOB-Spalten sehr langsam.

Der Import läuft typischerweise etwa zwei bis drei Mal so lange wie der Export. Bei Datenbanken im TB-Bereich landet man so schnell bei Auszeiten bis zu mehreren Tagen.

Export, Übertragung der Daten und Import können parallelisiert werden, wobei gleichzeitig das Disk-I/O zum Schreiben und Lesen der Dump-Dateien entfällt. Bei exp/imp geht dies auf Unix-/Linux-Systemen, indem jeweils eine Named Pipe als Dump-Datei angegeben wird und ein weiterer Prozess per SSH die Daten aus der einen Named Pipe in die andere schreibt. Bei Data Pump gibt es hierfür einen eigenen Parameter namens „NETWORK_LINK“. Data Pump erlaubt für die Enterprise Edition auch die Parallelisierung des Exports beziehungsweise Imports selbst mithilfe der PARALLEL-Klausel. Achtung: Die beiden Klauseln „NETWORK_LINK“ und „PARALLEL“ lassen sich erst ab dem Patch-Level 10.2.0.4 kombinieren (siehe MOS Doc-ID 5972794.8).

Rolling-Migration geht meistens

Nahezu beliebige Migrationen decken Werkzeuge ab, die eine rein logische Datenbank-Replikation leisten, also die Transaktionen aus der alten Datenbank rekonstruieren, diese als SQL-Befehle an die neue Datenbank schicken und somit die Datenbestände synchronisieren. Die gesamte Physik der Datenbank (Version, Plattform, Storage, Tablespace-Layout, Tabellenpartitionierung etc.) kann völlig unterschied-

lich sein, daher auch die Flexibilität dieser Methode.

Der Autor hat praktische Erfahrungen mit zweien dieser Werkzeuge: SharePlex on Oracle von Quest Software und Oracle GoldenGate. Beide sind von der Funktionsweise her sehr ähnlich, wobei SharePlex als das ältere und auch reifere der beiden Produkte bezeichnet werden kann. GoldenGate hat dagegen den Vorteil, dass es auch außerhalb der Oracle-Welt arbeiten, also zum Beispiel von Oracle zu SQL-Servern replizieren kann.

Die zusätzlich anfallenden Lizenzkosten sowie die höhere Komplexität der Migration muss man gegen zwei Vorteile abwägen, nämlich Auszeit- und Risiko-Minimierung: Zum einen liegt die Datenbank-Auszeit unabhängig von der Größe oder Transaktionslast der Datenbank typischerweise im Minuten-Bereich. Zum anderen erhält man die Möglichkeit, auch Tage nach der Migration bei auftretenden Problemen ohne Datenverlust wieder auf die alte Datenbank zurückschwenken zu können.

Abbildung 1 zeigt den groben Ablauf. Die neue Datenbank legt man zunächst als leere Datenbank an, etwa mit dem Database Configuration Assistant (DBCA). Anschließend startet man die Replikation von der bestehenden in die neue Datenbank und puffert diese sofort, da die Zieldatenbank ja noch leer ist (Schritt 1).

Nun überträgt man einen möglichst aktuellen und vor allem konsistenten Datenbestand, indem man die Werk-

zeuge „exp“ beziehungsweise „Data Pump Export“ mit dem Parameter „FLASHBACK_SCN“ aufruft. Nach erfolgreichem Import in die neue Datenbank müssen bei SharePlex bestimmte Datenbank-Objekte wie Trigger oder Datenbank-Jobs dort zunächst deaktiviert werden, damit replizierte Änderungen nicht doppelt ankommen. Die Replikation synchronisiert sich anschließend mit der benutzten „FLASHBACK_SCN“ und fährt alle zwischenzeitlich gepufferten Transaktionen nach (Schritt 2). Bis hierhin ist noch keinerlei Auszeit aufgetreten.

Während der nun folgenden kurzen Datenbank-Auszeit aktiviert man Trigger und Datenbank-Jobs auf der neuen Datenbank, deaktiviert sie auf der alten und dreht die Replikationsrichtung, um sich eine Fallback-Möglichkeit ohne Datenverlust zu sichern für den Fall, dass erst nach einigen Stunden oder Tagen Probleme mit der neuen Datenbank auftreten (Schritt 3). Sodann dürfen sich die Benutzer und Anwendungen auf der neuen Datenbank anmelden.

Sowohl GoldenGate als auch SharePlex haben Restriktionen hinsichtlich der unterstützten Oracle-Versionen, Datentypen und bestimmter Funktionen innerhalb der Datenbank. Für GoldenGate ist ein entsprechendes Check-Skript unter [1] verfügbar. Bei SharePlex fragt man am besten beim Hersteller Quest Software nach.

Reine Server-Migrationen

Für den Umzug auf einen anderen Server genügt es prinzipiell, auf dem neuen Server ein Oracle-Home zu installieren, die Anwendung zu stoppen, mittels RMAN ein Backup der Datenbank zu erstellen und dieses auf dem neuen Server einzuspielen. Wem diese Auszeit zu lang ist, der kann auch auf dem neuen Server eine Physical-Standby-Datenbank aufbauen, die ja einem sehr aktuellen Backup entspricht.

Reine Storage-Migrationen

Läuft die Datenbank bereits auf Basis von Oracle Automatic Storage Management (ASM) und möchte man

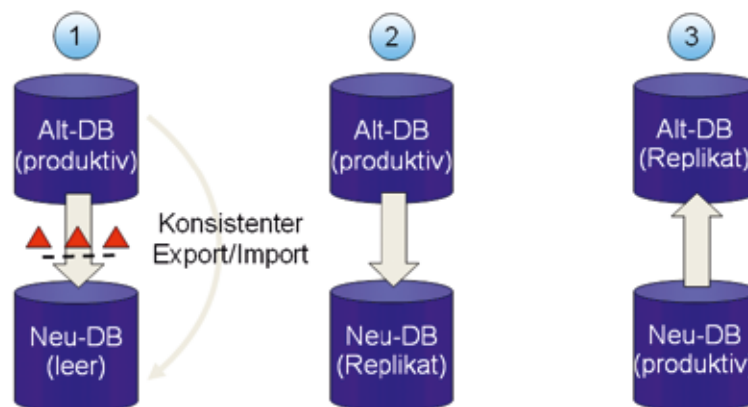


Abbildung 1: Ablauf einer Rolling-Migration

lediglich auf einen anderen Storage-Hersteller, Storage-Turm und/oder andere Platten migrieren, lassen sich sehr geschickt die Rebalancing-Fähigkeiten von ASM ausnutzen. Bei dieser Methodik entsteht keine Auszeit, sondern bloß zwischenzeitlich eine höhere I/O-Last auf dem Datenbank-Storage.

Nach Bereitstellung und Stamping der neuen Platten auf dem Datenbank-Server fügt man für jede ASM-Diskgruppe in einem ersten Schritt alle neuen Platten der Diskgruppe hinzu und entfernt in einem zweiten Schritt alle alten Platten:

```
SQL> ALTER DISKGROUP data ADD
DISK 'ORCL:ASMDISKNEW';
SQL> ALTER DISKGROUP data DROP
DISK 'ASMDISKOLD';
```

Das automatische Rebalancing sorgt jeweils für eine entsprechende Neuverteilung der Daten. Mit der View „V\$ASM_OPERATION“ in der ASM-Instanz lässt sich überprüfen, ob beziehungsweise wann das Rebalancing voraussichtlich beendet ist. Anschließend kann man die alten Platten abhängen.

Versions- und Storage-Wechsel

Mit dem Database Upgrade Assistant (DBUA), der mit der Datenbank-Software installiert wird, lässt sich bequem ein Versions-Upgrade bestehender Datenbanken durchführen. Der DBUA prüft vor der Migration automatisch

alle erforderlichen Voraussetzungen. Seit der Zielversion 11.1 kann zusätzlich zum reinen Versions-Upgrade auch eine Storage-Migration durchgeführt werden, insbesondere nach ASM. Außerdem ist ein Wechsel von 32-bit auf 64-bit-Oracle-Software möglich.

Die konkrete Upgrade-Zeit und damit Auszeit hängt von den installierten Datenbank-Optionen und -Komponenten ab. Typische Zeiten liegen zwischen 30 und 90 Minuten. Je weniger Optionen installiert sind (Text-, Spatial-Option, JVM, XML DB etc.), desto schneller das Upgrade. Es handelt sich um ein In-Place-Upgrade, daher ist die einzige Rückfallmöglichkeit das in [2] beschriebene, manuelle Downgrade, für das eine ähnliche Auszeit wie das eigentliche Upgrade veranschlagt werden kann.

Vor dem Beginn des Upgrade sollte man sich vergewissern, dass der geplante Versionsprung überhaupt möglich beziehungsweise unterstützt ist, ob gegebenenfalls vorher ein Patch-Release eingespielt werden muss oder sogar ein Upgrade in zwei Schritten durchgeführt werden muss. Abbildung 2 zeigt die möglichen Schritte beim Upgrade auf Version 11.2; im Zweifelsfall sollte man die offizielle Dokumentation [3] der neuen Datenbank-Version zu Rate ziehen, in der alle unterstützten Versionssprünge beschrieben sind.

Im Vorfeld wird die neue Datenbank-Software installiert und gegebenenfalls gepatcht. Nach dem Aufruf der neuen Softwareversion durch den

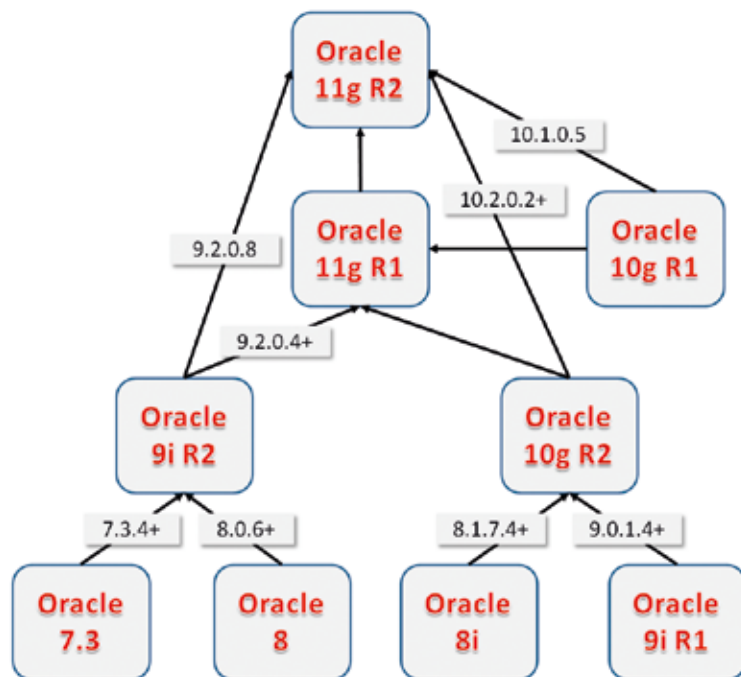


Abbildung 2: Mögliche Upgrade-Pfade mit dem DBUA

DBUA ist der Ablauf Wizard-gesteuert und in weiten Teilen selbsterklärend beziehungsweise in [3] beschrieben. Für umfangreiche Umgebungen bietet der DBUA auch ein Silent-Upgrade.

Nach erfolgreichem Upgrade ist der Server-Parameter „compatible“ der Datenbank noch auf den alten Versionsstand gesetzt. Die neue Datenbank nutzt also noch keine neuen Features. Der Pferdefuß dieser Methode ist, dass ein Rückfall auf die alte Version (Downgrade) auch nur möglich ist, solange dieser alte „compatible“-Wert gesetzt ist. Probleme nach dem Upgrade zeigen sich manchmal aber erst nach dem Heraufsetzen des „compatible“-Werts. Das Heraufsetzen ist jedoch eine Einbahnstraße und macht ein nachfolgendes Heruntersetzen oder Downgrade unmöglich. Ohne vorheriges Backup sollte man diesen Weg daher nicht beschreiten.

Plattform- und Storage-Wechsel

Ursprünglich aus dem Data-Warehouse-Bereich stammend bietet die sogenannte „Transportable Tablespace“-Technik (TTS) auch einen Migrationspfad. Seit der Version 10.1 ist hiermit das Kopieren von Tablespaces über die

Server-, Plattform- und Storage-Grenzen hinweg möglich, wobei das Backup-Werkzeug „RMAN“ die Konvertierung der Datendateien übernimmt.

Mit „Transportable Database“ (TDB) geht dies ab der Version 10.2 auch für die internen Tablespaces „SYSTEM“ und „SYSAUX“, allerdings nur für Plattform-Wechsel innerhalb derselben Endianness-Gruppe. So ist beispielsweise mit TDB ein Wechsel zwischen Linux und Windows (Little-Endian) oder zwischen AIX und HP-UX (Big-Endian) möglich, nicht aber zwischen AIX und Linux.

Jede Datenbank listet in der View „V\$TRANSPORTABLE_PLATFORM“ alle für TTS in Frage kommenden Ziel-Plattformen auf (dies sind nahezu alle Oracle-Plattformen); die View „V\$DB_TRANSPORTABLE_PLATFORM“ zeigt alle für TDB unterstützten Ziel-Plattformen, also alle Plattformen mit derselben Endianness. In jedem Falle muss zumindest die Quell-Datenbank als Enterprise Edition lizenziert sein.

Da es sich letztlich um das Erstellen einer physischen Kopie der bestehenden Datenbank nebst Konvertierung auf eine andere Plattform handelt, sind mit dieser Methode weder Versionssprünge noch Zeichensatz-Migra-

tionen oder Reorganisationen möglich. Für die Dauer der Übertragung ist die alte Datenbank „Read-only“. Die Auszeit ist also proportional zur Datenbank-Größe, aber deutlich geringer als bei exp/imp beziehungsweise Data Pump. Beide Methoden sind im Detail unter [4] und [5] beschrieben. Abbildung 3 skizziert den Ablauf. Mit einem kleinen Trick können in einem Aufwasch die Datendateien der alten Datenbank gelesen, konvertiert und weggeschrieben werden. Anstatt die alten Datendateien per SCP oder FTP auf den neuen Server zu übertragen, werden sie einfach per NFS exportiert und auf dem neuen Server gemountet – dort wird die Konvertierung angestoßen.

Reine Zeichensatz-Migrationen

Bei Zeichensatz-Änderungen dürfte es sich praktisch immer um Migrationen nach Unicode handeln. Eine solche Umstellung ist entweder innerhalb der bestehenden Datenbank oder durch Aufbau einer neuen Datenbank möglich. Letzteres ist insbesondere dann interessant, wenn weitere Änderungen wie Versions-Upgrade oder Plattform-Wechsel anstehen (also keine reine Zeichensatz-Migration) oder geringe Auszeiten beziehungsweise eine Fallback-Möglichkeit angestrebt werden. In diesem Falle muss man auf einen Export/Import oder eine Rolling-Migration zurückgreifen.

Beim Neuaufbau ist es wichtig, die Tabellen auf der Zieldatenbank zunächst ohne Daten zu erstellen, aber mit Längensemantik „Char“ anstelle der Standardeinstellung „Byte“, beispielsweise „VARCHAR2(50 CHAR)“ anstelle von „VARCHAR2(50 BYTE)“, analog für den Datentyp „Char“. Hierfür kann man entweder einen Export/Import der Tabellen ohne Inhalte fahren und anschließend mit MOS DocID 313175.1 die Längensemantik anpassen oder man lässt sich auf der alten Datenbank mittels „DBMS_METADATA“ oder einem Werkzeug wie „Toad“ ein CREATE-Skript für alle Tabellen ohne Semantik-Angabe erstellen und führt diese auf der neuen Datenbank wie folgt aus:

```
SQL> ALTER SESSION SET NLS_
LENGTH_SEMANTICS = ,CHAR';
SQL> -- erstelltes Skript, zum
Beispiel
SQL> CREATE TABLE (
    id NUMBER PRIMARY KEY,
    textcol VARCHAR2(50));
```

Anschließend werden per Import die Daten eingefügt, mit den Klauseln „IGNORE=Y“ beim konventionellen Import beziehungsweise „TABLE_EXISTS_ACTION=APPEND“ beim Datapump-Import. Die Alternative, also die In-Place-Umstellung einer Datenbank auf Unicode, geht dagegen mit dem sogenannten „CSALTER“-Werkzeug vonstatten, das in der Datenbank-Software enthalten ist. Wichtig zu wissen ist, dass dieses Werkzeug ausschließlich Data-Dictionary-Inhalte, also die Metadaten, in den neuen Zeichensatz konvertiert. Benutzerdaten, insbesondere Inhalte der Anwendungstabellen, werden nicht verändert.

Nur wenn sämtliche Benutzerdaten durch die Zeichensatzänderung unverändert bleiben, gibt das „CSALTER“-Werkzeug grünes Licht und ändert den Zeichensatz.

Alle nicht konformen Daten müssen zuvor exportiert, gelöscht und nach der Umstellung wieder importiert werden, wobei die Logik von exp/imp beziehungsweise Data Pump für die Konvertierung sorgt. Diese nicht konformen Daten zu finden ist Aufgabe des „CSSCAN“-Werkzeugs.

Bei einer Unicode-Konvertierung sind praktisch alle Zeilen betroffen, in denen Nicht-ASCII-Zeichen vorkommen. Je größer der Anteil dieser Daten, desto länger der Aufwand und damit die Auszeit mit dieser Methode. Im Extremfall müsste die ganze Datenbank exportiert und importiert werden. In diesen Fällen kommt alternativ nur eine Rolling-Migration in Frage. In jedem Falle sollte man vorbereitend oder als Teil der Auszeit die „VARCHAR2“- und „CHAR“-Spalten wie oben beschrieben auf „Längensemantik=‘CHAR‘“ konvertieren. Details zu dieser Methode stehen unter [6].

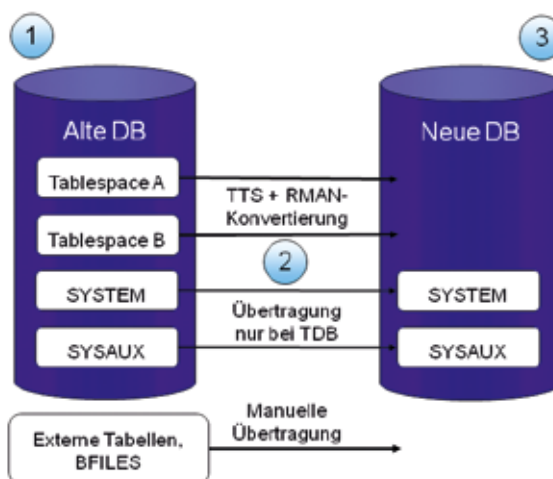


Abbildung 3: Migration mit Transportable Tablespace oder Transportable Database

P2V und V2P

Das Überführen einer bestehenden Datenbank in eine virtuelle Maschine oder auch eine entsprechende Migration zurück lässt sich auf einen der bereits besprochenen Fälle zurückführen. Für kritische Datenbanken empfiehlt der Autor, die Datenbank so aufzubauen, dass sie sowohl von einem physischen als auch von einem virtuellen Server aus gemountet werden kann. Dafür gibt es zwei Wege:

- Die VM greift direkt auf physischen Platten zu – beispielsweise mit VMwares Raw Device Mappings (RDM) – und nutzt diese mittels Oracle ASM für die Datenbank-Dateien.
- Die VM greift direkt auf NFS-Filer zu und nutzt diese mittels NFS-Mounts – ab Oracle 11g besser mit Oracle Direct NFS – für die Datenbank-Dateien.

