



*Liebe Mitglieder der Deutschen ORACLE-Anwendergruppe,
liebe Leserinnen und Leser,*

schneller, höher, weiter – welcher Hersteller hat die größte Datenbank? Welches Projekt verarbeitet den größten Datenbestand? Welcher DBA hat den größten Storage im Keller? „Arbeiten mit großen Datenbanken“ ist das Schwerpunktthema der vorliegenden Ausgabe. Die Artikel sollen Ihnen beim Umgang mit großen Datenmengen Lösungen und Anregungen zu einigen ausgewählten Fragestellungen aufzeigen. Dazu zählen zum Beispiel Strategien bei der Definition der Partitionierung von Datenbank-Objekten. Dies sind Entscheidungen, die bereits vor dem Aufbau solcher Datenbanken getroffen werden sollten.

Wie sich solche Festlegungen auf die Performance und den späteren Betrieb von großen Datenbanken auswirken, beleuchten gleich mehrere Artikel. Gerade dieses Wachstum der Daten-Inhalte macht die Betriebsaufgaben eines DBAs immer schwieriger. Solche Datenbanken zu reorganisieren, ist sicherlich keine leichte Aufgabe.

Das Thema „Very large databases“ ist auch eines der Hauptthemen von Oracle 11g. Hier bietet Oracle mit dem Feature Advanced Compression eine beträchtliche Storage-Einsparung, vor allem bei der zunehmenden Menge von unstrukturierten Daten.

Die ausgewählten Artikel können Ihnen hoffentlich genügend Anregungen für Ihre Aufgaben geben. Auf jeden Fall wünsche ich Ihnen viel Spaß beim Lesen und beim Ausprobieren des einen oder anderen Beispiels.

Ihr

Dr. Dietmar Neugebauer

Vorstand und Leiter Regionalgruppe München/Südbayern



Virtualisierung mit Oracle VM: Machen Sie es sich bequem!



- Skalierbare Servervirtualisierung für Oracle- und Nicht-Oracle-Applikationen
- Unterstützt Windows- und Linux-Umgebungen
- Weniger Hardware und bessere Auslastung verringern Betriebs- und Wartungskosten
- Vereinfachte Administration, z. B. durch Parallelbetrieb diverser DB-Versionen auf derselben Maschine
- Hohe Performance, ideal für den Einsatz in Produktionsumgebung und Rechenzentrum
- Bessere Ausnutzung von Ressourcen, Overhead gegenüber nicht-virtualisierten x86-Betriebssystemen drei Mal geringer
- Oracle VM ist lizenzkostenfrei! Wir bieten Ihnen einen besonders günstigen Support, z. B. für die optimale Installation und Konfiguration.
- HUNKLER ist Oracle-Partner der ersten Stunde – seit 20 Jahren Ihr kompetenter und zuverlässiger Ansprechpartner.



Hauptsitz Karlsruhe Bannwaldallee 32 76185 Karlsruhe Tel. 0721-490 16-0 Fax 0721-490 16-29 info@hunkler.de
 Geschäftsstelle Bodensee Lohnerhofstraße 2 78467 Konstanz Tel. 07531-8926-2492 Fax 07531-8926-2493 www.hunkler.de

Aus der DOAG

- 3** Editorial
Dr. Dietmar Neugebauer
- 6** Tagebuch
Fried Saacke
- 7** Die DOAG in der Presse
Wolfgang Taschner
- 10** Interview mit Jochen Boekel
„Jeden Tag entwickeln wir einen neuen Kurs ...“

Arbeiten mit großen Datenbanken

- 13** Zwei Themen für große Datenbanken:
Partitionierung und NOLOGGING-Operationen
Dierk Lenz
- 16** Sizing und Architektur einer 30 bis 60 TB großen Datenbank
Björn Brühl
- 19** Komprimierung in der Oracle Datenbank
Ulrike Schwinn und Thomas Niewel
- 23** Leitlinien zur Entwicklung und zum Betrieb
eines Multi-Terabyte-Data-Warehouses
Dr. Kurt Franke
- 31** Alternativen zum Hash-Join beim Zugriff
auf sehr große Datenmengen
Daniel Günther
- 39** Well Balanced Systems
Marcus Bender und Kai Fischer
- 42** Kampf dem Schweizer Käse – oder wozu reorganisieren?
Patrick Schwanke
- 46** Datenbank 11g macht den FULL OUTER JOIN tauglich
für das Data Warehouse
Karol Hajdu
- 49** Oracles HARD-Initiative
Peter Heumel
- 51** Langzeitspeicherung und Trendanalyse von Statistik-
und Performance-Daten mit einem RRD-Tool
Félix Castillo Sánchez

Datenbanken

- 54** Oracle 11g versus SQL Server 2008
Timo Meyer
- 57** Partitioning ist eine Frage der richtigen Strategie
André Schulte

Entwicklung

- 61** Einfaches Reporting mit OpenOffice Calc
Dr. Heike Püttmann

Infrastruktur

- 64** Service-orientierte Architekturen mit Oracle –
eine Vorschau auf die Oracle SOA Suite 11g
Christian Schmülling und Ingo Meier

Best Practice

- 67** Job Notification Package
Vladimir Poliakov

Trends & Tendenzen

- 69** Nichts geht mehr ohne Virtualisierung
Lutz Pöbneck und Dr. Dietmar Müller

Security

- 72** Oracle Security Corner
Alexander Kornbrust

Tipps & Tricks

- 75** Heute: JavaScript-API in Forms 11g
Gerd Volberg

Buchbesprechung

- 76** Secret ORACLE
Jaroslav Dutov

Aus der DOAG

- 63** Unsere Inserenten
- 77** Oracle Designer – Tool mit Zukunft?
Christian Schwitalla
- 79** Impressum
- 80** Die DOAG bezieht neue Geschäftsräume
Wolfgang Taschner
- 81** Wir begrüßen unsere neuen Mitglieder
- 81** Vorschau
- 82** DOAG Termine



Mit den richtigen Werkzeugen alle Trümpfe in der Hand.

Toad® hat für jede Aufgabe die passende Lösung.