Im einfachsten Falle handelt es sich um eine reine Migration von „physisch“ nach „virtuell“. Liegt die Datenbank wie oben beschrieben in ASM oder auf NFS, kann sie unmittelbar aus der VM heraus gemountet und geöffnet werden. Ist dies nicht der Fall, etwa bei Nutzung virtueller Platten in „.vmdk“- oder „.vhd“-Dateien, greifen die genannten Szenarien für reine Server-Migrationen. Auch wenn die Virtualisierung mit weiteren

Änderungen kombiniert wird, greifen die bereits beschriebenen Szenarien analog.

Referenzen

- [1] GoldenGate Checkskript: MOS Doc-ID 1298562.1
- [2] Downgrade nach DBUA-Upgrade: Oracle Database Upgrade Guide der jeweiligen Zielversion, Kapitel „Downgrading a Database“
- [3] Mögliche Upgrade-Pfade: Oracle Database Upgrade Guide der jeweiligen Zielversion, Kapitel „Preparing to Upgrade“
- [4] Plattformmigration mittels Transportable Tablespace: <http://www.oracle.com/technetwork/database/features/availability/maa-wp-11g-plattformmigration-129269.pdf>
- [5] Plattformmigration mittels Transportable Database: <http://www.oracle.com/technetwork/database/features/availability/maa-wp-10gr2-plattformmigrationtdb-131164.pdf>
- [6] In-Place-Zeichensatzmigration: Oracle Database Globalization Support Guide, Kapitel „Character Set Migration“

Patrick Schwanke
CarajandB GmbH
patrick.schwanke@carajandb.com



Kunden warten manchmal viele Jahre, bevor sie ein Migrationsthema anfassen. Das ist besonders in den Fällen verständlich, in denen von erprobter und stabiler Client/Server-Technologie in die neue komplexe Welt der Drei-Schichten-Architektur gewechselt werden soll. Wenn eine komplexe Software mit vielen Masken, Berichten und Schnittstellen zu migrieren und ein reibungsloser Wechsel in einer neue Welt gewünscht sind, müssen alle Details vorher genau untersucht werden. Dieser Artikel gibt einige Erfahrungen weiter und dem Entwickler ein paar hilfreiche Tipps an die Hand.

Erfahrungen bei der Migration von Forms/Reports 6i nach Forms/Reports 11.1.2 und 12

Frank Christian Hoffmann, Cologne Data GmbH

Ende Oktober 2011 ist auf den bekannten Download-Seiten von Oracle die hier getestete Version von Forms und Reports unter der Gruppe „Fusion Middleware“ aufgetaucht. Dort befinden sich auf weiteren Reitern Dokumentationen und Demos. Ein guter Einstieg für jeden, der sich in die Materie einlesen will, findet sich unter den folgenden Links:

- <http://www.oracle.com/technetwork/developer-tools/forms/overview/index.html>
- <http://www.oracle.com/technetwork/middleware/reports/overview/index.html>

Neuerungen im Release 11g R2

Die neue Version 11.1.2 verspricht neben zahlreichen Bugfixes eine Aufteilung von Entwickler- und Server-Installation, eine Client-Entwicklerunterstützung von Windows XP (32-Bit) bis hin zu Windows 7 (64-Bit-Systeme), Verbesserungen in der Monitoring-Software, Verbesserungen bei der Anzeige von „NCHAR“-Spalten sowie zahlreiche Fehler-Korrekturen. Eine sehr gute Optimierung stellt die Speichernutzung der Entwicklerumgebung dar. Die neue Umgebung 11.1.2 kommt mit etwa 800 MB Hauptspeicherbedarf für WebLogic sowie für Forms und Reports aus. Das ist zu den gut 3,5 GB der Vorgängerversion ein Schritt in die richtige Richtung.

Einen ersten Überblick für das Migrationsvorhaben liefert der knapp 60 Seiten umfassende Upgrade-Guide von Forms 6i nach Forms 11g (E24487-01). Hier kann man sich einen guten Überblick über die Anforderungen der Migration verschaffen. Wer es etwas genauer wissen will, kann dann die Deployment-Guides für Forms und Reports lesen und weitere knapp tausend Seiten studieren. Alle Dokumente lassen sich unter den beiden angegebenen Web-Links finden. Übersichtlich und praktisch erscheint der Download als PDF-File. Für ein erstes „Look&Feel“ sind auch die Demos und die Dokumentation dazu hilfreich.

Eine Migration kann aus Sicht des Autors durchgeführt werden, wenn alle kritischen Punkte des Software-Updates zuvor geklärt und Lösungen dazu gefunden wurden. Das kann dazu führen, dass Teile der Software neu geschrieben sowie Java-Funktionen ergänzt werden müssen und man sich von liebgewonnenen Benutzerdialogen verabschieden muss.

Schritt 1: Analyse der bisherigen 6i-Forms/Report-Software

Der Upgrade-Guide hilft, die Knackpunkte der Migration zu ermitteln. Überall dort, wo Schnittstellen zu anderen Software-Produkten wie Text- oder Tabellenverarbeitung, C-Programmen, Browser-Aufrufen und Erstellung oder Import von Dateien bestehen, wird eine

Anpassung notwendig sein. Eine Text-Suche mit einem geeigneten Tool nach Schlüsselbegriffen hilft hier weiter.

Wenn viele Änderungen notwendig sind, stellt sich die Frage nach der automatisierten Anpassung. Das kann eine externe Firma durchführen oder man macht sich selbst mit Hilfstools auf den Weg. Hinweise zu den Tools und Firmen, die eine Analyse unterstützen, hat Oracle unter „Customers und Partners“ auf der Forms-Seite zusammengestellt (siehe <http://www.oracle.com/technetwork/developer-tools/forms/learnmore/index.html>).

Soll die zukünftige Report/Forms-Welt unter Unix laufen, muss die komplette Software auf Groß- und Kleinschrift geprüft werden (Icons, Objekt- und Bibliotheks-Referenzen). Dafür empfiehlt der Autor den „Forms-API-Master“, der mit einem Skript diese komplette Arbeit in wenigen Minuten erledigt. Zudem muss man beachten, dass die Forms-Sessions dann in einer Unix-Umgebung laufen – alle „host“-Kommandos sind dann sicherlich auch zu prüfen und eventuell anzupassen.

Schritt 2: Auswahl von Software-Lösungen in der neuen Drei-Schicht-Umgebung

Es entstehen unterschiedliche Lösungsszenarien, abhängig davon, ob ein Unix- oder ein Windows-Server als Applikations-Server zum Einsatz kommt,

ob man seine Reports im Browser oder direkt als PDF-File mit einem PDF-Reader angezeigt bekommen will und welche Applet-Technologie zum Einsatz kommen soll. Es sind viele weitere Details zu beachten, wenn man eine nicht Oracle-zertifizierte, browserunabhängige Lösung mit „Java Web Start“ oder eine Barrierefreiheit über die Java-Bridge erreichen will. Hier empfiehlt der Autor den Lösungsansatz von Tom Kyte, für jede Anforderung ein kleines Beispiel zu entwickeln. Diese Beispiele sollten dann mit Anwendern und Entwicklern diskutiert und abgestimmt werden, damit alle in dem Migrationsprozess mitgenommen werden. Das hält dann etwas auf, dient aber auf jeden Fall der Akzeptanz des Vorhabens.

Schritt 3: Konfiguration von Server, Entwickler und Anwenderarbeitsplätzen

Das klingt nur nach etwas Software-Installation – ist jedoch in der Praxis sehr aufwändig. Wichtig ist, dass dieser Bereich nicht unterschätzt wird. Das Ganze fängt damit an, dass man die Zertifizierungsmatrix studiert und sich ein System überlegt, das auch von Oracle zertifiziert ist. Dann plant man eine zusätzliche Hardware-Beschaffung für den Applikations-Server und spart schon mal sein Geld für die dafür notwendigen Oracle-Lizenzen. Aber auch technische Fragen müssen zuvor beantwortet sein: „Wie viele Forms-Sessions müssen auf meinen Applikations-Server?“, „Deckt ein Server das alles ab?“ oder „Wie ist meine Backup- und Recovery-Strategie für meinen Applikations-Server?“ Hinzu kommen Software-technische Fragen: „Wo lege ich meine neuen Applikations-Parameter für Entwickler-, Test- und Produktions-Umgebung ab (Server-Namen, Report-Server-Namen, Verzeichnisse etc.)?“, „Wie führe ich meine Applikations-Software-Updates durch?“, „Wie kann ich prüfen, wer angemeldet ist, und notfalls Forms-Sessions stoppen, ohne den Applikations-Server anhalten zu müssen?“, „Wie verteile ich JAR-Files (etwa für Forms-eigene oder Windows-Komponenten)?“, „Wie richte ich einen Entwickler-Arbeitsplatz ein?“, „Wie soll ein Anwender-Arbeitsplatz

aussehen?“, „Wo generiere ich Dateien aus einer Forms-Session?“, „Wie lade ich Dateien auf den Client und in welche Verzeichnisse?“, „Welches Konzept nutze ich beim Report-Server?“, „Wie werden Reports auf den Client geladen, wo liegen sie und wie sicher ist das neue Konzept?“, „Wie laufen zukünftig Massen-Drucke?“, „Wie signiere ich meine JAR-Files, damit sie geladen werden können?“, „Wenn Entwickler unter Windows entwickeln und die Software unter Unix verteilt werden soll – wie bilde ich den passenden Workflow dazu ab?“ oder „Wie richte ich Bildschirmauflösung, Farbschema, Hintergrundfarben, Logos etc. ein?“ Es ist ein wirklich komplexes Feld, das viel Zeit in Anspruch nimmt.

Schritt 4: Durchführung der Migration

Wenn die ersten drei Schritte erfolgreich absolviert wurden, sollte der vierte Schritt nur noch eine Ausführung sein. Ein ordentliches Konzept wird dann auch eine vernünftige Aufwandsschätzung für die Migrationsarbeiten beinhalten und könnte auch als Feinkonzept und Styleguide für eine Ausschreibung erhalten. So wird auf jeden Fall am Ende ein Ergebnis erreicht, das den gewünschten Vorstellungen entspricht und abgenommen werden kann.

Für den Entwickler empfiehlt sich auf jeden Fall, einen Account für Oracle Support zu bekommen. Hier findet man aktuelle Java-Files („JAR“) sowie die Hinweise zur Signierung und zu möglichen Problemstellungen. Weiterhin empfiehlt sich das Oracle-Forms-Forum für Fragen und Austausch rund um das Thema. Zu empfehlen ist auch das deutsche Oracle-Forms-Forum unter <http://www.oracle.com/webfolder/technetwork/de/community/forms/index.htm>. Der Code für die nachfolgenden Tipps steht auf der Webseite www.cologne-data.de/oracle.html. Weitere Tipps von den Lesern nimmt der Autor gerne in diese Liste auf.

Tipp 1: Installation eines Entwicklungsrechners unter Windows 7, SP1 (64x)

Es sind folgende Schritte erforderlich:

1. Forms/Reports 6i (falls man die alte Version noch weiter testen will) plus den Reports-Patch aus Patch 3 einspielen (nnb60.dll in „bin“ kopieren) sowie WinHlp32.exe für Windows 7 (64-Bit-Version) von Microsoft in Deutsch (damit man die Hilfefunktion weiter nutzen kann)
2. Java SE SDK 6 in Version 29 (64 Bit) unter „c:\java\sdk64“ sowie während der Installation JRE (64 Bit) unter „c:\java\jre64“ installieren
3. WebLogic 10.3.5 unter dem installierten 64-Bit-Java-„bin“-Verzeichnis mit „java -jar wls1035_generic.jar“ einrichten
4. Development-Version Forms/Reports 11g R2 installieren
5. Den passenden Datenbank-Client (32 Bit) installieren
6. Installation der diversen 3rd-Party-SQL-Client-Tools
7. Installation diverser Browser zum Testen (IE9 32 Bit und 64 Bit, Firefox 8, Chrome)
8. Installation Java Realtime Environment (JRE) Version 29 (32 Bit) unter „c:\java\jre32“

Tipp 2: Timeout.jar

Dieses Bean von Frank Nimphius aus den Demos löst die Forms-6i-Timer-Events ab, die viel zu viel Netzwerk-Traffic verursachen. Das JAR-File und den Code gibt es bei Oracle Support.

Tipp 3: Anpassung von Icons

Oracle liefert einen kleinen Satz von GIF-Icons mit dem JAR-File „*frmall.jar*“ aus. Viele Applikationen haben jedoch eigene Icons für Toolbar und Masken. Diese müssen in GIF- oder JPG-Dateien konvertiert werden. GIF-Files, die in der MDI-Toolbar verwendet werden, müssen „links oben“ und GIF-Files, die in der Maske stehen, „mittig zentriert“ ausgerichtet sein. Am besten nimmt man ein gutes Grafikprogramm und führt diese Arbeit bei transparentem Hintergrund durch.

Werden Icons sowohl in der Toolbar als auch in der Maske verwendet, sind zwei Versionen zu erzeugen. Der Autor gibt daher den in seiner Applikation verwendeten, eigenen „MDI-

Icons“ das Präfix „mdi“. Das zweite Problem besteht darin, dass diese Grafiken nicht so einfach dargestellt werden. Entweder sind sie in einem dem WebLogic-Server bekannten Pfad abgelegt und in der Datei „Registry.dat“ (beispielsweise „webutil/icons“) zugewiesen (siehe Listing 1) oder besser in einem signierten JAR-File, damit das Java-Applet lokal darauf zugreifen kann (siehe Listing 2).

```
registry.dat default.icons.
iconpath=webutil/icons
default.icons.iconextension=gif
formsweb.cfg codebase=
```

Listing 1: Variante A (nur zum Testen)

```
jar -cvf meine_icons.jar *.gif
formsweb.cfg
codebase=imagebase
```

Listing 2: Variante B (Produktion)

Tipp 4: Eine Java-Applet-Einstellung für alle Browser

Jeder, der mit verschiedenen Browsern experimentiert, ärgert sich über die Standard-Einstellungen in der Konfigurationsdatei „formsweb.cfg“ nach der Installation der Software. Neuere Java-Versionen werden zum Teil ignoriert, einige Browser kennen das Java-Applet nicht und bei manchen Browsern startet die Anwendung noch nicht einmal. Listing 3 zeigt eine Einstellung, die ausgehend von der JRE-Version 1.6.0.29, für alle gängigen Browser funktioniert. Am besten installiert man zuerst alle Browser, die man testen will, und spielt unter „http://java.com/de/download/index.jsp“ dann die aktuelle Java-JRE-Version auf. Die Parameter „jpi_classid“ und „jpi_codebase“ sind nur für Microsoft relevant. Die Mimetype-Einstellungen sind für Firefox und Microsoft unterschiedlich und mit einem Semikolon getrennt, aber für alle Java-1.6x-Versionen möglich.

Tipp 5: Kleines Batch-File zur Kompilierung aller Sourcen

Alle Oracle-Forms-Module sind nur Upgrade-fähig, das heißt, wenn ein Source-

```
jpi_classid=clsid:CAFEEFAC-0016-0000-FFFF-ABCDEFEDCBA
jpi_codebase=http://java.sun.com/update/1.6.0/jinstall-6-windows-i586.
cab#Version=1,6,0,29
jpi_mimetype=application/x-java-applet;version=1.6
```

Listing 3: Windows-Einstellungen in „formsweb.cfg“

```
Echo compiling Reports..
for %%f in (*.rdf) do rwconverter userid=user/passwort@tns
stype=RDFFILE source=%%f dtype=REPFILE dest=%%f logfile=compilation_
log.log overwrite=YES batch=YES

Echo compiling Libraries
for %%f IN (*.pll) do frmcmp userid= user/passwort@tns module=%%f
batch=yes module_type=library compile_all=yes window_state=minimize

Echo compiling Menu
for %%f IN (*.mmb) do frmcmp userid= user/passwort@tns module=%%f
batch=yes module_type=menu compile_all=yes window_state=minimize

Echo compiling Forms....
for %%f IN (*.fmb) do frmcmp userid= user/passwort@tns module=%%f
batch=yes module_type=form compile_all=yes window_state=minimize
```

Listing 4: Das Batch-File

```
120 : 0 : „F9“ : 29 : „Werteliste“
81 : 2 : „Ctrl+Q“ : 32 : „Beenden“
112 : 2 : „Ctrl+F1“ : 35 : „Tastenbelegung“
121 : 0 : „F10“ : 36 : „Speichern“
114 : 1 : „Shift+F3“ : 61 : „Nächster Primärschlüssel“
115 : 1 : „Shift+F4“ : 62 : „Datensatz entfernen“
117 : 1 : „Shift+F6“ : 63 : „Datensatz löschen“
115 : 0 : „F4“ : 64 : „Datensatz duplizieren“
117 : 0 : „F6“ : 65 : „Datensatz einfügen“
```

Listing 5: Auszug aus der Datei „frmweb.res“

code einmal unter 11g R1 oder R2 kompiliert und gespeichert wurde, kann er nicht mehr für eine Vorgängerversion wiederverwendet werden. Daher sollte man nicht die Quelle migrieren, sondern immer nur eine Kopie davon, um die alten Sourcen zu erhalten. Listing 4 ist ein kleines Batch-File zur Konvertierung aller Sourcen. Es muss im Sourcen-Verzeichnis gestartet werden. Zuvor ist natürlich ein „tnsnames“-Eintrag für die Datenbank unter Forms 11g im Oracle „CONFIG“-Verzeichnis erfolgt, am besten für die Datenbank 11g in einer Zeile ohne Returns und Leerzeichen:

```
tns = (DESCRIPTION = (ADDRESS_
LIST = (ADDRESS=(PROTOCOL=TCP)
(HOST=129.247.1.1)(PORT=1526)))
(CONNECT_DATA=(SERVICE_NAME =
oraent)))
```

Tipp 6: Tastatur-Umsetzung in Web-Forms

Um das gleiche Tastaturverhalten wie in Forms 6i zu erreichen, muss ein Terminal-File zugewiesen werden, das die Tastatur-Eingaben richtig umsetzt. Die Zuordnung erfolgt in formsweb.cfg: „term=frmweb.res“. Listing 5 zeigt einen Auszug aus der Datei „frmweb.res“ im Forms-Verzeichnis.

Tipp 7: Das neue Report-Plug-in „Blob-Destination“ (seit 11g R1)

Der Report-Server kann einen Report seit Version 11g R1 auch direkt in eine Tabelle mit einer BLOB-Spalte (Binary Large Object) schreiben. Damit lässt sich der Report-Prozess besser steuern, man kann den Reports eigene IDs und Namen geben und sie danach browser-unabhängig direkt starten. Das Plug-in muss registriert sein, ein aktuelles 11g-JAR-File von Oracle Support heruntergeladen werden und die Aufrufe brauchen ein paar zusätzliche Parameter.

Tipp 8: Applet-Start browserunabhängig – Java Web-Start

Dieser Tipp ist nicht von Oracle zertifiziert und sollte deshalb zuvor durch eingehende Tests geprüft werden. Folgende Datei (siehe Listing 6) ist als „.jnlp“ zu speichern. Web-Start ist bereits in der Java-Realtime-Umgebung von 1.6 ab Version 12 integriert. Die Lösung ist denkbar, wenn Tipp 7 umgesetzt wird und Reports nicht mehr über „web.show_document“ in Browsern dargestellt werden. Der Autor hat das bisher nur mit einer Maske getestet und kann noch nicht viel dazu sagen.

Tipp 9: Forms-Builder 11.1.2 im „Look & Feel“ von 6i darstellen

Wenn man in der Registry „HKEY_LOCAL_MACHINE/SOFTWARE/Oracle“ von Forms 11.1.2 die nachfolgende Zeile einfügt, wird aus der blauen Forms-Builder-Darstellung wieder eine wie in Forms 6i: „FORMS_CLAF = TRUE“.

Tipp 10: Ein Java-Applet-Fenster aus dem Browser starten (Barrierefreiheit)

Wer das Applikationsfenster nicht in einem Browser, sondern als Extra-Applet darstellen möchte, wählt im „webforms.cfg“ die Anweisung „separateFrame=True“. Damit wird Forms auch barrierefrei, weil ein Screenreader über die Java-Bridge nun Titel, Spalten und Felder vorlesen kann. Mehr dazu auf den Oracle-Seiten unter „Accessibility“. Nachteil dabei bleibt, dass das Browserfenster auch beim

```
<jnlp spec="1.0+" codebase="http://host:7001/forms/java">
<information>
<title>Meine Applikation</title>
<vendor>Meine Firma</vendor>
<description>Forms Application - Java Web Start</description>
<homepage href="http://host:7001" />
</information>
<security>
<all-permissions />
</security>
<resources>
<j2se version="1.6+" />
<jar href="frmall.jar" main="true" />
</resources>
<applet-desc main-class="oracle.forms.engine.Main" name="Forms"
width="1350" height="1010">
<param name="separateFrame" value="false" />
<param name="lookAndFeel" value="oracle" />
<param name="splashScreen" value="false" />
<param name="colorScheme" value="swan" />
<param name="serverURL" value="http://host:7001/forms/
lservlet?ifcfs=http://VPCE001:7001/forms/frmservlet" />
<param name="serverApp" value="default" />
<param name="logo" value="false" />
<param name="envFile" value="default.env"/>
<param name="baseHTML" value="webutilbase.htm"/>
<param name="baseHTMLjpi" value="webutiljpi.htm"/>
<param name="WebUtilArchive" value="frmwebutil.jar,jacob.
jar,timeout.jar"/>
<param name="WebUtilLogging" value="off"/>
<param name="WebUtilLoggingDetail" value="normal"/>
<param name="WebUtilErrorMode" value="Alert"/>
<param name="WebUtilDispatchMonitorInterval" value="5"/>
<param name="WebUtilTrustInternal" value="true"/>
<param name="WebUtilMaxTransferSize" value="16384"/>
<param name="serverArgs" value="module=start.fmx userid=user/pwd@tns
debug=no buffer_records=no debug_messages=no array=no query_only=no
render=no host= port= record= tracegroup= log= usesdi=yes
param_jnlp=TRUE" />
</applet-desc>
</jnlp>
```

Listing 6: Die „.jnlp“-Datei

Schließen von Forms offen bleibt und die Report-Darstellung von Browser zu Browser sehr unterschiedlich erfolgt. Einzige Ausnahme bildet der Firefox-Browser. Dort kann man in der Version 8.0 das PDF-Plug-in „Native“ einstellen. Der Aufruf eines PDF-Files aus dem Browser startet direkt den PDF-Reader.

Tipp 11: Dateien auf dem Client mit „RUN32.DLL“ starten

Dokumente mit Endungen wie „.docx“ oder „.pdf“ können auch mit auf lokaler Betriebssystemebene verknüpften Programmen gestartet werden. Folgender Aufruf erlaubt die Anzeige eines ge-

nerierten Dokuments. Dieser setzt die Installation von „WEBUTIL“ voraus und erwartet das Dokument in dem Standardbenutzer-Verzeichnis des angemeldeten Anwenders:

- webutil_host.NonBlocking („rundll32 url.dll, FileProtocolHandler, ||
- webutil_clientinfo.get_system_property(user.home') || ,\' || p_filename);

Zuvor wurde das Dokument mit „webutil“ angelegt: „... client_text_io.fopen(webutil_clientinfo.get_system_property(user.home') || ,\' || p_filename,'W');“

Tipp 12: Anpassung von HOST-Befehlen

Windows7 speichert standardmäßig in der 64-Bit-Version 32-Bit-Software im Verzeichnis „Program Files (x86)“ und 64-Bit-Software unter „Program Files“. Die Sonderzeichen können beim Aufruf alter CMD-Befehle zu Fehlern führen. Dieser Tipp umgeht diese Einschränkung. Das Beispiel startet den IE9 aus beiden Umgebungen: „Start“ c:\progra~1\Internet Explorer\iexplore.exe“ www.google.de“ (64-Bit-Umgebung) und „Start“c:\progra~2\Internet Explorer\iexplore.exe“ www.google.de“ (32-Bit-Umgebung).

Tipp 13: Farb-Auswahl und „Look & Feel“

Die Farb-Auswahl „swan“ ist in einigen Dokumenten nicht aufgeführt und macht einen brauchbaren Eindruck. Listing 7 zeigt einen Vorschlag zu einem netten Layout.

```
webforms.cfg:
# colorScheme=teal, blaf, swan
colorScheme=swan
# lookandfeel=Generic, Oracle
lookandfeel=Oracle
```

Listing 7: Farbauswahl „swan“

Fazit

Mit der Version 11.2.0 hat Oracle die WebLogic-basierte Lösung von Forms/Reports gegenüber 11.1.1.x deutlich verbessert. Der Aufwand einer Migration kann nur durch eine sorgfältige Analyse der alten Software und eine möglichst präzise Konzeption der neuen Lösung bestimmt werden. Zu beachten sind hier besonders die bisherigen Schnittstellen und Office-Automatationen. Dabei sollte auch der Aufwand für die Konfiguration von Entwickler-, Anwender- und Server-Installation nicht vernachlässigt werden. Alle neuen

Technologien müssen exakt getestet, optimiert und mit dem Anwender abgestimmt werden, um einer so robusten und unverwüstlichen Client-Software wie Oracle Forms 6i ebenbürtig zu sein und beim Anwender nicht das Gefühl aufkommen zu lassen, die teure Migration wäre im Vergleich zur alten Lösung nur ein Rückschritt.

Frank Christian Hoffmann
Cologne Data GmbH
fch@cologne-data.de



Wir begrüßen unsere neuen Mitglieder

Firmenmitglieder

Fabian Klose	Steltix Deutschland GmbH
Claudia Rosenberger	usb GmbH
Klaus Blaschek	b.telligent GmbH & Co. KG
Harald Sellmann	its-people Frankfurt GmbH

Persönliche Mitglieder

Kai Liegl	Volker Sikora
Tobias Arnhold	Gabriele Herold
Hermann von Kobylinski	Reinhard Hellenkamp
Zekeriya Kalabalik	Jörg Kröger
Corinne Krüger	Peter Schubert
Stefanie Baudach	Matthias Korn
Enrico Lausch	Patrick Schwanke
Claudia Romanova	Joachim Hofmann

Das Upgrade kritischer Applikationen kann schwierig sein. Eines der Hauptprobleme hinsichtlich der Verfügbarkeit liegt darin, dass lange Ausfallzeiten nicht regelmäßig eingeplant werden können. Aus diesem Grund ist es für solche Applikationen wünschenswert, Online-Upgrades zu implementieren. Dies erfordert, dass sowohl die fragliche Applikation als auch jegliche Software, die von der Applikation genutzt wird (etwa die Datenbank), Online-Upgrades unterstützen.

Edition-Based Redefinition

Christian Antognini, Trivadis AG

Oracle hat dieses Problem schon seit Jahren erkannt. Unglücklicherweise sind bis einschließlich 11g R1 nur eine beschränkte Anzahl von Funktionalitäten hinsichtlich dieser Anforderungen implementiert. Erst mit 11g R2 hat sich diese Situation geändert. Mit der sogenannten „Edition-Based Redefinition“-Funktionalität bietet die Oracle-Datenbank nun endlich eine echte Unterstützung für Online-Upgrades an. Der Artikel liefert einen Überblick und einige Tipps zu diesem neuen Feature.

In einer Oracle-Datenbank wird ein Objekt durch seinen Namen, den zugehörigen Benutzer (Schema) und seine Edition identifiziert. Ziel des Ganzen ist, und das ist neu mit 11g R2, mehrere Kopien (Versionen) eines bestimmten Objekts zu unterstützen. Die Objekte, für die mehrere Kopien angelegt werden können, heißen „Editioned Objects“. Objekte, für die das nicht zutrifft beziehungsweise die keine Edition haben, werden als „Noneditioned Objects“ bezeichnet; die Spalte „EDITION“ in den Data Dictionary Views enthält dann den Wert „NULL“.

Das Ziel, zwei oder mehrere Versionen eines bestimmten Objekts zu unterstützen, ist einfach und klar: Während die Applikation eine Gruppe von Objekten benutzt, kann man eine andere Gruppe derselben Objekte nicht nur redefinieren, sondern auch für Testzwecke nutzen. Es ist also möglich, eine Gruppe von Objekten zu redefinieren und in einer isolierten Umgebung zu testen, bevor diese live gehen.

Nicht alle, nur die folgenden Objekt-Typen unterstützen dieses neue Konzept, das heißt, sie sind „Editionable Object Types“:

- FUNCTION
- LIBRARY

- PACKAGE und PACKAGE BODY
- PROCEDURE
- SYNONYM (PUBLIC SYNONYM ist ein „Noneditionable Object Type“)
- TRIGGER
- TYPE und TYPE BODY
- VIEW

Im Wesentlichen sind dies alle Objekt-Typen, die durch die erweiterte Funktionalität des Datenbank-Kernels über PL/SQL-Code oder externe Libraries genutzt werden. Die einzigen Ausnahmen sind die Views und die Synonyme. Man beachte, dass weder Java-bezogene noch datenspeichernde Objekt-Typen (wie Tabellen und Indizes) editionierbar sind.

Alle Objekte, die keine Editionen unterstützen, werden „Noneditionable Object Types“ genannt.

Editionserstellung für einen Benutzer ermöglichen

Der erste Schritt für die Nutzbarkeit des „Edition-Based Redefinition“-Features ist, den Eigentümer der zu redefinierenden Objekte freizuschalten. Nur Benutzer mit Berechtigung zur Editionserstellung können editionierte Objekte erstellen. Um dies zu ermöglichen, muss entweder die „ENABLE EDITIONS“-Klausel beim Erstellen eines Benutzers angegeben sein oder für einen bereits bestehenden Benutzer der „ALTER USER“-Befehl, wie im folgenden Beispiel dargestellt, ausgeführt werden.

```
SQL> ALTER USER cha ENABLE EDITIONS;
```

Es ist wichtig zu erwähnen, dass die Berechtigung zur Editionserstellung nicht wieder entzogen werden kann.

Mit anderen Worten, so etwas wie eine „DISABLE EDITIONS“-Klausel existiert nicht. Ob ein Benutzer die Möglichkeit hat, editionierbare Objekte zu erstellen, zeigt folgende Abfrage:

```
SQL> SELECT editions_enabled
2 FROM dba_users
3 WHERE username = ,CHA';

EDITIONS_ENABLED
-----
Y
```

Ist ein Benutzer dazu berechtigt, muss zur Identifizierung einer spezifischen Version von Datenbank-Objekten eine Edition angelegt werden.

Erstellen von Editionen

Jede Datenbank muss mindestens eine Edition haben. Dafür wird beim Anlegen einer neuen Datenbank eine Basis-Edition, die sogenannte „ORA\$BASE“ erstellt. Eine Datenbank unterstützt mehrere Editionen. Sie sind mithilfe einer Root-Edition über eine Hierarchie organisiert (Standardmäßig ORA\$BASE) und durch eine Eltern-Kind-Beziehung mit den anderen Editionen verbunden. Abbildung 1 zeigt eine Datenbank mit vier Editionen, der Basis-Edition und drei benutzerdefinierten Editionen. Jede der drei letzteren bezieht sich in unserem Beispiel auf ein Applikations-Release mit den Bezeichnungen REL1, REL2 und REL3.

Für das Erstellen einer Edition ist der „CREATE EDITION“-Befehl er-



Abbildung 1: Editionen werden über Hierarchien organisiert

forderlich. Um dieses SQL-Statement auszuführen, ist das „CREATE ANY EDITION“-Systemprivileg notwendig. Für das Anlegen der Editionen aus Abbildung 1 müssen folgende SQL-Befehle ausgeführt werden. Die Eltern-Kind-Beziehung ist über die „AS CHILD OF“-Klausel spezifiziert.

```
SQL> CREATE EDITION rel1 AS
CHILD OF ora$base;
SQL> CREATE EDITION rel2 AS
CHILD OF rel1;
SQL> CREATE EDITION rel3 AS
CHILD OF rel2;
```

Standardmäßig ist eine neue Edition nur über den anlegenden Benutzer oder durch den SYS-Benutzer verwendbar. Man kann die nötigen Privilegien an andere Anwender über das „USE ON EDITION“-Privileg vergeben. Wie der folgende SQL-Befehl zeigt, können die Privilegien entweder an einen bestimmten oder an alle Benutzer über einen „GRANT TO PUBLIC“ vergeben werden.

```
SQL> GRANT USE ON EDITION rel2
TO cha;
SQL> GRANT USE ON EDITION rel3
TO PUBLIC;
```

Das Anzeigen von Informationen hinsichtlich aller verfügbaren Editionen kann entweder über die View „ALL_EDITIONS“ oder über „DBA_EDITIONS“ selektiert werden. Das folgende Beispiel zeigt eine hierarchische Abfrage für einen solchen Zweck. Man beachte, dass Editionen nicht zu einem bestimmten Schema gehören und deshalb die beiden Views keine Spaltenspezifizierung eines Benutzers zu einer Edition zeigen.

```
SQL> SELECT *
2 FROM all_editions
3 CONNECT BY parent_edition_name = PRIOR edition_name
4 START WITH parent_edition_name IS NULL
5 ORDER BY level;
```

EDITION_NAME	PARENT_EDITION_NAME	USABLE
ORA\$BASE		YES
REL1	ORA\$BASE	YES
REL2	REL1	YES
REL3	REL2	YES

Da eine Datenbank mehrere Editionen unterstützt, muss ein Benutzer (oder wahrscheinlicher eine Applikation) in der Lage sein, die Edition zu referenzieren, welche die korrekte Version der editionierten Objekte enthält. Mit anderen Worten, die sogenannte „Session Edition“ muss ausgewählt sein.

Auswählen von Editionen

Die Auswahl einer Session Edition, von APIs (wie OCI und JDBC) oder Tools (wie SQL*Plus und Data Pump) ermöglichen die Spezifizierung während der Erstellung einer neuen Datenbank. Es ist zu beachten, dass, falls keine Edition angegeben wird, die Basis-Edition (definiert durch die Datenbankeinstellung des Parameters DEFAULT_EDITION) genutzt wird. Standardmäßig ist die Einstellung auf „ORA\$BASE“ gesetzt. Wie bei allen anderen Datenbank-Eigenschaften können diese über das „ALTER DATABASE“-Kommando geändert werden:

```
SQL> ALTER DATABASE DEFAULT
EDITION = rel1;

SQL> SELECT property_value
2 FROM database_properties
3 WHERE property_name = ,DE
FAULT_EDITION';

PROPERTY_VALUE
-----
REL1
```

Dabei sind besonders zwei Punkte einer solchen Konfiguration hervorzuheben:

1. Wenn man die Basis-Datenbank-Edition ändert, vergibt die Daten-

bank automatisch das „USE ON EDITION“-Privileg für die spezifizierte Edition an „PUBLIC“. Dies ist notwendig, weil die Basis-Datenbank-Edition für alle Benutzer verfügbar sein muss.

2. Das Konzept der Basis-Datenbank-Edition ist hauptsächlich aus Kompatibilitätsgründen für Applikationen gedacht, die während eines Aufbaus einer Datenbankverbindung nicht in der Lage sind, eine bestimmte Edition zu setzen. Deshalb sollte die Session Edition wenn möglich immer zum Verbindungszeitpunkt spezifiziert sein.

Ist die Datenbank-Verbindung erst einmal hergestellt, kann man die Session Edition selbstverständlich über ein „ALTER SESSION“-Kommando ändern. Beachten Sie, dass dieses SQL-Kommando nicht ausgeführt werden kann, wenn die aktuelle Session noch eine offene Transaktion beinhaltet. Das folgende Kommando setzt „REL1“ als Session Edition:

```
SQL> ALTER SESSION SET EDITION
= rel1;
```

Ebenfalls noch wichtig ist, dass das vorherige SQL-Kommando ein Top-Level-SQL-Kommando ist; daher kann es nicht in PL/SQL ausgeführt werden. Wenn man die Session Edition über PL/SQL ändern möchte (beispielsweise in einem Logon-Trigger), ist die „SET_EDITION_DEFERRED“-Prozedur aus dem „DBMS_SESSION“-Paket zu verwenden. Zwei Methoden sind zur Anzeige der verwendeten Session Edition verfügbar:

1. Nur für die aktuelle Session kann das „SESSION_EDITION_NAME“-Attribut aus dem „USERENV“-Kontext benutzt werden.
2. Für alle Sessions kann die View „V\$SESSION“ verwendet werden.

Die folgenden Select-Befehle zeigen die unterschiedlichen Abfragemöglichkeiten:

```
SQL> SELECT sys_context('USERENV','SESSION_EDITION_NAME') edition_
name
  2 FROM dual;

EDITION_NAME
-----
ORA$BASE

SQL> SELECT DISTINCT object_name AS edition_name
  2 FROM v$session, dba_objects
  3 WHERE session_edition_id = object_id;

EDITION_NAME
-----
ORA$BASE
```

Wenn beide – der Eigentümer der editionierten Objekte sowie die korrekte Edition – gesetzt sind, ist es an der Zeit, die editionierten Objekte selbst anzulegen.

Editionierte Objekte erstellen

Editionierte Objekte werden genauso wie alle anderen Objekte angelegt. Das einzige, was gemacht werden muss – neben der Tatsache, dass ein Benutzer verwendet wird, der auch editionierte Objekte besitzen darf –, ist die Edition auszuwählen, mit der das Objekt assoziiert werden soll. Beispielsweise ist das folgende SQL-Kommando für die Erstellung einer Prozedur und die Zuordnung zur Edition „REL1“ möglich:

```
SQL> ALTER SESSION SET EDITION
= rel1;

SQL> CREATE OR REPLACE PROCEDURE hello IS
  2 BEGIN
  3   dbms_output.put_
  4   line(,Hello from REL1');
  5 END;
```

Eine andere Methode für das Erstellen von editionierten Objekten besteht darin, die Editionen für einen Benutzer freizuschalten, der bereits Objekte besitzt. In diesem Fall sind alle Objekte automatisch mit der Root-Edition assoziiert, obwohl die Basis-Edition sinnvoller wäre.

Redefinieren editionierter Objekte

Für die Redefinition eines Objekts ist der erste Schritt, eine neue Edition zu erstellen und – falls dieser Schritt nicht

durch den Eigentümer des editionierten Objekts ausgeführt wird – die notwendigen Privilegien zu erteilen. Die einzelnen Schritte sind in den vorherigen Abschnitten bereits beschrieben. Wird eine neue Edition erstellt, so werden alle Objekte, die mit der Eltern-Edition assoziiert sind, von der Kind-Edition übernommen. Aber Vorsicht, die Objekte sind nicht mit der Kind-Edition assoziiert. Sie wurden für diese nur sichtbar gemacht. Beispielsweise ist, basierend auf der in Abbildung 1 dargestellten Hierarchie, eine in der REL1-Edition erstellte Prozedur ebenfalls in den Editionen REL2 und REL3 sichtbar (sofern sie nicht entfernt wurde).