Das Spiel heißt: „Wer liefert den größten wirtschaftlichen Nutzen?“ Die besten Gewinnchancen hat derjenige, der die richtigen Werkzeuge parat hat.

Quest Software weiß, wie es geht. Das Produkt, dem Oracle-Entwickler bereits seit langem vertrauen, bietet jetzt auch Lösungen, die speziell auf Administratoren und Analysten zugeschnitten sind. Ob Sie auf die Schnelle einwandfreien Code erstellen, die Datenbankverwaltung erleichtern oder den Zugriff auf beliebige Datenquellen ermöglichen möchten, Quest sorgt dafür, dass die Karten immer zu Ihren Gunsten gemischt sind.

Immer ein gutes Blatt in der Hand – Toad von Quest Software.

Hier finden Sie heraus, welche Lösung zu Ihnen passt: www.quest.com/aces



Toad World
community. knowledge. power.



©2008 Quest Software, Inc. Alle Rechte vorbehalten. Quest und Quest Software sind Warenzeichen bzw. eingetragene Warenzeichen von Quest Software. Alle sonstigen Marken- oder Produktnamen sind Warenzeichen oder eingetragene Warenzeichen der jeweiligen Rechteinhaber.

Februar 2008

*Fried Saacke,
Vorstandsvorsitzender
der DOAG*

Samstag, 16. Februar 2008

Einmal im Jahr treffe ich alle DOAG-Vorstände und Beiräte zur Beiratssitzung, um neue Ideen zu entwickeln und die Ziele für ein erfolgreiches DOAG-Jahr festzulegen. In den sehr konstruktiven Gesprächen beschließen wir eine neue Strukturierung unseres Themenangebots, um die Oracle-Anwender zielgerichteter ansprechen zu können. Im Kern stehen die drei Bereiche Business, Development sowie Datenbank & Infrastruktur. Übergreifend sind die Ressorts Management und Architektur angeordnet. Außerdem beschließen wir, dass die Anwenderkonferenz unter dem Label DOAG 2008 Konferenz + Ausstellung in diesem Jahr erstmals über einen Zeitraum von drei Tagen stattfindet.

Donnerstag, 21. Februar 2008

Ich besuche die Veranstaltung der SIG Development, die den Oracle Designer zum Thema hat. In einer zuvor durchgeführten Umfrage stellte sich heraus, dass für rund 23 Prozent der teilnehmenden 334 Unternehmen der Oracle Designer ein strategisch wichtiges Tool und für 30 % ein wichtiges Werkzeug in der Software-Entwicklung ist. Von daher kann ich nicht verstehen, dass Oracle ein so beliebtes Tool einfach verkümmern lässt (siehe Seite 77).

März 2008**Montag, 10. März 2008**

Zusammen mit Wolfgang Taschner, Chefredakteur der DOAG News, treffe ich Jochen Boekel, den frischgebackenen Vice President EMEA der Oracle University, zum Interview. In der Zusammenarbeit mit dem Schulungsbereich von Oracle sehe ich noch ein großes Potenzial für eine Zusammenarbeit (siehe Seite 10).

Donnerstag, 13. März 2008

Heute schalten wir unsere Webseiten zum Thema „Oracle-Lizenzierung“ frei. Dieser neue Service soll das für Anwender problematische Thema transparenter machen.

April 2008**Freitag, 4. April 2008**

Im Beisein des DOAG-Vorstands sowie des Berlin-Neuköllner Bezirksbürgermeisters Heinz Buschkowsky eröffne ich die neuen Geschäftsräume der DOAG. Den Mitarbeiterinnen und Mitarbeitern der DOAG-Geschäftsstelle, die von der DOAG Dienstleistungen GmbH betrieben wird, stehen nun insgesamt sieben große und helle Büroräume sowie ein Besprechungszimmer zur Verfügung. Darüber hinaus können wir in der angeschlossenen Konferenzlounge eigene Veranstaltungen durchführen (siehe Seite 80).

Dienstag, 8. April 2008

Auf Veranlassung von Oracle treffe ich mich im EMEA-Headquarter in Genf mit Jean Reicyzk, Senior Vice President EMEA Customer Service; Marcel van de Molen, General Counsel EMEA sowie Dieter Weißhaar, Oracle Vice President Customer Service Northern Europe. Zu meiner Überraschung ist Jean Reicyzk, der für den Oracle Support in ganz Europa verantwortlich ist, sehr gut über die Aktivitäten der DOAG informiert. Wir diskutieren einige wichtige Probleme und finden sehr gute Lösungsansätze, über die wir demnächst berichten werden.

A handwritten signature in black ink that reads "Fried Saacke". The signature is written in a cursive, flowing style.

Die DOAG in der Presse

Die nachfolgenden Ausschnitte reflektieren die Einschätzung der Fach- und Wirtschaftspresse zu bestimmten Themen über Oracle; die Veröffentlichungen geben nicht die Meinung der DOAG wider und sind auch nicht im Vorfeld mit der DOAG abgestimmt. Lediglich die Zitate einzelner DOAG-Vorstände geben die Meinung der DOAG wider.

ix vom 1. Januar 2008: Alles eins

Was als reine Datenbankkonferenz begann, hat sich mittlerweile zur Großveranstaltung für alle Oracle-Produkte entwickelt. In diesem Jahr trafen sich die DOAG-Mitglieder zum 20. Mal, erstmals in Nürnberg.

Noch steht das RDBMS im Mittelpunkt der Deutschen Oracle-Anwendergruppe (DOAG). Andere Produkte aus dem stetig wachsenden Angebot der US-Firma spielen jedoch von Jahr zu Jahr eine größere Rolle. So mussten sich die Teilnehmer diesmal zwischen 200 Vorträgen in 16 parallelen Vortragsreihen entscheiden, die von Neuerungen in Oracle 11g über serviceorientierte Architekturen bis zu PeopleSoft- und Business-Intelligence-Anwendungen reichten.

Im Datenbankteil bildete erwartungsgemäß die neue Version Oracle 11g den Schwerpunkt. Im Vordergrund standen Neuerungen bei Hochverfügbarkeitsfunktionen, im Optimizer, bei der XML-Verarbeitung und der Nutzung von Flashback. Überraschend viel Zulauf hatten Open-Source-Themen. So musste Robert Szilinski seinen Beitrag über den Einsatz von Eclipse, Perl und Subversion ein zweites Mal halten, weil der Raum nicht alle Interessenten hatte fassen können.

Professor Vossen vom ER-CIS (European Research Center for Information Systems) in Münster beschäftigte sich in seiner Keynote mit dem Web 2.0 und seinen möglichen Auswirkungen auf die Softwareindustrie. Dabei blieb er jedoch bei der Beschreibung des Status quo stehen. In weiten Teilen erinnerte der Beitrag an Politikeransprachen zu technischen Themen: Ehrfurchtsvoll der neuen, noch nicht richtig verstandenen Technik gegenüber, die nun aber wirklich endlich den großen Durchbruch bringen soll. Die interessante Frage, wer all die Inhalte in Blogs und Wikis lesen, geschweige denn, pflegen und aktualisieren soll, ließ der Redner folglich unbeantwortet.

Der Veranstalter DOAG zeigte sich ebenso zufrieden mit dem neuen Tagungsort Nürnberg wie viele Teilnehmer. Dass etwas weniger als die erwarteten 2000 Besucher gekommen waren, schrieb er dem drohenden

Lokführerstreik zu. Die Gruppe sieht sich weiterhin als Vertreterin aller Oracle-Kunden, unabhängig vom verwendeten Produkt. Ein Ausdruck dieses Anspruchs sei die Aufnahme eines Vertreters der ehemaligen Siebel-Anwendergruppe in den Vorstand. Mittlerweile hat die DOAG über 3000 Mitglieder und wächst jährlich um rund 10 %.

Als Ergebnis ihrer jährlichen Umfrage konstatierte sie eine gewachsene Zufriedenheit mit dem Oracle-Support. Gegenüber dem Vorjahr hatten mit 502 mehr als doppelt so viele Mitglieder die Fragen beantwortet. Von ihnen waren 48 % „zufrieden“ oder „sehr zufrieden“ – im Vorjahr hatten sich nur 37 % so geäußert. Von der DOAG sehen knapp über 50 % ihre Interessen in diesem Bereich „gut“ oder „sehr gut“ vertreten. Das mag auch daran liegen, dass die Vereinigung inzwischen Mitglied der Internationalen Oracle User Community ist und dadurch direkter Kontakt zum Oracle-Hauptquartier besteht.

Computerwoche vom 21. Dezember 2007:

Oracle lüftet erste Fusion-Geheimnisse. Mit Fusion Applications will der Hersteller zum Marktführer SAP aufschließen. Allerdings bleiben Fragen offen.

Oracle will im ersten Halbjahr 2008 mit ersten Softwaremodulen in sein Fusion-Applications-Zeitalter starten. Den Anfang machen mit „Sales Prospector“, „Sales Campaigns“ und „Sales Library“ drei Tools für den Bereich Sales Force Automation (SFA). Den Anwendern, hier speziell den Vertriebsmitarbeitern, verspricht der Hersteller eine Service-basierende und flexibel zu handhabende Applikationslandschaft. Fusion soll auf einer Service-orientierten Architektur (SOA) sowie einem einheitlichen Datenmodell aufbauen und Oracle zufolge offene Standards unterstützen. Daher lasse sich die Software einfach mit anderen Anwendungen integrieren, sofern diese ebenfalls offenen Standards entsprechen.

Zudem sollen Fusion Applications von Haus aus Business-Intelligence-Funktionen (BI) enthalten und auch in einem Software-as-a-Service-Modell (SaaS) eingesetzt werden können, erläutert Chris Leone, als Group Vice President verantwortlich für Oracles Applikationsstrategie.

Mit Hilfe eines flexiblen User Interface könnten die Anwender rollenbasierte Oberflächen zusammenstellen. Oracles Ziel sei es darüber hinaus, den Implementierungsaufwand möglichst gering zu halten. Auch spätere Upgrades sollen sich mit einer speziellen Stream-Technik ohne längere Systemabschaltungen (Downtime) zügig einspielen lassen. Die Kunden können Schritt für Schritt in die neue Softwarewelt wechseln, verspricht Leone. Außerdem werde kein Benutzer zum Umstieg gezwungen. Jeder Anwender entscheide selbst, wann er auf Fusion wechseln möchte.

Ob Oracle diese Versprechen einlösen kann, bleibe allerdings abzuwarten, bis die ersten Bausteine auf den Markt gelangen und die Fusion-Roadmap ins Rollen kommt, mahnen Anwendervertreter. Bislang habe Oracle nur wenige Einblicke in die Entwicklung der Fusion-Applikationsfamilie gewährt, berichtete zuletzt die Deutsche Oracle Anwendergruppe (DOAG). Fried Saacke, Vorstandsvorsitzender der DOAG, wünscht sich eine stärkere Einbindung der Anwender. Nur so lasse sich gewährleisten, dass die Software auch wirklich für die einzelnen Märkte passt. Saacke spricht aus Erfahrung. Gerade in Deutschland hatte es in der Vergangenheit wiederholt Kritik wegen fehlender lokaler Anpassungen von Oracle-Produkten gegeben.

Doch nach wie vor hüten die Oracle-Verantwortlichen viele Geheimnisse rund um Fusion. Der Hersteller wolle der Konkurrenz nicht zu früh zu viele Informationen preisgeben, hieß es in der Vergangenheit immer wieder. Wie der weitere Fahrplan aussehen wird, vermag Oracles Deutschland-Geschäftsführer Jürgen Kunz noch nicht zu sagen.

Doch so viel scheint festzustehen: Anwender sollen im Rahmen ihres Wartungsvertrags kostenlos auf die neue Fusion-Generation wechseln können, hatten die Oracle-Verantwortlichen in den zurückliegenden Jahren wiederholt versprochen. Damit will der Konzern die Einstiegshürden bewusst niedrig halten. Allerdings dürfte sich das Upgrade-Versprechen lediglich auf bereits genutzte Funktionen beziehen. Wollen die Anwender zusätzliche Fusion-Funktionen implementieren, werden auch neue Lizenzgebühren fällig.