Für die Redefinition eines editionierten Objekts muss die neue Edition ausgewählt und anschließend das editionierte Objekt entweder geändert, hinzugefügt, entfernt oder im Fall eines „Stored Programs“ durch reguläre DDL-Kommandos neu kompiliert werden. Beispielsweise wird durch die folgenden SQL-Kommandos die vorher in der REL1-Edition erstellte Prozedur in der REL2-Edition geändert und schließlich in der REL3-Edition gelöscht:

```
SQL> ALTER SESSION SET EDITION
= rel2;

SQL> CREATE OR REPLACE PROCEDURE hello IS
  2 BEGIN
  3   dbms_output.put_
  4   line(,Hello from REL2');
  5 END;
```

```
SQL> ALTER SESSION SET EDITION
= rel3;

SQL> DROP PROCEDURE hello;
```

Jetzt, da die Prozedur redefiniert ist, sind zwei unabhängige Objekte im Data Dictionary gespeichert. Infolgedessen kann man nun auswählen, welches von beiden Objekten man verwenden möchte. Für diese Zuordnung ist die entsprechende Edition, wie im folgenden Beispiel dargestellt, auszuwählen. Zu beachten ist, dass „USER_OBJECTS“ (das Gleiche gilt auch für „DBA_OBJECTS“ und „ALL_OBJECTS“) die Version anzeigt, die mit der Session Edition assoziiert ist.

```
SQL> ALTER SESSION SET EDITION = rel1;

SQL> SELECT edition_name
  2 FROM user_objects
  3 WHERE object_name = ,HELLO';

EDITION_NAME
-----
REL1

SQL> EXECUTE hello
Hello from REL1

SQL> ALTER SESSION SET EDITION = rel2;

SQL> SELECT edition_name
  2 FROM user_objects
  3 WHERE object_name = ,HELLO';

EDITION_NAME
-----
REL2

SQL> EXECUTE hello
Hello from REL2

SQL> ALTER SESSION SET EDITION = rel3;

SQL> SELECT edition_name
  2 FROM user_objects
  3 WHERE object_name = ,HELLO';

no rows selected
```

Wenn man alle Versionen anzeigen möchte, die im Data Dictionary gespeichert sind, sind entweder die Views „USER_OBJECTS_AE“, „DBA_OBJECTS_AE“ oder „ALL_OBJECTS_AE“ zu selektieren. Es ist im folgenden Beispiel zu beachten, dass das mit „REL3“ assozi-

ierte Objekt weiter verfügbar ist, obwohl es als nicht existent (non-existent) markiert ist.

```
SQL> SELECT edition_name, object_type
2 FROM user_objects_ae
3 WHERE object_name = 'HELLO';
```

EDITION_NAME	OBJECT_TYPE
REL1	PROCEDURE
REL2	PROCEDURE
REL3	NON-EXISTENT

Zusammenfassend sind für das Redefinieren editionierter Objekte folgende typische Schritte auszuführen:

- Erstellen der neuen Edition und Setzen einer Session Edition
- Modifizieren der editionierten Objekte und Sicherstellen der Gültigkeit aller Objekte
- Prüfen, ob die Applikation wie erwartet mit der neuen Edition arbeitet
- Dauerhaft auf die neue Edition wechseln

Die bisher beschriebenen Techniken werden benutzt, um Objekte eines editionierbaren Objekt-Typs zu redefinieren. Das Redefinieren von Tabellen – diese sind nicht editionierbar – erfordert die Einführung eines anderen Konzepts, dem sogenannten „View Editionieren“.

Views editionieren

Tabellen sind nicht editionierbar, daher kann man für die Struktur-Redefinition den Vorteil von Editionen nicht nutzen. Stattdessen kann jede Tabelle mit einer sogenannten „Editioning View“ versehen werden, und gleichzeitig können in der Applikation alle Tabellen-Referenzen durch Referenzen auf „Editioning Views“ ersetzt werden. Es ist wahrscheinlich nicht möglich, alle Referenzen zu ersetzen – zum Beispiel kann ein TRUNCATE-Befehl nicht auf eine View ausgeführt werden. So gesehen ist dies nicht unbedingt ein Problem, denn ein TRUNCATE-Befehl arbeitet unabhängig von einer Tabellenstruktur. Vereinfacht gesagt wird eine (trans-

parente) Ebene zwischen der Applikation und dem physikalischen Datenbankdesign erstellt. Eine „Editioning View“ ist eine normale View mit einigen speziellen Eigenschaften:

- Der Zweck einer „Editioning View“ dient nur dem Selektieren von Daten und um optional Aliasse für eine bestimmte Untermenge von Spalten innerhalb einer Tabelle bereitzustellen. Operationen wie Joins, Subqueries, Set Operatoren, Aggregations- und hierarchische Abfragen werden nicht unterstützt.
- „Editioning Views“ unterstützen DML-Trigger. Demzufolge sollten die Trigger, die normalerweise auf Tabellen-Level spezifiziert sind, auf View-Level präzisiert werden. Solche Trigger feuern nur, wenn man DML-Operationen auf „Editioning Views“ absetzt. Mit anderen Worten, sie werden nicht ausgeführt, wenn DML-Operationen auf die Tabelle selbst abzielen, auf der wiederum die „Editioning View“ basiert.
- Eine „Editioning View“ unterstützt keine „INSTEAD OF“-Trigger.

Für die Erstellung einer „Editioning View“ ist dem „CREATE VIEW“-Kommando das Schlüsselwort „EDITIONING“ anzuhängen. Das folgende SQL-Kommando zeigt beispielhaft, wie eine Tabelle durch eine „Editioning View“ die Zuordnung der physi-

kalischen Spaltennamen zu logischen Namen vornimmt. Zusätzlich ist ein DML-Trigger mit der View selber assoziiert. Der gleiche Trigger wäre bei einer normalen View nicht möglich, siehe Listing unten.

Während eines anfallenden Upgrades können die „Editioning Views“ gegebenenfalls in den „read-only“- oder „read-write“-Modus gesetzt werden. Für die Vereinfachung eines Upgrades sollten alle auf Tabellen basierenden „Editioning Views“ in den „read-only“-Modus gesetzt sein. Jedoch kann die Applikation die Daten aus diesen Tabellen nur lesen und nicht ändern. Wenn der „read-only“-Modus nicht in Frage kommt, muss überlegt werden, ob es eventuell notwendig ist, sogenannte „Crossedition Trigger“ einzusetzen.

Crossedition Trigger

Ein Crossedition Trigger kommt zum Einsatz, wenn zwei Anforderungen zu treffen:

1. Die Struktur muss während eines Upgrades redefiniert werden.
2. Die „Editioning View“, die über einer zu redefinierenden Tabelle liegt, kann nicht in den „read-only“-Modus gesetzt werden.

Nehmen wir an, dass eine Applikation den vollständigen Zugriff auf eine

```
SQL> DESCRIBE persons
Name                               Null?    Type
-----
ID                                  NOT NULL NUMBER(10)
FIRST_NAME                          NOT NULL VARCHAR2(100)
LAST_NAME                           NOT NULL VARCHAR2(100)
EMAIL                                NOT NULL VARCHAR2(100)

SQL> RENAME persons TO persons_tab;

SQL> CREATE EDITIONING VIEW persons AS
2 SELECT id, first_name AS firstname, last_name AS lastname, email
3 FROM persons_tab;

SQL> CREATE TRIGGER persons_bi_trg
2 BEFORE INSERT ON persons FOR EACH ROW
3 BEGIN
4     :new.id := persons_seq.nextval;
5 END;
6 /
```

Tabelle benötigt, die eine Spalte beinhaltet, die wiederum E-Mail-Adressen speichert (wie die eine aus dem Beispiel im vorherigen Abschnitt). Die Aufgabe besteht nun darin, diese Spalte in zwei aufzuteilen: Eine Spalte beinhaltet den Empfänger-Namen und die andere den Domain-Namen. Das Hinzufügen der beiden Spalten stellt in den meisten Fällen für diese Tabelle kein Problem dar. Einen „Table Lock“ auf eine stark frequentierte Tabelle zu bekommen ist vielleicht problematisch. Demzufolge muss eventuell ein Timeout mit dem Initialisierungsparameter „DDL_LOCK_TIMEOUT“ gesetzt sein. Für die Redefinition einer Tabelle ist vielleicht auch das „DBMS_REDEFINITION“-Package hilfreich. Zum Beispiel kann man dafür das folgende SQL-Kommando ausführen:

```
SQL> ALTER TABLE persons_tab
ADD (
  2   email_recipient VAR
      CHAR2(100),
  3   email_domain VAR
      CHAR2(100)
  4 );
```

Das Redefinieren dieser Tabelle ist ein guter Anfang. Aber was passiert mit den darin gespeicherten Daten? Grundsätzlich müssen für das korrekte Ausführen der Redefinition zwei Anforderungen erfüllt sein:

1. Die Daten, die während des Upgrades durch die Applikation geändert werden, müssen in der neuen Struktur gespeichert werden
2. Die Daten, die bereits in der zu redefinierenden Tabelle gespeichert sind, müssen ebenfalls in die neue Struktur konvertiert werden

Für die Erfüllung der ersten Anforderung, kann man in der neuen Edition einen „Forward Crossedition Trigger“ erstellen (niemals die alte Edition ändern). Das Ziel eines solchen Triggers besteht darin, eine Zeile von der alten Struktur in die neue zu überführen. In dem hier diskutierten Beispiel erfüllt das folgende SQL-Kommando diese Anforderung:

```
SQL> CREATE TRIGGER persons_fc_trg
  2   BEFORE INSERT OR UPDATE ON persons_tab FOR EACH ROW
  3   FORWARD CROSSEDITION
  4   DISABLE
  5   BEGIN
  6     :new.email_recipient :=
  7       regexp_substr(:new.email, '(.*)@', 1, 1, NULL, 1);
  8     :new.email_domain :=
  9       regexp_substr(:new.email, '@(.*)', 1, 1, NULL, 1);
 10  END;
 11  /
```

Es ist zu beachten, dass selbst, wenn der Trigger in der neuen Edition erstellt wurde, dieser erst feuert, wenn ein DML-Kommando durch eine Session aus einer Vorgänger-Edition, in der der Trigger erstellt wurde, ausgeführt wird. Aus diesem Grund muss durch die „FORWARD CROSSEDITION“-Klausel explizit definiert werden, dass es sich um einen Forward Crossedition Trigger handelt. Zu beachten ist auch, dass dieser Trigger im Status „Disabled“ zu erstellen ist. Das ist insofern wichtig, als ein ungültiger Trigger nicht die Verfügbarkeit der Applikation stört. Infolgedessen sollte man ihn nur aktivieren, wenn man sicher weiß, dass er gültig ist. Für die Erfüllung der zweiten Anforderung muss die Transformation auf die bereits gespeicherten Daten angewendet sein. Für dieses Szenario gibt es mehrere Möglichkeiten. Das Einfachste, aber nicht das Beste aus Performance-Sicht ist, den Trigger für jede der bereits in der redefinierten Tabelle gespeicherten Zeile auszulösen. Das „DBMS_SQL“-Package stellt, wie im folgenden PL/SQL-Block zu sehen ist, eine neue Version der „PARSE“-Funk-

```
SQL> DECLARE
  2   c INTEGER;
  3   r INTEGER;
  4   BEGIN
  5     c := dbms_sql.open_cursor;
  6     dbms_sql.parse(
  7       c => c,
  8       statement => ',UPDATE persons SET email_domain = email_domain',
  9       language_flag => dbms_sql.native,
 10     apply_crossedition_trigger => ',persons_fc_trg'
 11   );
 12   r := dbms_sql.execute(c);
 13   dbms_sql.close_cursor(c);
 14   COMMIT;
 15  END;
 16  /
```

tion bereit. Es ist zu beachten, wie ein Dummy-„UPDATE“-Kommando dafür benutzt wird, den durch den „APPLY_CROSSEDITION_TRIGGER“-Parameter spezifizierten Trigger auszulösen, siehe Listing unten.

Es ist essenziell, zur Vermeidung eines inkonsistenten Zustands zu verstehen, dass die erforderlichen Operationen, die diese beiden Anforderungen erfüllen sollen, in der folgenden Reihenfolge ausgeführt werden müssen:

- Aktivieren des Forward Edition Triggers:

```
SQL> ALTER TRIGGER persons_fc_
trg ENABLE;
```

- Warten auf jede Transaktion der redefinierten Tabelle – entweder muss diese bestätigt oder zurückgesetzt sein. Zu diesem Zweck wurde eine neue Funktion, die sogenannte „WAIT_ON_PENDING_DML“-Funktion, innerhalb des „DBMS_UTILITY“-Package hinzugefügt. Wenn dieser Schritt nicht ausgeführt wird, können die Transaktionen, die wäh-

rend des Aktivierens des Forward Edition Triggers geöffnet waren, in einen inkonsistenten Zustand gelangen:

```
SQL> DECLARE
  2   r BOOLEAN;
  3   scn INTEGER;
  4 BEGIN
  5   r := dbms_utility.wait_on_
  6     pending_dml(
  7       tables => ,CHA.PER
  8       SONS_TAB',
  9       timeout => NULL,
10       scn => scn
11     );
12 END;
13 /
```

- Start des PL/SQL-Blocks, der den Forward Edition Trigger für jede Zeile auslöst

Sobald diese Operationen abgeschlossen sind, sind vielleicht auch einige neue Constraints zu erstellen. Beispielsweise sind in diesem Fall die neuen Spalten als „NOT NULL“ gesetzt.

```
SQL> ALTER TABLE persons_tab MODIFY (
  2   email_recipient NOT NULL,
  3   email_domain NOT NULL
  4 );
```

Das bisher gezeigte Beispiel legt dar, was getan werden kann, um die Änderungen, die von einer mit der alten Edition arbeitenden Applikation auf die von der neuen Edition genutzten Datenstrukturen zu propagieren. Eine weitere Situation, die berücksichtigt werden muss: Was passiert, wenn aktuell beide, die alte und die neue Edition, gleichzeitig genutzt werden? Dies tritt auf, wenn ein sogenannter „Hot Rollover“ auf die neue Edition erforderlich ist. Für die Erfüllung dieser neuen Anforderung muss ein anderer

```
SQL> CREATE TRIGGER persons_rc_trg
  2   BEFORE INSERT OR UPDATE ON persons_tab FOR EACH ROW
  3   REVERSE_CROSSEDITION
  4   DISABLE
  5 BEGIN
  6   :new.email := :new.email_recipient || ',' || :new.email_domain;
  7 END;
  8 /
```

Typ von Trigger verwendet werden, der sogenannte „Reverse Crossedition Trigger“. Einfach gesagt ist der Zweck eines solchen Triggers, das Gegenteil eines Forward Crossedition Triggers zu erreichen. Mit anderen Worten hat dieser die Funktion, eine Zeile der neuen in die alte Struktur zu überführen. Das folgende SQL-Kommando zeigt ein Beispiel dafür, siehe Listing unten.

In diesem Fall ist der Trigger ebenfalls in der neuen Edition zu erstellen (auch hier gilt, dass man niemals die alte Edition ändern soll). Zu beachten ist außerdem, dass ein Reverse Crossedition Trigger nur ausgelöst wird, wenn eine Session ein DML-Kommando ausführt, in welcher der Trigger erstellt wurde (oder ein Nachfolger davon). Aus diesem Grund ist explizit über die „REVERSE_CROSSEDITION“-Klausel zu definieren, dass es sich um einen Reverse Crossedition Trigger handelt.

Die hier gezeigten Beispiele zu Crossedition Triggern sind sehr einfach. Wenn die Komplexität der Redefinition steigt, beispielsweise wenn SQL-Kommandos innerhalb des Triggers ausgeführt werden müssen, sind verschiedene andere Fragen zu berücksichtigen. Da diese kurze Einführung nicht alle diese Details abdecken kann, wird empfohlen, für zusätzliche Informationen die am Ende des Papiers gelisteten Dokumente zu referenzieren.

Entfernen von Editionen

Das „DROP EDITION“-Kommando entfernt eine Edition. Für die Ausführung dieses Kommandos ist das „DROP ANY EDITION“-Privileg erforderlich. Zusätzlich müssen folgende Bedingungen erfüllt sein:

- Die Edition darf nicht die letzte Datenbank-Edition sein

- Die Edition ist nicht die Basis-Datenbank-Edition
- Die Edition wird nicht als Session Edition verwendet
- Die Edition ist entweder Root oder Kind der Hierarchie
- Wenn die Edition Kind der Hierarchie ist und sie noch assoziierte Objekte besitzt, dann muss die CASCADE-Option spezifiziert werden
- Wenn die Edition Root der Hierarchie ist, darf die nachfolgende Edition keine Objekte der Root Edition übernehmen. Mit anderen Worten müssen alle übernommenen Objekte durch die nachfolgende Edition redefiniert werden

Das Entfernen einer nicht mehr genutzten Edition ist optional. Statt eine nicht mehr genutzte Edition zu entfernen, kann sie auch über das Widerrufen des „USE ON EDITION“-Privilegs für alle Benutzer außer Dienst gestellt werden.

Einschränkungen

11g R2 hat zwei wichtige Implementierungseinschränkungen, die sich auf die Hierarchien beziehen, die mit Editionen erstellt wurden:

- Jede Edition kann höchstens eine nachfolgende Edition haben. Demzufolge ist eine Hierarchie, wie auf der linken Seite in Abbildung 2 gezeigt, nicht erlaubt.
- Jede Datenbank besitzt nur eine Root Edition. Folglich kann eine Datenbank keine zwei Hierarchien, wie auf der rechten Seite in Abbildung 2 gezeigt, beinhalten.

Während die erstgenannte Einschränkung nicht ausschlaggebend ist, kann die letztere kritisch sein. In der Tat ist es nicht erlaubt, unabhängige Editionen für jede Applikation, die die gleiche Datenbank für Konsolidierungszwecke nutzt, zu verwenden. Für das Beispiel in Abbildung 2 könnte man eine Hierarchie für die CRM-Applikation nutzen und eine andere für die HR-Applikation.

Ein Feature, das aus Sicht des Autors fehlt, ist die Möglichkeit, die Basis-Edition

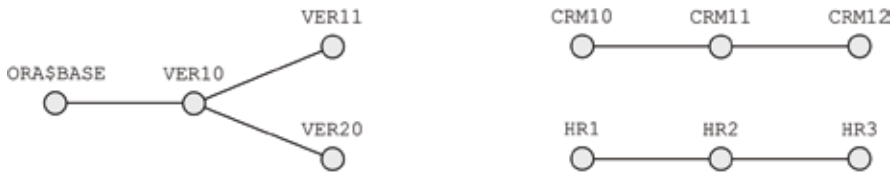


Abbildung 2: Beispiele für nicht zulässige Hierarchien

tion mit einem Datenbank-Service zu verknüpfen (und nicht nur mit der Datenbank, wie in der aktuellen Implementierung). Dies könnte, wenn mehrere Applikationen in einer einzigen Datenbank konsolidiert werden, sehr nützlich sein.