Damit tauchen jedoch weitere Fragezeichen auf. Es bleibt abzuwarten, wie Oracle seine Fusion-Anwendungen paketierte. Noch ist nicht absehbar, ob es bei einer modulartigen Auslieferung bleibt, wie es sich mit den ersten CRM-Werkzeugen aus der Fusion-Familie abzeichnet. Bleibt es dabei, dürfte es für die Anwender

schwierig werden, den Durchblick durch den Lizenz- und Gebührendschungel zu behalten. Bislang hat sich der Hersteller noch nicht zu den Lizenzmetriken für Fusion geäußert.

Bis die Kunden auf Oracles neue Anwendungsgeneration wechseln, werden jedoch noch einige Jahre vergehen. Selbst die Verantwortlichen des Softwarekonzerns vermögen nicht einzuschätzen, wann diese Übergangsphase abgeschlossen sein soll. Um die Migration so reibungslos wie möglich zu gestalten, hat Oracle verschiedene Initiativen gestartet.

So will der Konzern beispielsweise im Rahmen des „Lifetime Supports“ alle bestehenden Softwarelösungen, seien sie selbst entwickelt oder zugekauft, weiterhin mit Support unterstützen. Die Zusage gilt allerdings nur für die aktuellen Produkt-Releases, die zum Zeitpunkt der Ankündigung auf dem Markt sind, schränkt Karsten Roigk, Vice President für den Bereich Applications Strategy bei Oracle in der Region Emea, ein ...

... Um seinen Kunden eine Integration der einzelnen Produktlinien zu bieten, hat Oracle die Application Integration Architecture (AIA) aufgesetzt. Die Verzahnung funktioniert mit Hilfe einer Metaschicht, erläutert Roigk. Dort sind applikationsübergreifend Business-Objekte (Beispiel: Auftrag) sowie Dienste (Beispiel: Anlegen eines Kunden) definiert, die mit diesen Objekten arbeiten. Angesichts der großen Zahl von Objekten und Diensten sei jedoch eine Strukturierung erforderlich, erklärt der Oracle-Manager weiter. Anhand eines Verzeichnisses könnten Anwender Informationen zu einzelnen Versionen sowie über Abhängigkeiten zwischen verschiedenen Release-Ständen der Objekte und Dienste erhalten. Darüber hinaus arbeitet Oracle an Referenzmodellen aus miteinander gekoppelten Objekten und Diensten, die horizontal generische Prozesse und vertikal bestimmte Branchenabläufe abdecken sollen.

Roigk zufolge arbeitet Oracle außerdem mit Hochdruck daran, die Probleme rund um Anpassungen und Lokalisierungen seiner Produkte in den Griff zu bekommen. Dazu beitragen soll ein klar getrenntes Schichtenmodell der Applikationen. Rund um einen Kern mit Standardfunktionen sind in einer zweiten und dritten Schicht regionale (Beispiel: Europa) und länderspezifische (Beispiel: Deutschland) Spezifika definiert. Darüber hinaus gebe es Anpassungen aus dem eigenen Consulting-Bereich sowie von Partnern und Kunden ...

... Ob der Umstieg auf Fusion so zwanglos funktioniert, wie Oracle dies verspricht, bleibt abzuwarten. Mit dem Lifetime Support könnten die Anwender längst nicht alle Probleme lösen, warnt beispielsweise Debra Lilley, Oracle-Expertin von Fujitsu Services und Deputy Chairman der britischen Oracle User Group (UKOUG). Anwenderunternehmen müssten auf Abhängigkeiten zwischen den einzelnen Versionen achten. Beispiels-

weise könnte es in Einzelfällen nicht möglich sein, nur bestimmte Produkte auf einen neueren Stand zu bringen und andere Softwareversionen auf einem älteren Release-Stand zu halten.

Zudem könnte Oracle mit der Verfügbarkeit von bestimmten Funktionen sanften Druck auf die Anwender ausüben, auf die aktuellsten Versionen zu wechseln. Demnach wird beispielsweise die Unterstützung für Single Euro Payments Area (Sepa) erst mit Version 12.1 oder 12.2 der E-Business-Suite folgen. Eine Implementierung in Release 11 sei dagegen nicht vorgesehen.

searchdatacenter.de vom 12. Dezember 2007:

DOAG kritisiert Oracle-Support

Oracle soll am Support nachbessern, so das Credo einer Mitgliedsbefragung der Deutschen Oracle-Anwendergruppe (DOAG). Trotz Fortschritten bei der Kundenbetreuung reagiere der Hersteller mitunter zu träge. Weitere Kritik muss der Softwareanbieter für das weltweite Weiterleiten von Hilfeersuchen einstecken.

Im Vergleich zum Vorjahr haben sich bei der aktuellen DOAG-Umfrage doppelt soviel Firmen beteiligt. Durchschnittlich sei die Zufriedenheit der 500 Teilneh-

mer leicht angestiegen. Dementsprechend versöhnlich gibt sich auch DOAG-Vorsitzender Fried Saacke: „Die gemeinsamen Bemühungen von DOAG und Oracle, den Support zu verbessern, haben sich ausgezahlt [...]“.

Bei komplexen Problemfällen sehen die DOAG-Mitglieder dennoch Nachbesserungsbedarf. „[...] Hier erwarten die Anwender vom Support insbesondere deutschsprachige Ansprechpartner sowie Kenntnisse der lokalen Besonderheiten bei geschäftlichen Abläufen“, so Saacke. Einer der Kritikpunkte: Oracle versuche immer wieder, Telefonanrufer auf den englischsprachigen Dienst Metalink umzuleiten. Zudem führe das weltweite Routen von Supportanfragen immer wieder zu Problemfällen. Auf Kundenwunsch sollte der Support daher in einer Hand und in Deutschland bleiben, fordert die DOAG.

Bei Oracle scheint man sich keiner Schuld bewusst. Selbstverständlich biete man zu den üblichen Geschäftszeiten einen deutschsprachigen Support an. Gemäß dem „Follow-the-Sun-Prinzip“ stünden auch darüber hinaus Ansprechpartner zur Verfügung. Weltweit beschäftigt das Unternehmen derzeit 7.500 Support-Mitarbeiter. Zudem verweist der Hersteller auf die zahlreichen Auszeichnungen, die das Support-Tool Metalink erhalten habe.




www.trivadis.com

Das Unternehmen
Trivadis ist ein erfolgreicher, expandierender und unabhängiger Anbieter von IT Lösungen und Services. Trivadis ist für mehr als 600 Kunden an 12 Standorten in Deutschland, in Österreich und in der Schweiz tätig. Das Dienstleistungsportfolio umfasst Anwendungsentwicklung, Consulting, Systemmanagement, Projektmanagement und Schulung.

Ausbildungsprogramm
Während der Ausbildung durchlaufen die Teilnehmer zahlreiche Stationen in ihrer Heimatniederlassung und in Zürich (Schweiz). Nach erfolgreichem Abschluss der Ausbildung erhalten die Absolventen ein Zeugnis, in dem die erworbenen Qualifikationen auch inhaltlich dokumentiert werden und einen Arbeitsplatz als Junior-Consultant in einer unserer Niederlassungen in Deutschland oder in der Schweiz.

Was wir Ihnen bieten
Trivadis bietet Ihnen ein selbständiges, flexibles Aufgabengebiet, viel Teamgeist, gute Weiterbildungs- und Entwicklungsmöglichkeiten sowie ein Know-how-Netzwerk, auf das Sie sich verlassen können. Attraktive Anstellungsbedingungen, moderne Arbeitsmittel und ein leistungsorientiertes Einkommen runden unser Angebot ab.

Interessiert?
Wir freuen uns auf Ihre Bewerbungsunterlagen inkl. IT-Skill-Liste an jobs@trivadis.com. Telefonische Auskünfte erteilt Ihnen gerne Frau Janine Käser, Human Resources, Tel. +49 69 264 933 031.

Trivadis GmbH · Human Resources · Täfernstr. 5 · CH-5405 Baden-Dättwil

Application Development
Application Performance Management
Business Communication
Business Intelligence
Managed Services
Security
Training

■ Weiterbildung zum Oracle Datenbank Administrator

Trivadis beschäftigt bereits über 500 Mitarbeiter und will weiter wachsen. Zur Verstärkung unseres Teams im Bereich Oracle Datenbanken, bieten wir per 1. Juni 2008 jungen, engagierten und motivierten Persönlichkeiten mit abgeschlossenem Informatik Grundausbildung ein 6-monatiges Weiterbildungsprogramm zum Oracle Datenbank Administrator an.

Wir bieten Ihnen eine kompakte Ausbildung:

- Sie werden von erfahrenen und ausgewiesenen Oracle Spezialisten ausgebildet.
- Sie lernen den Umgang mit Oracle Datenbanken von Grund auf:
 - Installation auf unterschiedlichen Plattformen
 - Administration und Betrieb von fehlertoleranten DB-Architekturen
 - Datenbank Monitoring und Alerting mit unterschiedlichen Hilfsmitteln
 - Sicherstellung und Wiederherstellung (Backup & Recovery)
 - Einführung in die Datenbanksprachen SQL und PL/SQL
 - Tunen von Datenbank Instanzen

Was Sie mitbringen sollten:

- Fundierte Informatik-Ausbildung (IT Lehre, Fachhochschule oder Studium)
- Neugier für neue Technologien und Lernbereitschaft
- Freude und Interesse am Dienstleistungsgeschäft
- Kundenorientierung
- Sehr gute Deutsch- und Englischkenntnisse

trivadis

makes IT easier. ■ ■ ■

Basel · Baden · Bern · Lausanne · Zürich · Düsseldorf · Frankfurt/M. · Freiburg i.Br. · Hamburg · München · Stuttgart · Wien

„Jeden Tag entwickeln wir einen neuen Kurs ...“

Oracle University schult EMEA-weit mehr als 100.000 Kursteilnehmer im Jahr. Mit dem frischgebackenen Vice President EMEA Oracle University, Jochen Boekel, sprachen Fried Saacke, Vorstandsvorsitzender der DOAG, und Wolfgang Taschner, Chefredakteur der DOAG News.

DOAG News: Wie ist Oracle University aufgebaut?

Boekel: Das Ziel von Oracle University ist es, die Kunden hochqualitativ zu schulen. In Deutschland haben wir dafür ein professionelles Team von fest angestellten Mitarbeitern. Hinzu kommen zahlreiche Experten aus der EMEA-Region und den USA, die zu Spezialthemen eingeflogen werden.

DOAG News: Wie ist Oracle University in der Organisation von Oracle positioniert?

Boekel: Wir sind global in der Region EMEA angesiedelt und seit zwei Jahren Teil der Customer Service Organisation. Neben Lizenzvertrieb, Consulting und Support ist Oracle University die vierte Business-Line im Unternehmen.

DOAG News: Was ist Ihre Funktion?

Boekel: Ich bin seit 1. März 2008 als Vicepräsident Oracle University für die gesamte EMEA-Region verantwortlich.

DOAG News: Was sind Ihre Ziele für die kommenden Jahre?

Boekel: An erster Stelle steht für Oracle die Kundenzufriedenheit. Dafür gibt es Kennzahlen, an denen wir uns messen. Da sich das Produkt-Portfolio von Oracle in den vergangenen Jahren vervierfacht hat, ist ein großer Output erforderlich – und wir entwickeln deshalb jeden Tag einen neuen Kurs. Das dritte Ziel ist die Steigerung der internationalen Zusammenarbeit, um vorhandenes Expertenwissen allen Kunden EMEA-weit zur Verfügung zu stellen.

DOAG News: Wie viele Teilnehmer hatten Sie im letzten Jahr?

Boekel: EMEA-weit waren es im letzten Jahr über 100.000 Schulungsteilnehmer. Die durchschnittliche Kursdauer lag zwischen 3 und 4 Tagen.