Die übrigen wichtigen Einschränkungen beziehen sich auf Indizes und Constraints. Da beide nur auf einer Tabelle erstellt werden können, gibt es keine Möglichkeit, den Vorteil von Editionen für diese zu nutzen. Im Falle von neuen Indizes kann ihre Auswirkung durch das Setzen auf „unsichtbar“ („Invisible“) limitiert werden. Mit

Constraints hingegen ist es eventuell nicht möglich, eine Definition zu verwenden, die korrekt mit mehr als einer Edition funktioniert. Die einzige generelle Ausnahme bezieht sich auf Check Constraints. Tatsächlich kann für diese ihre Logik auf der Session Edition basieren.

Fazit

Obwohl es immer noch einige Einschränkungen gibt, offeriert Edition-Based Redefinition komplett neue Features, die Online-Upgrades ermöglichen. Nichtsdestotrotz darf man die

Komplexität solcher Upgrades nicht unterschätzen.

Referenzen

1. Oracle White Paper, Edition-Based Redefinition in 11g R2
2. 11g R2-Dokumentation, Advanced Application Developer's Guide
3. 11g R2-Dokumentation, SQL Language Reference
4. 11g R2-Dokumentation, PL/SQL Packages and Types Reference

Christian Antognini
christian.antognini@trivadis.com
Trivadis AG



KeepTool mit neuer Version 10

Das handliche Werkzeug für Oracle™-Datenbanken



Zahlreiche neue Funktionen, z.B.

- Darstellung Ihrer Daten als Pivottabelle, ggf. mehrstufig.
- Praktische Hinweistexte bei der Datenerfassung.
- Überwachung und Steuerung der Optimizer-Statistiken.
- Data Pump-Schnittstelle.
- Jumplist für den Windows 7™ Taskbar.

Laden Sie die kostenlose Testversion unter www.keeptool.com herunter.



keeptool

Der Artikel stellt Test-Ergebnisse zur Skalierbarkeit und Performance des Oracle RAC vor. Getestet wurde auf der 10gR2-Chip-Multithreading-Architektur UltraSPARC T2, die auch unter ihrem Codenamen „Niagara 2 CPU“ bekannt ist. Die Tests erfolgten in einem realen Projekt für einen Kunden und brachten durchaus überraschende Resultate.

Vorsicht bei parallelen Abfragen eines Oracle RAC auf Multithreading-Chipsätzen

Piotr Sajda, OPITZ CONSULTING GmbH

Ziel der Tests war es, die Nutzbarkeit der Chip-Multithreading-(CMT)-Architektur innerhalb einer skalierbaren und effizienten RAC-Umgebung, die mehrere OLTP-Datenbanken innerhalb eines Host-RAC umfasst, zu untersuchen.

Die Tests erfolgten unter hoher Auslastung mit unterschiedlichen Typen kommerzieller und Open-Source-Anwendungen. In diesem Artikel stelle ich die Tests mit DVD Store vor, einer Open-Source-Anwendung für E-Commerce.

Testumgebung

Die Untersuchungen fanden in den folgenden Umgebungen statt (siehe Beispiel 1):

- **Middleware**
Zwei geclusterte Knoten mit dem WebLogic Server 10. Initial/Maximal Connection Pool von 200 Verbindungen/Datenbank (100 physikalische Verbindungen per Datenbank-Instanz)
- **Datenbank**
Oracle 10g R2 RAC Datenbank
- **Betriebssystem**
Solaris 10
- **Hardware**
2 x Sun T5220
- **Storage**
Storage der High-End-Klasse
- **Client**
DVD-Shop-J2EE-konforme Applikation, eingesetzt auf dem geclusterten WLS-Server

Mögliche Ursachen für die Überlastung

Eine erste kurze Analyse des AWR-Berichts brachte schnell die problematischen Bereiche ans Licht, die im Wesentlichen für die Überlastung des RAC Interconnect verantwortlich waren. Die weitere Analyse des „gc-buffer busy“-Wait-Events führte zu einer SQL-Abfrage, die im hohen Maße zur Leistungsminderung beitrug. Deutlich zeigt dies die Spitze kurz nach 5 Uhr bei den Average-Active-Sessions (siehe Beispiel 2).

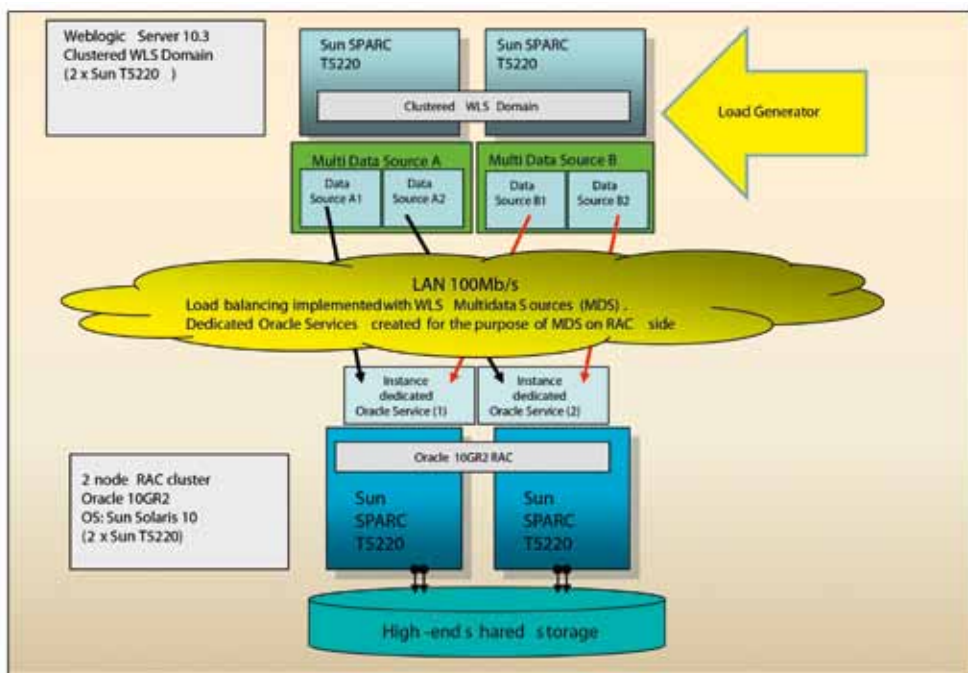
Die Latenz des RAC Interconnect lieferte keine klaren Hinweise auf Hardware-Probleme oder eine fehlerhafte Konfiguration (siehe Beispiel 3).

Dem Performance-Problem auf der Spur

Die vom „Average CR Block Receive Time“ geschätzte Latenz schwankte zwischen zwei und drei Millisekunden und die verfügbare Bandbreite von einem Gigabyte pro Sekunde war auch ausreichend. Und doch gab es anscheinend Probleme in diesem Bereich. Letztendlich war die direkte Ursache dieses Peaks die Abfrage „select count(*) from ORDERS“. Merkwürdigerweise wurde diese relativ kleine partitionierte Tabelle (70 MB) mit einem Full Table Scan (FTS) gelesen.

Bei der Analyse des Ausführungsplans der Abfrage und der Antwortzeit kam die entscheidende Abfrage:

```
SQL> select count(*) from ds2.orders ;
```



Beispiel 1: Die Testumgebung

Der Execution-Plan zeigte, dass der Optimizer eine Parallel Query benutzte, die natürlich einen FTS auf die Tabelle ORDERS anwendete und etwa 34 Sekunden dauerte. Im Falle solcher Abfragen wie „select count (*) from <Tabelle>“ besteht keine Notwendigkeit, auf einen einzelnen Block der Tabelle zuzugreifen, wenn ein geeigneter Index existiert. In diesem Fall könnte ein (Fast) Full Index Scan das Ergebnis liefern.

Schon bald fand der Autor einen Index, der als perfekter Kandidat ein schnelleres Ergebnis der Abfrage lieferte. Er verwendete nun einen Hint, um zu testen, ob bei der Verwendung dieses Index bessere (schnellere) Ergebnisse zu erwarten waren:

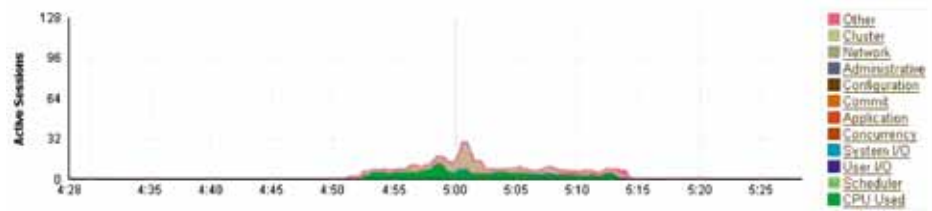
```
SQL> select /*+ index(orders
PK_ORDERS) */ count(*) from
ds2.orders;
```

Erwartungsgemäß lief die Abfrage viel schneller (siehe Beispiel 4), wenn der Optimizer einen INDEX FULL SCAN verwendete. Das Ergebnis stand jetzt nach 2,86 Sekunden fest.

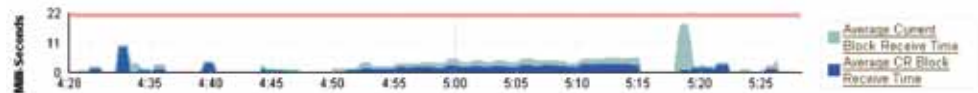
Warum aber entschied sich der Optimizer für einen so ungünstigen Ausführungsplan? Um die Entscheidungswege besser nachvollziehen zu können, wurde der Ausgang des Events 10053 analysiert. Auf diese Weise kam heraus, warum die Verwendung des bekannten Index nicht als bester Ausführungsplan gewählt wurde, warum die Kosten für einen FTS (COST 6) so gering und im Gegensatz dazu die Kosten für einen Full-Index-Scan (COST 3420) so hoch waren.

Der relevante Teil der Trace-Datei zeigte nach dem Tracing der Entscheidungspfade des Optimizers folgendes Ergebnis (siehe Beispiel 5).

In der Tat war der (full) TableScan (siehe Access Path: TableScan) für den Optimizer der effizienteste Plan. Die Zugriffskosten durch einen FTS wurden mit 6,39 berechnet. Im Gegensatz dazu wertete der Optimizer den bevorzugten Zugriff mit dem Full Index Scan mit enormen Kosten in Höhe von 809,89. Doch was bringt den Optimizer dazu, die Kosten des Ausführungsplans mit FTS mit 6,39 zu berechnen,



Beispiel 2: Überblick über die Anzahl der Zugriffe, Average Active Sessions (Current Up instances: 2/2)



Beispiel 3: Überblick über die Zugriffsgeschwindigkeit, Global Cache Block Access Latency (Current Up instances: 2/2)

```
Execution Plan
-----
```

Id	Operation	Name	Rows	Cost (%CPU)	Time
0	SELECT STATEMENT		1	3420 (3)	00:00:54
1	SORT AGGREGATE		1		
2	INDEX FULL SCAN	PK_ORDERS	1423K	3420 (3)	00:00:54

```
-----
Statistics
-----
4335 consistent gets
4480 physical reads
```

Beispiel 4: Ergebnisse des Hints

```
SINGLE TABLE ACCESS PATH
Table: ORDERS Alias: ORDERS
Card: Original: 1207210 Rounded: 1207210 Computed: 1207210.00 Non
Adjusted: 1207210.00
Access Path: TableScan
Cost: 1472.06 Resp: 6.39 Degree: 0
Cost_io: 1410.00 Cost_cpu: 181081500
Resp_io: 6.12 Resp_cpu: 785944
Access Path: index (index (FFS))
Index: PK_ORDERS
resc_io: 749.00 resc_cpu: 177670962
ix_sel: 0.0000e+00 ix_sel_with_filters: 1
Access Path: index (FFS)
Cost: 809.89 Resp: 809.89 Degree: 1
Cost_io: 749.00 Cost_cpu: 177670962
Resp_io: 749.00 Resp_cpu: 177670962
Access Path: index (FullScan)
Index: PK_ORDERS
resc_io: 3420.00 resc_cpu: 279905125
ix_sel: 1 ix_sel_with_filters: 1
Cost: 3515.92 Resp: 3515.92 Degree: 1
Best:: AccessPath: TableScan
Cost: 6.39 Degree: 256 Resp: 6.39 Card: 1207210.00 Bytes: 0
```

Beispiel 5: Ergebnis nach dem Tracing der Entscheidungspfade des Optimizers

was anschließend zu einem suboptimalen Ausführungsplan führt?

Resp steht für „Kosten für ‚full parallel execution‘“, die Antwort war wohl in der Architektur des Ausführungsmechanismus‘ der parallelen Query zu finden. Des Weiteren fiel auf, dass Resp_cpu, also die Kosten für die Nutzung der CPU-Ressourcen, wenn die parallele Option verwendet wird, relativ niedrig war: 785944. Damit war Resp_cpu rund 230-mal niedriger als die „serielle“ Option (Cost_cpu: von 181081500).

Auf den ersten Blick könnte man meinen, der Optimizer hätte den Ausführungsplan mit 230 parallelen Prozessen (Slaves) mitgerechnet. Die Analyse des nächsten Abschnitts ergab, dass die Maximalzahl der laufenden parallelen Slaves das Minimum der beiden folgenden Zahlen war:

```
Effective # of Parallel Max
Servers =
min ( parallel_max_servers ,
      (cpu_count * parallel_threads_
per_cpu) ) = 128
```

Dies bestätigte die obige Vermutung: 128 Slaves je Instanz ergaben in diesem Fall 256 Slaves für das ganze System, das auf einen Cluster mit zwei Knoten lief. Dies erklärte auch, warum die ermittelten „parallelen“ CPU-Kosten etwa 230 mal geringer waren als die der seriellen Option. Getestet

wurde auf einem Sun T5220-Rechner mit Multithreading-Chipsätzen mit 64 Threads in einer physikalischen CPU. Damit war die Herkunft dieser Zahlen einfach zu erklären: Der Optimizer-Algorithmus unterscheidet offensichtlich nicht zwischen den physikalischen CPU-Einheiten auf der Platine und den Verarbeitungsthreads innerhalb einer solchen physikalischen CPU. Die Hardware-Threads erscheinen dem OS als logische CPU-Einheiten. Diese wiederum verwendet der Optimizer zum Erstellen des Werts des cpu_count-Parameters, der beim Starten der Instanz ausgewertet wird. Die Arithmetik, die auf diesen Daten zur Anwendung kommt, führt letztendlich zu einer falschen Entscheidung beim Berechnen der Kosten der einzelnen Ausführungsszenarien.

Ob die parallele Abfrage wirklich 256 Slaves benutzt, ist leicht zu testen, indem man die Abfrage verwendet (siehe Beispiel 6), während im Hintergrund die Abfrage „select count(*) from orders“ ausgeführt wird.

Damit ist eindeutig belegt, dass die Abfrage 256 Slaves verwendete, die den Output an den Query Coordinator übergaben (SID=2503). Die Slaves verteilten sich auf zwei Instanzen, die jeweils 128 Slaves erzeugten. Die Zwischenergebnisse wurden an den Query Coordinator, in diesem Fall auf Instanz #2 geschickt.

An dieser Stelle lassen sich folgenden Schlüsse ziehen:

1. Offensichtlich lag kein RAC-System mit 128 CPUs auf jedem Knoten vor, sodass Contention in den inneren Knoten der CPU zu erwarten war.
2. Selbst wenn es diese Ressourcen gegeben hätte, wäre ein derart gestalteter Ausführungsplan nicht skalierbar, da er alle verfügbaren CPU-Ressourcen nur für diese eine Abfrage verwendet. Eine Abfrage, die zur gleichen Zeit ausgeführt würde, müsste also um die CPU konkurrieren.
3. Da die Hälfte der 256 Slaves, die auf Knoten #1 liefen, ihr Ergebnis an den Query Coordinator (auf Knoten #2) übergab, wurde der Interconnect stark beansprucht und die Reaktionszeit nochmals stark reduziert.

Das Letztere kann man in den „Average Active Sessions“ unmittelbar nach 5 Uhr beobachten. Interessant ist hierzu der Peak in „Cluster Waits“ (siehe Beispiel 2).

Tuning-Möglichkeiten

Es gibt ein paar Wege, die das Problem sofort lösen, doch alle sind weit davon entfernt, mustergültig zu sein. Man könnte beispielsweise Hints einbauen oder den DEGREE-Parameter der Tabelle ändern. Aber diese Ansätze hätten die Situation nur lokal verbessert und nicht global auf andere Objekte gewirkt. Deshalb sollte man zuerst einmal den Optimizer wissen lassen, dass ihm nicht die Anzahl an CPUs zur Verfügung stand, die er angenommen hatte. Ein weiterer Grund für die Reduzierung dieser Größe war, das System so skalierbarer zu machen.

Nach Setzen des Parameters „parallel_max_servers“ auf 8 und „parallel_threads_per_cpu“ auf 1 war tatsächlich eine signifikante Verbesserung der Reaktionszeit zu erkennen. Der folgende Code erzeugte 16 parallel ausgeführte Slaves, die auf zwei Instanzen laufen und Daten an den Query Coordinator zurückgeben:

INST_ID	SID	QCSID	SERVER#	DEGREE	REQ_DEGREE
1	3139	2503	1	256	256
1	3113	2503	2	256	256
(...)					
1	3211	2503	126	256	256
1	3224	2503	127	256	256
1	3209	2503	128	256	256
2	2399	2503	129	256	256
2	1744	2503	130	256	256
(...)					
2	2663	2503	254	256	256
2	396	2503	255	256	256
2	3169	2503	256	256	256
2	2503	2503			

257 rows selected.

Beispiel 6: Ergebnisse der Abfrage „SQL> select inst_id, sid, qcsid, server#, degree, req_degree from gv\$px_session where QCSID = 2503 order by INST_ID, SERVER# ;“

```
SQL> alter system set parallel_max_servers=8 scope=both sid='*' ;
SQL> alter system set parallel_threads_per_cpu=1 scope=both sid='*' ;
```

Nach Neustart beider Instanzen läuft die Abfrage erneut, nun mit den folgenden neuen Einstellungen (siehe Beispiel 7):

```
SQL> select count(*) from orders
COUNT(*)
-----
1423615
Dauer: 00:00:03.21
```

Als Erstes ist zu sehen, dass die Reaktionszeit drastisch gesunken ist – bis auf etwa drei Sekunden im Gegensatz zu den vorherigen 34 Sekunden, als 128 Slaves pro Instanz verwendet wurden. Auch die Zahl der nun ausgeführten Slaves wurde bestätigt: Sie war beim Ausführen der Abfrage mit den neuen Instanz-Einstellungen in der Tat 16 (8 pro Instanz) plus ein Query Coordinator.

Obwohl es nun viel bessere Ergebnisse gibt, erstellt der Optimizer den Ausführungsplan weiterhin mit Full Table Scan (FTS). Unter bestimmten Umständen hätte man damit zufrieden sein können, nämlich dann, wenn die Größe der zugrundeliegenden Tabelle winzig gewesen wäre. Die Segmente dieser Tabelle belegten zu dieser Zeit nur 8905 Blöcke (dies entspricht etwa 70 MB). Diese Tabelle konnte aber durchaus wachsen und viele Gigabyte belegen, womit die Reaktionszeit inakzeptabel würde.

Beim Untersuchen des Outputs von „autotrace“ in Hinblick auf „consistent gets“ (beziehungsweise „physical reads“, falls sich die Blöcke nicht im Buffer Cache befanden), kommt man zu dem Schluss, dass FTS im Vergleich zu Fast Index Scan eine weniger gut skalierbare Lösung darstellt. Um dies zu belegen, reicht ein Blick in die „Statistics“ der letzten Ausführung mit FTS, wo 8755 physikalische Lesevorgänge vorgenommen wurden. In der Tat korrelieren diese mit der Anzahl der von der Tabelle belegten Blöcke (8905). Sobald man den Optimizer zwingt, den Index zu benutzen, fällt die Zahl der physikalischen (und logischen) Lese-

und Schreibvorgänge ungefähr auf die Anzahl der Index-Blöcke, die der Index belegt (4401).