DOAG News: Decken Sie alle Oracle-Themen ab?



Jochen Boekel, Vice President EMEA Oracle University

Boekel: Wir bieten in Deutschland insgesamt mehrere hundert Standardkurse an und decken damit das gesamte Produkt-Portfolio ab. Davon findet rund die Hälfte regelmäßig statt. Die anderen Kurse werden nur sehr selten angefragt, bei Bedarf finden wir dann EMEA-weit eine Lösung.

DOAG News: Werden alle Kurse auf Deutsch und mit deutschsprachigen Unterlagen abgehalten?

Boekel: Das trifft auf rund 90 Prozent aller durchgeführten Kurse zu. Kurse auf Englisch kommen nur bei sehr speziellen Themen vor.

DOAG News: Welche Themen kommen bei den Teilnehmern besonders gut an, welche sind weniger gefragt?

Boekel: Traditionell kommen unsere Datenbank-Themen sehr gut an. Gleichzeitig wächst das Applications-Umfeld stark an, insbesondere das Siebel-Curriculum ist gefragt.

DOAG News: Wie trägt Oracle University dazu bei, die Applications-Themen in Deutschland besser zu platzieren?

Boekel: Wir schulen die Berater, um das Wissen im Applications-Bereich zu verbreiten.

DOAG News: Wie ist Ihre Strategie bei der Zusammenarbeit mit anderen Schulungspartnern?

Boekel: Wir setzen seit drei Jahren verstärkt auf Schulungspartner, um den Kunden eine optimale geografische Abdeckung zu bieten und in möglichst vielen Städten in Deutschland präsent zu sein. Durch unser Zertifizierungsprogramm stellen wir sicher, dass die Qualität der Kurse bei den Partnern genauso hoch ist wie in unseren eigenen Schulungszentren.

DOAG News: Wie sieht ein typischer Schulungstag aus?

Boekel: Wir bieten unterschiedliche Schulungen an, darunter auch solche, bei denen der Kunde die Inhalte selbst bestimmt. Ein Applications-Thema wird ganz anders angegangen als ein Kurs über Datenbank-Administration. Wichtig ist, dass der Kunde den Stoff versteht und sich dabei wohlfühlt. Generell vermitteln wir erst das notwendige Wissen durch Theorie und anhand von Demonstrationen an praxisorientierten Fallbeispielen. Anschließend führen wir praktische Übungen durch, um das Erlernete zu festigen. Manche Gruppen nutzen auch die Möglichkeit, nach dem Kurs in unseren Räumen noch weiter zu üben.

DOAG News: Wie entstehen die Kursunterlagen?

Boekel: Die Rohfassung unserer Schulungen wird vom Produkt-Development entworfen. Das heißt, die Entwickler erstellen parallel zur Software die Skripte für die Schulungen. Unsere Trainingsteams bereiten diese Unterlagen dann didaktisch auf.

Fotos: Wolfgang Taschner



Schulungsraum in der Münchner Oracle-Zentrale

DOAG News: Gibt es Feedback seitens der Teilnehmer, und trägt dieses auch dazu bei, die Qualität der Oracle-Produkte zu verbessern?

Boekel: Wenn Schulungsteilnehmer über Probleme auf dem Markt berichten, wie beispielsweise im Bereich Lokalisierung, dann geben wir das selbstverständlich an den Support weiter.



Fried Saacke (links) im Gespräch mit Jochen Boekel

DOAG News: Wie wählen Sie Ihre Referenten aus?

Boekel: Mehr als 90 Prozent der Kurse führen wir mit eigenen Mitarbeitern durch. Das sind sowohl feste Referenten, die ausschließlich Schulungen durchführen, als auch Consultants, die das Wissen aus ihrer täglichen Arbeit weitergeben. Der Prozess zur Auswahl von Referenten ist schriftlich definiert und ISO-9001-2000 zertifiziert.

DOAG News: Findet bei den Kursen eine Qualitätssicherung statt?

Boekel: Zum einen bilden wir unsere Referenten regelmäßig fachlich und didaktisch weiter und zum anderen finden sowohl für unsere eigenen Mitarbeiter als auch die externen Referenten „Train the Trainer“-Veranstaltungen statt. Jeder Kurs wird am Ende von den Teilnehmern beurteilt und diese Beurteilungen werden kontinuierlich ausgewertet. Das gesamte Qualitätsmanagementsystem ist nach ISO-9001-2000 dokumentiert und zertifiziert.

DOAG News: Was sind die Stärken Ihrer Kurse, wo sehen Sie noch Schwächen?

Boekel: Der ständige Zuwachs an neuen Produkten, insbesondere im Applications-Bereich, ist natürlich eine große Herausforderung, da wir für alle 45 Akquisi-

Interview

tionen der letzten Jahre die Schulungen parat haben müssen. Auf der anderen Seite sind wir sehr stolz auf unsere Schulungsunterlagen und die sehr guten Kursbewertungen, die wir in allen Bereichen von unseren Schulungsteilnehmern bekommen.

DOAG News: Bieten Sie auch andere Schulungsmedien an?

Boekel: Oracle hat sehr viel in den Aufbau web-basierter Trainings investiert und stellt hier hervorragendes Material zur Verfügung. Dennoch ziehen es die meisten Kunden vor, eine persönliche Schulung zu besuchen.

DOAG News: In welchem Maß werden Ihre Abschlusszertifikate anerkannt?

Boekel: Das Oracle Zertifizierungsprogramm stellt einen internationalen Standard dar, unabhängig davon, in welchem Land der Kurs absolviert wurde. Es ist sicher für eine Bewerbung sehr nützlich.

DOAG News: Muss ein Kunde Voraussetzungen bei bestimmten Kursen erfüllen?

Boekel: Wir geben bei allen Kursen das erforderliche Grundwissen an, prüfen aber bei den Teilnehmern nicht, ob sie es tatsächlich haben. Sollten sich im Kurs Kunden befinden, die sich extrem unter- oder überschätzt haben, finden wir immer eine individuelle Lösung.

DOAG News: Wie finden Sie die DOAG?