Die Reaktionszeit ist im zweiten Fall (bei serieller Ausführung von Full Index Scan) langsamer als bei Verwendung von FTS (Full Table Scan mit paralleler Abfrage).

Mit der parallelen Ausführung von Full Index Scan nutzen wir die Chance, dass der Optimizer einen Full Index Scan mit paralleler Abfrage als beste Ausführung wählt, ohne Hints zu benötigen. Und wirklich, nachdem parallele Operationen auf den Index erlaubt wurden, wählte der Optimizer den parallel ausgeführten Fast Full Index Scan als den besten Ausführungsplan. Nachfolgend die einzelnen Schritte dazu:

1. Zuerst den Buffer Cache leeren, um physikalisches I/O zu erzwingen. Dies wurde auf beiden Instanzen ausgeführt:
SQL> alter system flush buffer_cache; System altered.
2. Anschließend die Abfrage ohne Hint ausführen, also trifft der Optimizer die Entscheidung basierend auf seinem Wissensstand (siehe Ab-

Execution Plan

Id	Operation	Name	Rows	Cost (%CPU)	Time	TQ	IN-OUT	PQ Distrib
0	SELECT STATEMENT		1	9 (12)	00:00:01			
1	SORT AGGREGATE		1					
2	PX COORDINATOR							
3	PX SEND QC (RANDOM)	:TQ10000	1			Q1,00	P->S	QC (RAND)
4	SORT AGGREGATE		1			Q1,00	PCWP	
5	PX BLOCK ITERATOR		1423K	9 (12)	00:00:01	Q1,00	PCWC	
6	INDEX FAST FULL SCAN	PK_ORDERS	1423K	9 (12)	00:00:01	Q1,00	PCWP	

Statistics

```
48 recursive calls
0 db block gets
6817 consistent gets
4404 physical reads
0 redo size
518 bytes sent via SQL*Net to client
488 bytes received via SQL*Net from client
2 SQL*Net roundtrips to/from client
1 sorts (memory)
0 sorts (disk)
1 rows processed
```

Beispiel 7: Ergebnis der Abfrage

Execution Plan										
Id	Operation	Name	Rows	Cost (%CPU)	Time	Pstart	Pstop	TQ	IN-OUT	PQ Distrib
0	SELECT STATEMENT		1	18 (6)	00:00:01					
1	SORT AGGREGATE		1							
2	PX COORDINATOR									
3	PX SEND QC (RANDOM)	:TQ10000	1					Q1,00	P->S	QC (RAND)
4	SORT AGGREGATE		1					Q1,00	PCWP	
5	PX BLOCK ITERATOR		1423K	18 (6)	00:00:01	1	26	Q1,00	PCWC	
6	TABLE ACCESS FULL	ORDERS	1423K	18 (6)	00:00:01	1	26	Q1,00	PCWP	

Statistics	
6296	recursive calls
4	db block gets
11241	consistent gets
8755	physical reads
672	redo size
518	bytes sent via SQL*Net to client
488	bytes received via SQL*Net from client
2	SQL*Net roundtrips to/from client

Beispiel 8: Ergebnis der Abfrage ohne Hint

Unsere Inserenten	
Hunkler GmbH & Co. KG www.hunkler.de	Seite 3
KeepTool GmbH www.keeptool.com	Seite 49
Libelle AG www.libelle.com	Seite 25
MuniQsoft GmbH www.muniqsoft.de	Seite 13
DOAG 2012 Integrata-Kongress http://integrata-kongress.doag.org	Seite 55
DOAG 2012 Logistik & SCM www.doag.org/go/logistik2012	Seite 27
DOAG 2012 Applications applications.doag.org	U 2
ORACLE Deutschland B.V. & Co. KG www.oracle.com	U 3
Trivadis GmbH www.trivadis.com	U 4

bildung 8). Wie der Output in Abbildung 7 zeigt, wählt der Optimizer nun tatsächlich einen Index Fast Full Scan in paralleler Ausführung als den besten Ausführungsplan. Die Reaktionszeit verkürzt sich auf etwa zwei Sekunden gegenüber 3.21 Sekunden mit FTS und etwa 34 Sekunden bei FTS mit 256 Slaves (als Vergleich zum Ausgangspunkt).

3. Zum Schluss die Optimierung bei großer Auslastung (sechs aktive DVD-Store-Clients, die etwa 1020 Transaktionen pro Sekunde auf einen RAC-Cluster mit zwei Knoten generieren) überprüfen. Diese Abfrage wird bei stark ausgelasteter Datenbank ausgeführt.

Im Vergleich zur Reaktionszeit der Abfrage, die zuvor über zwei Minuten benötigte, haben wir mit vier Sekunden zwar erheblich bessere Ergebnisse, aber im Hinblick auf die Anforderungen an effiziente OLTP-Systeme konnten wir diese Zeitersparnis nicht als akzeptabel betrachten.

Fazit

Dieser Artikel zeigt nur einen geringen Teil der Testergebnisse. Die hier

angerissenen Skalierbarkeits- und Performance-Tests wurden weitergeführt und es haben sich noch weitere interessante Aspekte ergeben. Für den gegenseitigen Austausch und weitere Informationen steht der Autor gerne zur Verfügung.

Piotr Sajda
OPITZ CONSULTING GmbH
piotr.sajda@opitz-consulting.com



Kongress der Integrata-Stiftung 2012

integrata
stiftung



10. & 11. Mai 2012 | Berlin

Humane Nutzung der Informationstechnologie – Mehr Demokratie durch IT!

**HOCHKARÄTIGE
KEYNOTE SPEAKER**

- Prof. Julian Nida-Rümelin
- Wolf-Dieter Hasenclever
- Franz-Reinhard Habel

Politische Partizipation

Stärkung der Demokratie bei Sicherstellung
der weltanschaulichen Neutralität

Moderation

Michael Mörike
Vorstand Integrata-Stiftung

Qualifizierende Bildung

Befähigung zur Beteiligung an
gesellschaftlichen Veränderungen

Moderation

Prof. Dr. Karsten Wendland
Hochschule Aalen, Vorstand Integrata-Stiftung

Befriedigende Arbeit

Schaffung von Arbeits- und Erwerbsmöglichkeiten
für ein erfülltes Leben

Moderation

Welf Schröter, Forum Soziale Technikgestaltung

Gesunde Umwelt

Bewahrung und Fortführung einer natürlichen Umwelt

Moderation

Prof. Dr. Marco Mevius, HTWG Konstanz

*Verleihung des Wolfgang-Heilmann Preises für
humane Nutzung der Informationstechnologie 2012*

Weitere Infos <http://integrata-kongress.doag.org>

Bei der Entscheidung für eine ETL-Technik mit hohen Aktualitätsansprüchen (Near-Real-Time) sind Effizienz und Stabilität zwei wesentliche Kriterien. Dies gilt natürlich für die Transformation der Daten innerhalb des Data Warehouse, ganz besonders aber auch für die Extraktion der Daten aus den Quellsystemen und für die Befüllung – den Load der Staging Area. Der Artikel stellt eine Lösung vor, die auf Oracle Data Guard und Transportable Tablespaces basiert.

Effizientes und einfaches Staging in „Near Real Time“-Data-Warehouses

Mathias Zarick und Karol Hajdu, Trivadis AG

Die Märkte in zahlreichen Branchen werden immer unberechenbarer und kurzlebiger. Viele Fachanwender sind inzwischen nicht mehr bereit, Tage oder mehrere Stunden auf die neuesten Werte zu warten. Sie benötigen kürzere Latenzzeiten. Das „Near Realtime“-Data-Warehouse hat das Licht der Welt erblickt. Gleich, für welchen Ansatz man sich entscheidet, ein wesentlicher Aspekt bleibt immer die Verfügbarkeit einer Staging Area mit extrem kurzer Latenzzeit.

Unterschiedliche Lösungen mit verschiedenen Vorteilen

Es gibt zahlreiche technische Ansätze zur Datenextraktion und Befüllung von

Staging Areas (siehe Tabelle 1). Die Ansätze unterscheiden sich dabei in den folgenden grundlegenden Kriterien:

- Menge der transferierten Daten, die für eine Aktualisierung benötigt werden
- Vollständigkeit der transferierten Daten
- Einfluss auf die Performance der Quellsysteme
- Einfluss auf die Verfügbarkeit und Stabilität der Quellsysteme
- Kosten
 - Hardwarekosten und Software-Lizenzen sowie Implementierungsaufwände
 - Komplexität im Betrieb (Aufwände, Stabilität)

Falls das Quellsystem eine Oracle-Datenbank nutzt, gibt es eine effektive Lösung für die Extraktion der Daten und die Befüllung der Staging Area.

Der hier vorgestellte Ansatz basiert auf Oracle Data Guard, Flashback und Transportable Tablespaces. Die Lösung bietet größtenteils dieselben Vorteile wie andere Replikationstechniken, zum Beispiel Oracle Streams (Asynchronous Change Data Capture) oder Oracle GoldenGate:

- Nur relativ kleine Datenmengen müssen transferiert werden
- Es gibt nur einen geringen Einfluss auf die Performance und Verfügbarkeit des Quellsystems

Konzept / Implementierung	Transferiertes Datenvolumen	Vollständigkeit	Einfluss auf Performance des Quellsystems	Einfluss auf Verfügbarkeit / Stabilität des Quellsystems	Komplexität des Betriebs
Daten-Vollextraktion	☹️	😊	☹️	😊	😊
Marker-basierte Extraktion	😊	😊	😊	😊	😊
Journal-basierte Extraktion	😊	😊	☹️	😊	😊
Oracle Streams	😊	😊	😊/😊	😊	😊/☹️
Oracle Golden Gate	😊	😊	😊/😊	😊	😊/☹️

Tabelle 1: Überblick über die Aktualisierungsmöglichkeiten für Data-Warehouse-Staging-Areas

Konzept / Implementierung	Zu transferierende Datenmengen	Komplexität	Performance-einbußen im Quellsystem	Verfügbarkeit des Quellsystems	Betriebskomplexität
Neue Lösung mit Data Guard und Transportable Tablespaces	😊	😊	😊	😊	😊

Tabelle 2: Neue Lösung ermöglicht eine deutlich geringere Betriebskomplexität als Streams oder GoldenGate

- Es werden alle Arten von Änderungen bemerkt und transferiert

Die hier vorgestellte Lösung hat allerdings eine signifikant geringere Betriebskomplexität als Oracle Streams oder Oracle GoldenGate (siehe Tabelle 2).

Eine Lösung mit Data Guard

Guard verwaltet Standby-Datenbanken als Kopien von Primär-Datenbanken. Eine Standby-Datenbank kann dabei auch direkt für die Aktualisierung einer Staging Area in einem Data Warehouse genutzt werden. Die Idee basiert auf Data Guards Fähigkeit, eine physische Standby-Datenbank zeitweilig für Schreib- und Leseoperationen zu öffnen und sie anschließend wieder in den Ursprungszustand zurückzustellen, den sie beim Öffnen hatte. Dies wird durch die Flashback-Datenbank-Funktion und die sogenannten „garantierten Restore-Points“ ermöglicht. Abbildung 1 zeigt eine auf dem Data Warehouse Host (DWH) eingerichtete physische Standby-Datenbank für die OLTP-DB. Die primäre Datenbank liegt dabei auf dem Host-OLTP.

Aus der Staging Area lesen

Benötigt nun ein ETL-Programm im Data Warehouse die Daten aus der Staging Area, wird folgender Prozess angestoßen:

- Der Recovery-Prozess auf der Standby-Datenbank wird gestoppt
- Die physische Standby-Datenbank wird zu einer sogenannten „Snapshot“-Datenbank konvertiert und damit im Read-Write-Modus geöffnet
- Tablespaces der Standby-Datenbank (hier der Tablespace CRM) werden mit der „Transportable Tablespace“-Funktion in die DWH-Datenbank eingehängt
 - Der CRM-Tablespace in der Snapshot-Datenbank wird dabei auf „Read-only“ gesetzt
 - Die Metadaten (Tabellendefinitionen, Indexdefinitionen etc.) des Tablespace werden in die DWH-Datenbank geladen

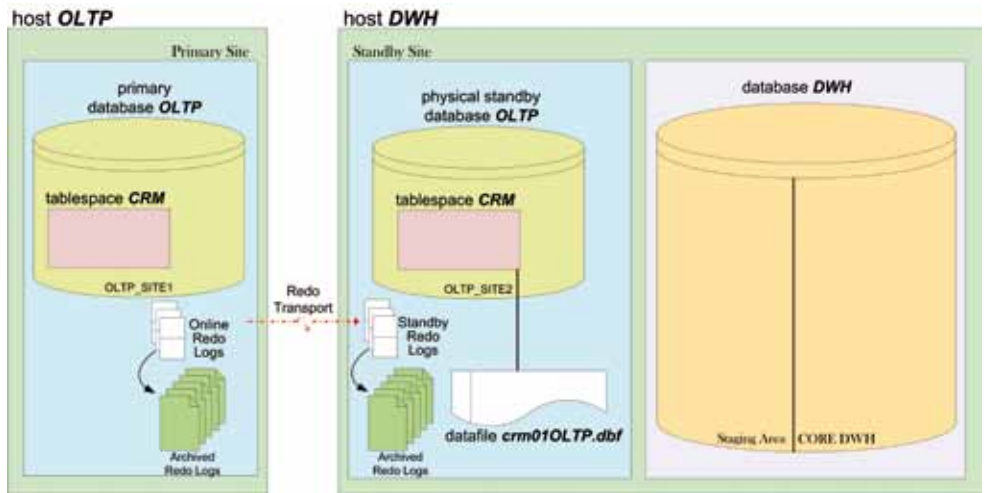


Abbildung 1: Auf dem DWH-Host wird eine physische Standby-Datenbank für OLTP mit Data Guard erstellt

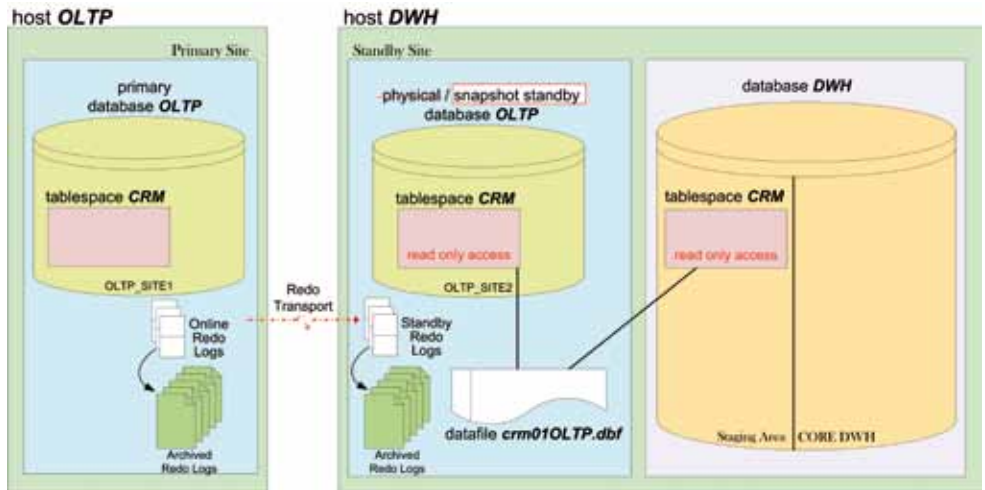


Abbildung 2: Auf dem DWH-Host ist die Datendatei crm01OLTP.dbf nun Teil beider Datenbanken (Read-only)

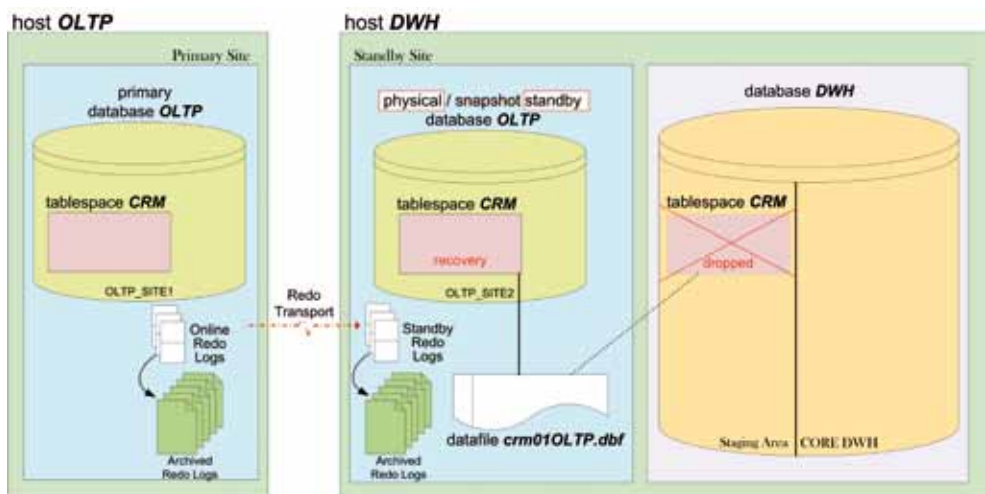


Abbildung 3: Tablespace CRM wird aus der DWH-Datenbank gelöscht und die Snapshot-Standby-Datenbank in eine physische Standby-Datenbank zurückkonvertiert

- Die Datendatei crm01OLTP.dbf ist jetzt Teil beider Datenbanken (Snapshot-Standby-OLTP und -DWH, siehe Abbildung 2)
- Das ETL-Programm kann nun die Daten aus der Staging Area lesen

Aktualisieren der Staging Area

Sobald nun neue Daten in den CRM-Teil der Staging Area geladen werden sollen, sind diese zu aktualisieren (siehe Abbildung 3):

- Der eingebundene Tablespace CRM wird aus der DWH-Datenbank gelöscht (dropped)
- Die Snapshot-Standby-Datenbank wird zurück in eine physische Standby-Datenbank konvertiert
- Gleichzeitig startet der Recovery-Prozess für alle Datendateien der OLTP-Standby-Datenbank

Diese Lösung hat folgende Vorteile:

- Die Staging Area beinhaltet alle Daten, nicht nur die Inkremente
- Es entsteht keine zusätzliche Last auf dem OLTP-Host
- Es werden keine Datendateien kopiert
- Die zwischen OLTP und DWH transferierte Datenmenge ist ausschließlich durch die Datenänderungen im OLTP-System bestimmt

- Redo-Log-Informationen werden fortlaufend übertragen, dabei ist es unerheblich, ob die Standby-Datenbank im Snapshot-Standby- oder im Physical-Standby-Modus ist
- Die Laufzeit des Aktualisierungsprozesses der Staging Area entspricht dem Refresh der Standby-Datenbank und ist nicht von der Größe des Tablespace abhängig, sondern nur von der Menge der Datenänderungen seit der letzten Aktualisierung
- Einmal aufgesetzt, ist der Betrieb dieser Lösung sehr einfach
- Es werden weder „Remote-Queries“ noch „Distributed Joins“ benötigt
- Unter Berücksichtigung des Einflusses auf das Quellsystem und der Last auf dem DWH-Host ist die hier vorgestellte Lösung die effizienteste
- Nur Redo-Logs werden benötigt, keinerlei weitere Strukturen
- Die Lösung arbeitet nur auf Basis von Datenblockänderungen, nicht auf SQL-Level

Dazu müssen einige technische Voraussetzungen erfüllt sein. Diese lassen sich in verschiedene Kategorien unterteilen:

- *Identische Zeichensätze in allen Datenbanken*
Um Transportable Tablespaces nutzen zu können, müssen die Zeichensätze der OLTP- und der DWH-Datenbank identisch sein.