Boekel: Ich halte die Plattform, auf der die DOAG den Erfahrungsaustausch und die Wissensvermittlung unter den Oracle-Anwender ermöglicht, für sehr gut. Der Community-Gedanke ist für jeden Kunden wichtig. Deshalb arbeiten wir auch gerne mit der DOAG zusammen.

DOAG News: Welches Potenzial sehen Sie in der Zusammenarbeit mit der DOAG?

Boekel: Wir werden auch künftig für Veranstaltungen der DOAG entsprechende Referenten zur Verfügung stellen. Darüber hinaus sind wir am gebündelten Feedback der Anwender stark interessiert, weil die DOAG in einem guten und direkten Kontakt mit unseren Kunden steht. Auch an den Erfahrungen der langjährigen Oracle-Experten in der DOAG möchten wir gerne teilhaben.

DOAG News: Werden Sie auch in diesem Jahr im Rahmen der DOAG 2008 Konferenz + Ausstellung auf dem abschließenden Schulungstag vertreten sein?

Boekel: Wir werden in jedem Fall als Schulungsanbieter vor Ort sein. Wenn ich es einrichten kann, werde ich auch persönlich anwesend sein, da mir die direkte Diskussion mit den Teilnehmern sehr wichtig ist.

DOAG News: Wie verbringen Sie Ihre Freizeit?

Boekel: Meine Frau und ich kochen gerne und als frischgebackene Eltern verbringen wir natürlich jede freie Minute mit unserer Tochter.

Zur Person: Jochen Boekel

Jochen Boekel trägt als Vice President EMEA Oracle University Umsatzverantwortung für die Aktivitäten der Oracle University in den Regionen Europa, Naher Osten und Afrika. Die Oracle University bietet weltweit ein umfassendes Schulungs- und Zertifizierungsangebot für Produkte und Lösungen von Oracle. In seiner vorherigen Rolle bei Oracle war Jochen Boekel als Director Education für die Education Line of Business in Deutschland, Österreich, der Schweiz sowie in den skandinavischen Ländern zuständig.

Vor seiner Tätigkeit bei Oracle hat Jochen Boekel das Telekommunikationsunternehmen DSLcomp mit Sitz in Düsseldorf gegründet, und als CEO von 2001 bis 2003 geleitet. Davor war er als Director verantwortlich für das 1998 von ihm mitgegründete Unternehmen Novento Telecom AG, ebenfalls mit Sitz in Düsseldorf. Von 1995 bis 1998 war er für den Telekommunikationsanbieter Esprit Telecom PLC mit Sitz in Reading (UK) tätig.

Jochen Boekel hat sein Studium mit dem Master of Arts in Public Administration der Universität Rotterdam und Leiden abgeschlossen. Darüber hinaus erlangte er den International MBA des Instituto de Empresa in Madrid.



Zeit für den persönlichen Austausch

Zwei Themen für große Datenbanken: Partitionierung und NOLOGGING-Operationen

Autor: Dierk Lenz, Herrmann & Lenz Services GmbH

Die Grundregel Nr. 1 beim Arbeiten mit Datenbanken heißt: Die Datenmenge wächst – und zwar in den meisten Fällen stärker als angenommen! Neue Funktionen, historische Daten, Data Warehouses etc. führen dazu, dass auf dem gerade angeschafften und für die nächsten Jahre ausgelegten Storage-Turm häufig nach kurzer Zeit der Platz eng wird. In diesem Artikel sind einige Erfahrungen zusammengefasst, die in verschiedenen Projekten mit großen Datenbanken gemacht wurden.

Wann ist eine Datenbank groß? Vor 15 Jahren war eine relationale Datenbank groß, wenn sie mehr als 10 GB hatte. Heute hat man bereits einige GB zusammen, wenn man nur die Systemstrukturen (Redo-Logs, Undo-Tablespace, Temporär-Tablespace etc.) für eine Datenbank angelegt hat, die nicht unbedingt nur das Beispielschema SCOTT enthält.

Groß ist eine Datenbank dann, wenn bestimmte Operationen (Sicherungen, Index-Rebuilds, Reorganisationen etc.) so lange dauern, dass sie nicht mehr „mal eben“ nebenher eingeplant werden können. Größe kann sich sowohl innerhalb der Datenbank (Tabellen- beziehungsweise Index-Speicherplatz) bemerkbar machen als auch außerhalb – etwa bei der Menge der generierten Archivelog-Informationen. Im zweiten Fall muss die Datenbank selbst dann gar nicht mehr so groß sein. Nach Erfahrung des Autors ist von großen Datenbanken die Rede, wenn man es auf die eine oder andere Art mit mehreren hundert GB zu tun hat.

Zwei Dinge braucht man fast immer ab einer entsprechenden Datenbank-Größe: Partitionierung und NOLOGGING-Operationen. Beide Themen werden im folgenden behandelt.

Partitionierung

Datenbank-Objekte wie Tabellen und Indizes haben die in Grundregel Nr. 1 beschriebene Eigenschaft: Sie wachsen. Betrachtet man die Details der Segment- und Extent-Verwaltung bei Oracle, kann man mit Sicherheit eins sagen: Eine Tabelle oder ein Index wird ohne administrativen Eingriff nie kleiner als aktuell, auch wenn man alle Daten löscht. Fügt man wieder Daten ein, wird der bestehende Platz mehr oder weniger gut wiedergenutzt – meist bei der Tabelle mehr, beim Index weniger ...

Daher sind Tabellen und Indizes – wenn sie einer großen Veränderungsrate unterliegen – von Zeit zu Zeit zu reorganisieren. Dazu existieren diverse, an anderer

Stelle diskutierte Verfahren, die allesamt die Eigenschaft haben, gut, schnell und teilweise sogar online zu funktionieren – solange man es mit einer maximal etwa siebenstelligen Anzahl von Datensätzen zu tun hat. Darüberhinaus kann man sich eine Zeitlang mit besserer Hardware retten, aber irgendwann wird die Zeit zu lang. Spätestens dann gilt es, die Daten zu partitionieren.