- *Self-Contained Tablespace Sets*
Um eine Menge, also ein Set von Tablespaces, von einer zu einer anderen Datenbank zu transportieren, muss diese „Self-Contained“ sein. Das bedeutet, dass man kein Set von Tablespaces transportieren kann, das Objekte mit abhängigen Objekten wie zum Beispiel Materialized Views oder Tabellenpartitionen auf anderen Tablespaces enthält, solange diese Objekte nicht alle auf einmal in einem Set transportiert werden.
- *Benötigte Oracle-Datenbank-Releases*
Die OLTP-Datenbank muss unter Oracle 10g oder höher laufen. Dabei ist Oracle 0 empfehlenswert, da das Snapshot-Standby-Feature damit verfügbar ist. Mit Oracle 10g wird diese Funktionalität manuell nachgebildet, indem man einen garantierten Restore Point in der Standby-Datenbank erzeugt, bevor man sie im Read-Write-Mode öffnet. Das bedeutet folgende Einschränkungen mit 10g:
 - Es muss ein wenig programmiert werden, was allerdings keine große Sache und relativ simpel ist.
 - Der Redo-Transport zwischen der Primary- und der Standby-Datenbank wird während der Phase, in der letztere Read-Write geöffnet ist, angehalten. Mit 11g bleibt der Log-Transport dagegen die ganze Zeit über aktiv.

Um die Transportable Tablespaces nutzen zu können, muss die Data-Warehouse-Datenbank dasselbe oder ein höheres Release als die Standby-Datenbank haben.

- *Benötigte Oracle-Lizenzen*
Die Lösung benötigt eine Oracle-Datenbank-Enterprise-Edition-Lizenz für den OLTP-Host und den DWH-Host. Alle weiteren benötigten Features wie Data Guard, Transportable Tablespaces und die Snapshot-Datenbank (letztere ab 11g) sind darin bereits enthalten. Darüber hinaus ist keine weitere Option notwendig – weder Active Data Guard (eine kostenpflichtige Option von 11g mit „Real Time Query“ sowie inkrementellem Backup auf der Standby-Datenbank) noch Partitioning.

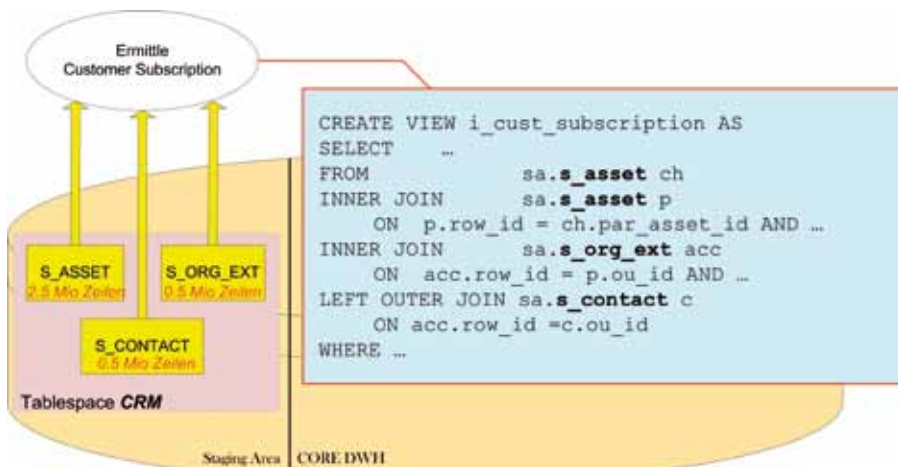


Abbildung 4: Der Transformationsprozess liest Siebel-Tabellen und transformiert die Daten in eine neue Struktur namens „Customer Subscription“

Ein Beispiel aus der realen Welt

Die vorgestellte Lösung arbeitet mit praktisch jeder Oracle-basierten Anwendung – sei es eine Standardanwendung oder eine Eigenentwicklung. Um den Ansatz an einem realistischen Beispiel zu demonstrieren, wird ein Auszug aus dem Datenbank-Schema einer CRM-Applikation namens „Siebel“ verwendet, einer weitverbreiteten CRM-Lösung von Oracle. Im Beispiel werden als repräsentative Auswahl der Siebel-Tabellen mit hoher Komplexität und Kardinalität die Tabellen „S_CONTACT“, „S_ORG_EXT“ und „S_ASSET“ verwendet.

Folgende typische Data-Warehouse-Situation wird angenommen: ETL-Prozesse sollen den Inhalt aus den Siebel-Tabellen extrahieren, in eine neue Zielstruktur namens „Customer Subscription“ transformieren und dafür eine entsprechende View verwenden (siehe Abbildung 4).

Das Data Warehouse muss dabei nicht nur jeweils den letzten gültigen Datensatz in „Customer Subscription“ speichern, sondern alle jemals bekannten historischen Zustände. Der ETL-Prozess muss also immer den neuesten „Abzug“ der drei Tabellen mit dem Inhalt im Core-DWH vergleichen, dann – sofern Änderungen, Löschungen oder neue Datensätze gefunden werden – neue Versionsdatensätze im Zielmodell erzeugen und schließlich die alten Datensätze als „ab jetzt ungültig“ markieren. Dieses Verfahren nennt man „Historisierung“ (siehe Abbildung 5).

Angenommen, der Data-Warehouse-Architekt trifft folgende Design-Entscheidungen:

- Wegen zahlreicher Abhängigkeiten innerhalb der Quellsystem-Tabellen und -Prozesse soll die Staging Area immer den kompletten Satz an Daten enthalten und nicht nur die Inkremente
- Der Transfer von vielen Millionen Records pro Nacht ist nicht akzeptabel
- Im Quellsystem gibt es keine Datensatz-Marker oder Journal-Tabellen, die eine zuverlässige Identifikation von Inkrementen ermöglichen

- Der Architekt entscheidet sich für den Einsatz der hier vorgestellten Lösung

Wegen der hohen Kardinalität der Daten (viele Millionen Datensätze) wird zudem eine gute Skalierbarkeit des zugrundeliegenden DWH-Datenmodells angenommen. Die wichtigsten Schritte zum Aufbau und Betrieb der Lösung sehen folgendermaßen aus: Oracle Data Guard wird entsprechend den bereits erwähnten Vorgaben aufgesetzt. Es wird eine Data-Guard-Broker-Konfiguration erzeugt, wobei der Protection-Mode auf „Maximum Performance“ (Default) belassen und der Log-Transport auf „asynchron“ gesetzt wird. In den beiden Datenbanken DWH und OLTP wird die Rolle „dwh_sa_crm_role“ erzeugt:

```
CREATE ROLE dwh_sa_crm_role;
```

Als nächstes wird dieser Rolle in der OLTP-Datenbank das SELECT-Privileg für alle relevanten Siebel-Tabellen eingeräumt:

```
GRANT SELECT ON s_contact TO dwh_sa_crm_role;
GRANT SELECT ON s_org_ext TO dwh_sa_crm_role;
GRANT SELECT ON s_asset TO dwh_sa_crm_role;
```

Schließlich muss noch der Eigentümer der transportierten Tabellen in der Data-Warehouse-Datenbank erzeugt werden:

```
CREATE USER crm IDENTIFIED BY thisIsASecretPassword;
```

Dieser User benötigt im Übrigen weder das „create session“- noch das „create table“-Privileg.

Solange der Aktualisierungsprozess ausgeführt wird, sind die CRM-Daten in der Staging Area nicht verfügbar. Die Datendateien des CRM-Tablespace auf dem DWH-Host sind gegenwärtig exklusiv der physischen Standby-Datenbank des OLTP-Systems (hier OLTP_SITE2) zugeordnet. Das Konzept kann natürlich dahingehend erweitert werden, mehrere Tablespaces sogar von mehreren Datenbanken zu transferieren.

Um den Aktualisierungsprozess zu beenden, sind die folgenden Schritte nötig:

Zunächst wird die physische Standby-Datenbank in eine Snapshot-Standby-Datenbank konvertiert sowie anschließend der betroffene Tablespace von OLTP_SITE2 auf Read-only gesetzt und via Datapump und Datenbank-Link ins Data-Warehouse eingebunden:

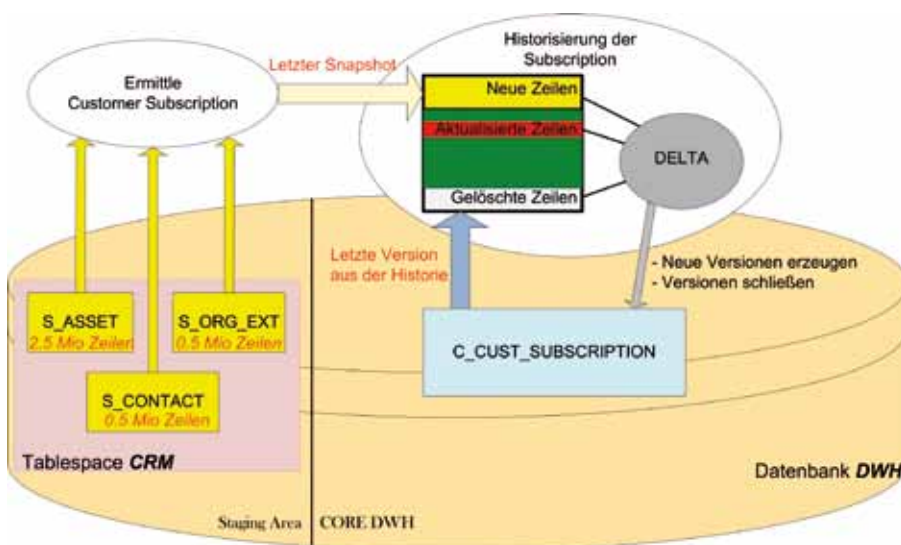


Abbildung 5: ETL vergleicht den neuen „Abzug“ der drei „Customer Subscription“-Tabellen mit den letzten bekannten Daten aus dem Core Data Warehouse und erzeugt wenn nötig neue Versionen der Daten

```
SQL> alter tablespace crm read
only;
# impdp system@DWH logfile=imp_
crm.log network_link=OLTP_SNAP
transport_tablespaces=CRM
transport_datafiles=d:\oradata\
oltp\crm01oltp.dbf
```

Wichtig: Sofern 11.2.0.2 genutzt wird, ist es aufgrund von Bug 10185688 nötig, entweder XDB in der Quelldatenbank oder den verfügbaren Patch zu installieren.

Beim Transfer können die zu transferierenden Tabellen angegeben werden. Außerdem kann man wählen, ob man Indizes, Objekt-Privilegien, Tabellen-Trigger oder Tabellen-Constraints importieren möchte.

Um den Aktualisierungsprozess zu starten, sind folgende Maßnahmen erforderlich:

Zunächst wird der CRM-Tablespace aus der DWH-Datenbank entfernt:

```
SQL> drop tablespace crm inclu-
ding contents;
DGMGRL> convert database ,OLTP_
SITE2' to physical standby
```

Es wird nun geprüft, ob die Daten im CRM-Tablespace der Standby-Datenbank wieder aktuell genug für eine erneute Einbindung in das DWH sind. Dazu wird der Data Guard Broker befragt:

```
DGMGRL> show database ,OLTP_
SITE2';
Database - OLTP_SITE2
  Role:          PHYSICAL
STANDBY
  Intended State: APPLY-ON
  Transport Lag:  0 seconds
  Apply Lag:     11 minutes
23 seconds
  Real Time Query: OFF
  Instance(s):
    oltp

Database Status:
SUCCESS
```

Als Aktualisierungsmethode setzt die Standby-Datenbank das äußerst effiziente parallele Media-Recovery ein.

Es arbeitet blockorientiert und ist damit deutlich schneller und Ressourcen-schonender als die Mechanismen von GoldenGate oder Streams, bei denen die Änderungen aus den Logs extrahiert und per SQL Datensatz für Datensatz abgearbeitet werden.

Fazit

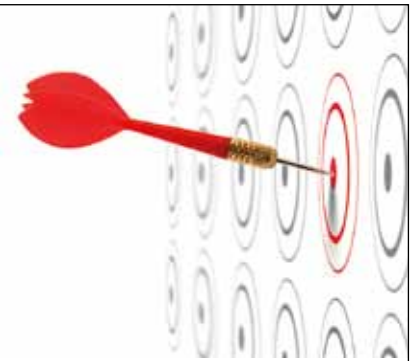
Eine Staging Area profitiert von der beschriebenen Lösung in folgenden Fällen am meisten:

- Das Quellsystem weist große Datenmengen mit komplexen Beziehungen auf: Es gibt Datenextraktion aus dedizierten Online-Transaction-Applikationen wie CRM oder SCM (Supply Chain Management), bei denen mehrere Millionen individuelle Kunden, Verträge, Produkte oder Produktkomponenten, Warenlager etc. verwaltet werden
- Die Anforderungen an die Aktualität im Data Warehouse sind besonders hoch

Literatur und Links

1. Trivadis Whitepaper „Solution for Staging Area in Near Real-Time DWH – Efficient in Refresh and Easy to Operate“: http://www.trivadis.com/uploads/tx_cabagdownloadarea/WhitePaper_Solution_for_Staging_Area_01.pdf
2. Oracle Data Guard Concepts and Administration: http://download.oracle.com/docs/cd/E11882_01/server.112/e17022/toc.htm
3. Oracle Database PL/SQL Packages and Types Reference – Chapter 46: http://download.oracle.com/docs/cd/E11882_01/appdev.112/e16760/d_datmpm.htm
4. Data Warehousing mit Oracle – Business Intelligence in der Praxis. Chapter 3.4. Jordan et al., Hanser, 2011

Karol Hajdu
Trivadis Delphi GmbH
karol.hajdu@trivadis.com
Mathias Zarick
mathias.zarick@trivadis.com



Berliner Experten- seminare

- Wissensvertiefung für Oracle-Anwender
- Mit ausgewählten Schulungspartnern
- Von Experten für Experten
- Umfangreiches Seminarangebot

Termine

28./29. März 2012
Backup & Recovery
Referent: Marco Patzwahl

18./19. April 2012
Apex
Referent: Peter Raganitsch

23./24. Mai 2012
Oracle VM 3.0
Referent: Martin Bracher

30./31. Mai 2012
Oracle Partitionierung
Referent: Klaus Reimers

5./6. Juni 2012
Solaris11, Container „deep dive“
Referent: Heiko Stein

DOAG
Deutsche ORACLE-Anwendergruppe e.V.

www.doag.org

Tipps und Tricks aus Gerds Fundgrube

Heute: Generisches List-Item, Teil 2

Gerd Volberg, OPITZ CONSULTING GmbH

In der letzten Ausgabe wurde das generische List-Item vorgestellt, mit dem Daten beliebiger Queries in einem List-Item dargestellt werden können. Teil 2 zeigt nun, wie diese Technik verwendet werden kann, um aus einem Foreign Key ein Lookup-Feld zu machen.

Gegeben sei ein Block auf Basis der Tabelle EMP (siehe Abbildung 1). Die Felder „EMPNO“, „ENAME“ und „JOB“ sind Standard-Text-Items. Der Item-Typ des Feldes „DEPTNO“ wird nun auf „List-Item“ geändert.

EmpNo	EName	Job	DeptNo
7369	SMITH	CLERK	20
7499	ALLEN	SALESMAN	30
7521	WARD	SALESMAN	30
7566	JONES	MANAGER	20
7654	MARTIN	SALESMAN	30
7698	BLAKE	MANAGER	30
7782	CLARK	MANAGER	10

Abbildung 1: Employees-Datenblock (vorher)

Im „WHEN-NEW-FORM-INSTANCE“-Trigger muss das List-Item mit den Abteilungsnamen aus „DEPT“ gefüllt werden. Von diesem Zeitpunkt an werden die Abteilungsnamen und nicht mehr die Abteilungsnummern angezeigt (siehe Abbildung 2).

```
Init_List_Item (P_Item => ,EMP.DEPTNO',
               P_Query => ,SELECT initcap (DNAME), to_char (DEPTNO)' ||
                          , FROM DEPT ORDER BY DEPTNO');
Go_Block (,EMP');
Execute_Query;
```

EmpNo	EName	Job	DName
7369	SMITH	CLERK	Research
7499	ALLEN	SALESMAN	Sales
7521	WARD	SALESMAN	Sales
7566	JONES	MANAGER	Research
7654	MARTIN	SALESMAN	Sales
7698	BLAKE	MANAGER	Sales
7782	CLARK	MANAGER	Accounting

Abbildung 2: Employees-Datenblock (nachher)

Der Sourcecode zu „Init_List_Item“ steht in der letzten DOAG News oder kann unter <http://code.google.com/p/forms-framework/> heruntergeladen werden (Library „frw_lib.pll im Archiv 111205_frw.zip“). Der Einsatz dieser Technik sorgt dafür, dass die Abteilungsnamen nicht in einem POST-QUERY selektiert werden müssen (siehe Abbildung 3). Damit ist diese Technik sehr performant und erzielt eine Geschwindigkeit, die sonst nur bei Verwendung einer View als Basetable erzielt werden kann.

Diese Technik sollte immer dann genutzt werden, wenn die Lookup-Tabelle

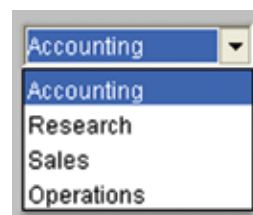


Abbildung 3: Lookup-Werte für DName

nur wenige Datensätze beinhaltet. Die Usability steigt damit enorm an, da List-Items viel schneller bedient werden können als LOVs.

Gerd Volberg
 gerd.volberg@opitz-consulting.com
 talk2gerd.blogspot.com



Impressum

Herausgeber:

DOAG Deutsche ORACLE-
 Anwendergruppe e.V.
 Tempelhofer Weg 64, 12347 Berlin
 Tel.: 0700 11 36 24 38
 www.doag.org

Verlag:

DOAG Dienstleistungen GmbH
 Fried Saacke, Geschäftsführer
 info@doag-dienstleistungen.de

Chefredakteur (ViSDP):

Wolfgang Taschner
 redaktion@doag.org

Chefin von Dienst (CvD):

Carmen Al-Youssef
 office@doag.org

Titel, Gestaltung und Satz:

Claudia Wagner
 DOAG Dienstleistungen GmbH

Titelfoto: Bundesagentur für Arbeit**Anzeigen:**

CrossMarketteam Ralf Rutkat, Doris Budwill
 www.crossmarketteam.de
 Metadaten und Preise finden Sie unter:
 www.doag.org/publikationen/

Druck:

adame Advertising and Media
 GmbH Berlin, www.adame.de



Die DOAG veranstaltet die größte Oracle-

Die DOAG 2011 Konferenz + Ausstellung vom 15. bis 17. November 2011 im CongressCenter Ost in Nürnberg war an drei Tagen das Zentrum der Oracle-Community.

Die erfahrenen Referenten und Keynote-Speaker sprachen über alle Themen, die die Oracle-Welt gerade bewegen: Ob Neuerungen, Erfahrungsberichte oder Fachvorträge – in den sechs Haupt-Streams stand das gesamte Wissen der Oracle-Community bereit. Ein anschließender Schulungstag rundete die Veranstaltung ab. Als ein Highlight wurde die neue Version 12 des Enterprise Managers erstmals in Deutschland vorgestellt.

Die 24. Auflage der Anwenderkonferenz hat in diesem Jahr neue Rekorde aufgestellt: Noch nie waren am ersten Konferenztag so viele Teilnehmer auf der Konferenz anwesend. „Die DOAG

2011 Konferenz + Ausstellung ist ein Erfolg auf der ganzen Linie“, resümiert Dr. Dietmar Neugebauer, Vorstandsvorsitzender der DOAG. „Unser Konzept hat sich bewährt.“

Der Austausch mit Oracle stand im Mittelpunkt der Q&A-Session: Jürgen Kunz, Senior Vice President und verantwortlich für die ORACLE Deutschland B.V. & Co. KG, und Günther Stürner, Vice President ORACLE Deutschland B.V. & Co. KG, haben Mängel im Configuration Manager eingeräumt und angekündigt, diese vom Product Development prüfen lassen zu wollen.

Absolutes Highlight war die Abendveranstaltung am Mittwochabend mit Rolf Stahlhofen und Freunden. „Die DOAG Konferenz hat noch nie so ge-groovt. Einfach genial!“, kommentiert ein Teilnehmer den Abend.

Auch für immer mehr Schweizer sind die drei Tage im November nicht mehr aus der Agenda wegzudenken, an denen sich in Nürnberg jährlich die deutschsprachige Oracle-Gemeinschaft trifft. Hans-Jörg Büttler von der Swiss Oracle User Group (SOUG) schildert seine Eindrücke: „Auch dieses Jahr fand am Dienstag ein Schweizer-Abend der besonderen Art statt. Nach der letzten Vortrags-Session trafen sich die rund 50 Schweizer und Freunde zum Apéro im Foyer der Konferenz. Gestärkt und gespannt der Geschichten um das rote Bier ging es danach, auf einer kurzen Stadtrundfahrt mit interessanten Erläuterungen zu Gebäuden und Geschichte von Nürnberg, Richtung Altstadt. Nach einem kurzen Fußmarsch führten uns die Treppen in den Untergrund! Unter kundiger Leitung erfuhren wir, wann und wieso in Nürnberg ein bis zu vier





Highlights 2011

- Beitragsreform
- Beiratssitzung 02/11
- Neustrukturierung + außerordentliche
- Print und Online Medien, Mitgliederum
- Neuer Internetauftritt
- Application Konferenz
- BIOW Konferenz

Konferenz in Europa

Ebenen tiefes Keller- und Gang-System in den Sandstein gehauen wurde. Dabei wurde an sämtlichen Tischen rege über das eine oder andere Orakel diskutiert, bis sich dann gegen Mitternacht die Letzten auf den Weg zum Hotel machten. An dieser Stelle den Organisatoren und Sponsoren (Bison, Diso, iRIX, Oracle) nochmals herzlichen Dank!“

Positives Feedback der Aussteller

Die gut besuchte Ausstellung überzeugte in diesem Jahr wieder mit etwa 60 Ausstellern aus aller Welt, die ihre Produkte und Dienstleistungen rund um den Einsatz von Oracle vorstellten. „Vielen Dank für die optimale Organisation und Betreuung. Nicht nur für die Besucher, auch für uns als Aussteller war die DOAG ein Highlight. Hier erreichen wir unsere Zielgruppe. Be-

sonders in Kombination mit den Vorträgen. Wir freuen uns schon jetzt wieder auf den regen Erfahrungsaustausch mit der Oracle-Community in 2012“, resümierte Erwin Pitwell, Salesmanager der Libelle AG.

„Die DOAG 2011 Konferenz + Ausstellung war erneut eine erfolgreiche Veranstaltung für TREvisto. Das hohe Niveau und die Bandbreite der Vorträge bringen anspruchsvolle Besucher zur DOAG, die sich an unserem Stand über unsere Kompetenzen im Bereich DWH und BI informieren konnten“, so das Fazit von Birgit Linsenmaier von der TREvisto GmbH.

„Wir waren als junges Unternehmen zum ersten Mal auf der DOAG, haben uns gleich für 2012 wieder angemeldet und werden auch Themenkonferenzen besuchen. Das spricht wohl für sich. Mein Dank für die hervor-

gende Organisation und an die Interessenten, die sich über unsere Lösungen informiert haben“, so Wolfgang Hack von dimensio informatics GmbH.

„Für uns als Schweizer Unternehmen bietet die DOAG-Konferenz eine interessante Plattform für den Zugang zum deutschen Markt. Unser Framework findet positive Resonanz“, schließt Kurt Meier, Director der Bison Schweiz AG den Reigen ab.

Zum Vormerken

Die DOAG 2012 Konferenz + Ausstellung wird vom 20. bis 22. November wieder im Nürnberg Convention Center (NCC) stattfinden. Mitgliedern, die sich für die Oracle Applications interessieren, sei die DOAG 2012 Applications vom 8. bis 10. Mai 2012 in Berlin empfohlen.

Fotos: Anne Eilers





Dr. Dietmar Neugebauer,
Vorstandsvorsitzender der DOAG

Erste Sitzung des neuen DOAG-Leitungsteams

In seiner ersten konstituierenden Sitzung am 9. Dezember 2011 in Berlin legte das DOAG-Leitungsteam, bestehend aus dem auf der letzten Mitgliederversammlung neu gewählten Vorstand und den Leitern der vier Communities, die Zuständigkeiten im Vorstand neu fest. So übernimmt Ralf Kölling zusätzlich zu seiner bisherigen Rolle als IT-Koordinator den Bereich „Finanzen und Beteiligungen“ und löst hier Stefan Kinnen ab, der jetzt die Development Community leitet. Franz Hüll, neuer stellvertretender Vorstandsvorsitzender, ist weiterhin für die Koordination der Regionalgruppen zuständig.

Die Competence Center bleiben unter der Obhut von Michael Paege, der ebenfalls zum stellvertretenden Vorstandsvorsitzenden gewählt wurde. Als Geschäftsführer der DOAG Dienstleistungen GmbH ist Fried Saacke im Vorstand weiterhin verantwortlich für die Geschäftsstelle und die Hochschul-Themen. Darüber hinaus vertritt er die Interessen der DOAG in Vereinen, bei denen die DOAG Mitglied ist, wie im Interessenverbund der Java User Groups e.V. (iJUG), in der Deutschsprachigen MySQL-Anwendergruppe e.V. (DMYSQLAG), in der Nürnberger Initiative für die Kommunikationswirtschaft e.V. (NIK) sowie in der Logistik-Initiative Hamburg e.V.

Im Rahmen der Leitungssitzung haben die vier Community-Leiter, die bereits im September benannt wurden, die Mitglieder ihrer Leitungsteams vorgestellt. Diese hat der Vorstand

einstimmig bestätigt. Weiterhin wurden die Festlegung der regionalen Repräsentanten, der Repräsentanten der Competence Center, der Repräsentanten der Hochschulgruppen sowie die Ansprechpartner in den Hochschulgruppen beschlossen.

In diesem Zusammenhang wurde in der Sitzung auch nochmals der Ablauf der Mitgliederversammlung 2011 diskutiert. Die Möglichkeiten, sich tagsüber bereits über die Arbeit der DOAG zu informieren und dadurch am Abend die Abstimmungen und Diskussionen zur Vereinsarbeit kompakter durchzuführen, sind gut angekommen.

Das Leitungsteam ging eingehend auf den Rückblick zur DOAG 2011 Konferenz + Ausstellung ein. Die Veranstaltung war sehr erfolgreich und die Teilnehmerzahlen lagen über denen der Vorjahre. Das Feedback war überaus positiv, im Besonderen waren die ausländischen Referenten von der fachlichen und organisatorischen Qualität sehr beeindruckt.

Für die DOAG 2012 Konferenz + Ausstellung beschloss das Leitungsteam einstimmig aufgrund der gestiegenen Basiskosten eine Erhöhung der Teilnehmergebühren um fünf Prozent. Für DOAG-Mitglieder wird das Ticket weiterhin um zwanzig Prozent reduziert angeboten. Zudem wird es bis 31. Mai 2012 ein Super-Saver-Ticket geben, das dem Frühbucherpreis von 2011 entspricht.

Die Programmgestaltung der Konferenz ist wieder Thema der Beiratssitzung Anfang Februar 2012. Dort soll die Grobplanung des Konferenzprogramms in den drei Communities erfolgen. Auch die einzelnen Streams und die Aufteilung der Slots werden durch die Communities im Rahmen der Beiratssitzung festgelegt. Übergreifende Keynotes sowie den Management Stream plant der Vorstand.

In diesem Zusammenhang wird im Leitungsteam nochmals das Kooperationsangebot der Deutschen Bahn vorgestellt (99 Euro für alle Bahnreisen zu DOAG-Veranstaltungen). Das Kooperationsangebot der Deutschen Bahn wird einstimmig für ein Jahr angenommen und an alle Mitglieder kommuniziert.

Themen des zweiten Tags der Leitungssitzung waren die Vorbereitung

der Beiratssitzung 2012, das Projekt „Satzungsänderung“ und die Ziele 2012. Schwerpunkt der diesjährigen Beiratssitzung wird die Arbeit in den vier Communities sein. Hier sollen die Planung von Veranstaltungen und Publikationen, die Zusammenarbeit mit den Regional- und Querschnittsgruppen sowie Maßnahmen zur Aktivierung der Mitgliederkommunikation erarbeitet werden.

Zum Projekt „Satzungsänderung“ beschloss das Leitungsteam einen groben Fahrplan. Ein Arbeitskreis wird zunächst zusammen mit dem DOAG-Syndikus Carsten Diercks Ideen sammeln und bewerten. Die ersten Ergebnisse werden auf der Beiratssitzung vorgestellt. Ziel ist es, den Vorschlag zur neuen Satzung ab September an die Mitglieder zu kommunizieren und dann auf der Mitgliederversammlung 2012 zu verabschieden.

Zentrales Ziel im Jahr 2012 ist es, den Mehrwert der neuen Struktur für die DOAG-Mitglieder zu etablieren. Nachdem sich durch die Einrichtung der Communities die interne Zusammenarbeit verbessert hat, muss nun daran gearbeitet werden, den Nutzen der neuen Struktur für Mitglieder am besten sichtbar zu machen, damit jedes Mitglied seine Heimat bei der DOAG in den Communities findet.



Michael Paege, Stellvertretender Vorstandsvorsitzender der DOAG
Leiter Competence Center Lizenzfragen

Arbeitskreis Lizenzierung hat Arbeit aufgenommen

Unter der Leitung von Michael Paege hat sich im Rahmen der DOAG 2011

Konferenz + Ausstellung erstmals ein Team getroffen, um die Ziele des neugegründeten Arbeitskreises „Lizenzierung“ zu definieren. Die Gruppe will als Vermittler zwischen Anwendern und Oracle fungieren und das vorhandene Know-how aus unterschiedlichen Bereichen zusammenzuführen, um Meinungen und Anregungen der Community gesammelt an Oracle zu kommunizieren. Wichtig ist auch die bessere Information der Mitglieder über die Oracle-Lizenzierungsregeln und Business Practices.

Als erste Aktion plant die Gruppe einen DOAG Licensing-Guide, der die Oracle-Lizenzierungsregeln ausführlich, verständlich und einfach erklären soll. Neben klassischen Themen der Lizenzierung von Oracle-Produkten sollen auch Randthemen wie Lizenzierung unter SAP, Royalty-Modelle, Lizenzmanagement und Tool-Unterstützung behandelt werden. Obgleich der Schwerpunkt zuerst im Bereich „Datenbank“ liegt, will das Team auch die Lizenzierungsaspekte der Middleware unter die Lupe nehmen. Thematisiert werden in dem Guide zudem die vertraglichen Hintergründe, Grundlagen und Dokumente der Lizenzierung, Lizenzmodelle und -metriken sowie Datentransfer und Architekturen. Zusätzlich zu dem Guide soll auch eine Übersicht über verfügbare Features und deren Lizenz Auswirkungen entstehen. Auch das Thema „Kostenoptimierung“ soll innerhalb der Gruppe Beachtung finden.

Info: Der Arbeitskreis braucht Unterstützung! Wer aktiv mitwirken

möchte, kann sich mit Michael Paege (michael.paege@doag.org) in Verbindung setzen.



Dr. Frank Schönthaler
Leiter der Business Solutions Community

News aus der Business Solutions Community

Vom 8. bis zum 10. Mai 2012 findet bereits zum dritten Mal das Top-Event der DOAG Business Solutions Community in Berlin statt, die DOAG 2012 Applications. Der „Call for Presentations“ ist bereits beendet und die Bewertung der Einreichungen hat begonnen. Die Keynote-Speaker stehen im Wesentlichen fest und die finale Programmabstimmung erfolgt Mitte Februar 2012. Experten und erfahrene Anwender werden auf der Konferenz zeigen, wie die Oracle-Applikationen mit den zugrunde liegenden Technologien richtig konfiguriert werden. Sie geben ihre Erfahrungen aus Einführungs- und Migrationsprojekten und der täglichen Anwendung weiter und sind offen, ihre

Erkenntnisse mit den Konferenzbesuchern zu diskutieren. Und ein Highlight vorweggenommen: Im Rahmen der Konferenz wird der Oracle-Cloud-Computing-Launch stattfinden (siehe <http://bs.doag.org/de/events-bs/konferenzen/doag-2012-applications.html>).

Auf der Fachkonferenz DOAG 2012 Logistik & SCM am 5. Juni 2012 in Hamburg dreht sich alles darum, welche IT-Systeme, Werkzeuge und Methoden geeignet sind, um die notwendige Flexibilität der Unternehmen zu gewährleisten. Die Besucher erwarten spannende Vorträge zu den Themenbereichen Intralogistik sowie Transport und Dienstleistung. Erfahrene Referenten werden Einblicke in Theorie und Praxis gewähren. Weitere Informationen stehen auf <http://bs.doag.org/de/events-bs/konferenzen/doag-2012-l-scm.html>.

Neben eigenen Veranstaltungen engagiert sich die DOAG Business Solutions Community auch als Partner auf dem Kongress der Integrata-Stiftung, der vom 10. bis zum 12. Mai 2012 ebenfalls in Berlin stattfindet. Unter dem Motto „Mehr Lebensqualität durch Informationstechnologie“ hat die Stiftung ein hochinteressantes Programm zusammengestellt, das mit Keynotes bekannter Persönlichkeiten gespickt ist. Unter anderem wird der Staatsminister a.D. und Präsident der Deutschen Gesellschaft für Philosophie, Julian Nida-Rümelin, zum Thema „Politische Partizipation“ referieren. Weitere Informationen unter <http://bs.doag.org/de/events-bs/partnerveranstaltungen/integrata-kongress.html>.

Newsticker

Database Firewall mit verbesserten Reporting-Funktionen und nun auch für MySQL verfügbar

Die neue Version, die in Echtzeit vor Bedrohungen wie SQL-Injektionen und unerlaubten Zugriffen schützt, unterstützt nun auch die MySQL Enterprise Edition. Neben dem Betrieb mit Oracle Database ist außerdem die Nutzung mit IBM DB2 auf Linux, Unix oder Windows, Microsoft SQL Server, Sybase Adaptive Server Enterprise (ASE) und Sybase SQL Anywhere möglich. Oracle stellt in der neuen Version eine neue Reporting-Infrastruktur zur Verfügung, die das Ausführen und Verändern von Layouts bestehender Berichte ermöglicht. BI-Publisher-Anwender können so die Vorteile der ihnen vertrauten Desktop-Produkte nutzen, um formatierte Dokumente zu erstellen, zu verwalten und bereitzustellen. Das neue Release bietet auch eine neue Proxy-Betriebsart, die Anwendern noch mehr Flexibilität bei einer Bereitstellung auf dem Netzwerk ermöglicht. Darüber hinaus gewährleistet die Firewall eine bessere Skalierbarkeit: Aufgrund der neuen „Multi Core“-Management-Funktionen ist für Administratoren eine fein granulいたete Kontrolle über die Ressourcen möglich, die zum Schutz von bestimmten Datenbanken eingesetzt werden.

Um eine umfassende Defense-in-Depth-Sicherheitsstrategie zu garantieren, integriert Database Firewall jetzt auch Oracle Advanced Security und ermöglicht somit einen verschlüsselten Datenverkehr, der wiederum auf mögliche Bedrohungen überwacht werden kann. Zur Ergänzung ihrer Security-Vorkehrungen können Anwender neben der Database Firewall auch Oracle Advanced Security, Audit Vault und Database Vault nutzen.



08.02.2012

SIG Development

Mobile Computing im Oracle-Umfeld
Andreas Badelt, Christian Schwitalla
sig-development@doag.org

10.02.2012

DOAG Webinar

ORACLE Datenbank & ORACLE Lizenz
DOAG Geschäftsstelle
office@doag.org

13.02.2012

Regionaltreffen München/Südbayern

Franz Hüll, Andreas Ströbel
regio-muenchen@doag.org

16.02.2012

Regionaltreffen Würzburg

Oliver Pyka
regio-wuerzburg@doag.org

20.02.2012

Regionaltreffen Bremen

Ralf Kölling
regio-bremen@doag.org

21.02.2012

Regionaltreffen Hamburg/Nord

Stefan Thielebein
regio-nord@doag.org

27.02.2012

**Regionaltreffen Osnabrück/
Bielefeld/Münster**

Andreas Kother, Klaus Günther
regio-osnabrueck@doag.org

01.03.2012

SIG Java

Andreas Badelt
sig-java@doag.org

01.03.2012

SIG Database OEM

Johannes Ahrends, Christian Trieb
sig-database@doag.org

01.03.2012

**Regionaltreffen Trier/Saarland/
Luxemburg**

Bernd Tuba, Holger Fuchs
regio-trier@doag.org

06.03.2012 / 07.03.2012

**Berliner Expertenseminar:
Oracle Security**

Pete Finnigan zum Thema Oracle Security
Cornel Albert
expertenseminare@doag.org

06.03.2012

Regionaltreffen Jena/Thüringen

Jörg Hildebrandt
regio-thueringen@doag.org

07.03.2012

Regionaltreffen Berlin/Brandenburg

Michel Keemers
regio-bb@doag.org

08.03.2012

SIG Infrastruktur

Virtualisierung
Björn Bröhl
sig-infrastruktur@doag.org

14.03.2012

SIG SOA

Torsten Winterberg, Hajo Normann
sig-soa@doag.org

20.03.2012

SIG Security

Franz Hüll
sig-security@doag.org

22.03.2012

Regionaltreffen Stuttgart

Jens-Uwe Petersen
regio-stuttgart@doag.org

22.03.2012

Regionaltreffen Rhein-Neckar

Kai F. Christianus
regio-rhein-neckar@doag.org

22.03.2012

Regionaltreffen München/Südbayern

Franz Hüll, Andreas Ströbel
regio-muenchen@doag.org

28.03.2012

Regionaltreffen NRW

Stefan Kinnen
regio-nrw@doag.org

28.03.2012 / 29.03.2012

**Berliner Expertenseminar:
Backup & Recovery**

Marco Patzwahl zum Thema
Backup & Recovery
Cornel Albert
expertenseminare@doag.org

29.03.2012

Regionaltreffen Dresden/Sachsen

Helmut Marten
regio-sachsen@doag.org

Aktuelle Termine und
weitere Informationen finden Sie unter
www.doag.org/termine