Dazu benötigt man die Partitioning-Option (und damit auch eine Enterprise Edition). Damit kann man Tabellen und Indizes nach diversen Methoden partitionieren, das heißt, das logische Objekt in verschiedene Segmente (Partitionen) aufteilen, wobei das Partitionierungskriterium die Zuordnung von Datensatz zu Partition bestimmt. Üblich ist meist die RANGE-Partitionierung, bei der Wertebereiche von Spalten (oft Datumsbereiche) Partitionen zugeordnet werden. Damit erhält man beispielsweise Quartals- oder Monatspartitionen. Diese haben den Vorteil, nach einer gewissen Aufbewahrungsfrist komplett entsorgt werden zu können.

Partitioniert man die Indizes lokal, also mit dem gleichen Kriterium wie die Tabelle, ist das Löschen einer Partition in wenigen Sekunden erledigt. Der DELETE der entsprechenden Datensätze aus einer nicht partitionierten Tabelle dauert hingegen oft stundenlang.

Neben der RANGE-Partitionierung gibt es noch LIST- und HASH-Partitionierung, bei denen Wertelisten beziehungsweise Hash-Funktionen die Partitionierungskriterien sind. Eine Verschachtelung ist ebenfalls möglich, etwa eine RANGE-Partitionierung, deren Partitionen nochmals HASH-partitioniert sind. Mit 11g sind hier alle denkbaren Kombinationen möglich; 10g war noch recht eingeschränkt.

Das für die RANGE-Partitionierung notwendige rechtzeitige Anlegen der Partitionen übernimmt in 11g die INTERVAL-Partitionierung. Wann immer möglich, sollte diese auch eingesetzt werden, da ansonsten die Gefahr besteht, dass man eine Partition überläuft – womit das

Arbeiten mit großen Datenbanken

Problem von zu groß geratenen Segmenten entsteht, das man ja eigentlich mit Partitionierung lösen wollte ...

NOLOGGING-Operationen

Oracle bietet die Möglichkeit, bestimmte Operationen wie das Füllen einer Tabelle unter gewissen Voraussetzungen unter NOLOGGING, das heißt ohne Generierung von Redo-Log-Informationen durchzuführen. Dies kann nützlich sein, aber auch sehr gefährlich!

Wofür sind NOLOGGING-Operationen gut? Fügt man beispielsweise eine große Anzahl von Datensätzen in eine Tabelle ein, ist die Menge der generierten Redo-Log-Information stark davon abhängig, ob bereits Indizes existieren oder nicht. Ist die Tabelle vorher leer (oder fast leer), so lohnt es sich, die INSERTs vorzunehmen, bevor die Indizes aufgebaut werden. Existieren bereits Datensätze, so kann man oft die Indizes für die Befüllung nicht loswerden. In einer solchen Situation ist die Redo-Log-Menge meist riesig, da jeder eingefügte Datensatz im Index-Baum einzeln behandelt werden muss – oft sind für einen Datensatz Änderungen an mehreren Blöcken erforderlich.

Eine Alternative ist eine Direct-Path-Operation (etwa mit dem SQL*Loader oder dem APPEND-Hint beim INSERT), bei der die neuen Datensätze einen eigenen Bereich oberhalb der High-Water-Mark der Tabelle bekommen. Nachdem alle Datensätze eingebunden sind, wird ein eigener Sub-Index für die neuen Datensätze erstellt, der mit dem vorhandenen Index in einer MERGE-Operation zu einem neuen Index verarbeitet wird. Normalerweise wird nun der komplette neue Index in die Redo-Logs geschrieben – NOLOGGING verhindert genau das.

Dies hat natürlich die Konsequenz, dass der Index nach einer Datenbank-Wiederherstellung nicht zur Verfügung steht – es sei denn, man führt nach der Aktion von den betreffenden Daten-Dateien eine neue Sicherung durch. Daraus leitet sich auch die allgemeine Empfehlung ab, sich entweder sicher zu sein, dass man

die mit NOLOGGING bearbeiteten Segmente zügig selbst wiederherstellen kann (etwa mit CREATE INDEX ... NOLOGGING!) oder nach der NOLOGGING-Aktion eine entsprechende Sicherung anstößt. Nachfolgend finden Sie Anmerkungen zu einigen Gerüchten und Fakten zum Thema NOLOGGING:

- *Gerücht 1: Für eine Tabelle, die das Attribut NOLOGGING gesetzt hat, wird keine Redo-Log-Information mehr erzeugt.*

Das ist falsch! Das NOLOGGING-Attribut einer Tabelle beziehungsweise eines Indizes heißt lediglich, dass NOLOGGING-Operationen möglich sind. Dies betrifft keinesfalls das „normale“ DML (INSERT, UPDATE, DELETE), sondern lediglich die genannten Direct-Path-Operationen.

- *Gerücht 2: Für einen Tablespace, der das Attribut NOLOGGING gesetzt hat, wird keine Redo-Log-Information mehr erzeugt.*

Das ist ebenfalls falsch! Das NOLOGGING-Attribut eines Tablespaces dient lediglich als Standardwert für die darin kreierten Tabellen und Indizes (siehe Gerücht 1).

- *Gerücht 3: Operationen wie CREATE TABLE, CREATE INDEX etc., die die Option NOLOGGING enthalten, erzeugen keinerlei Redo-Log-Informationen.*

Auch das ist falsch! Zunächst werden immer Redo-Log-Informationen für interne Operationen wie Space Management erzeugt. Das heißt, der Platz für das Segment ist auf jeden Fall belegt. Aber die Operation kann trotzdem Redo-Log-Informationen erzeugen, wenn die Datenbank im FORCE-LOGGING-Modus ist. Dieser wird zum Beispiel für Data-Guard-Konfigurationen benötigt.

Kontakt:

Dierk Lenz

dierk.lenz@hl-services.de

ORACLE Newsticker

Clusterware für Oracle Unbreakable Linux verfügbar

Seit 1. April 2008 steht Oracle Clusterware für Kunden, die einen Basis- oder Premier-Support-Vertrag für Oracle Unbreakable Linux haben, zum kostenlosen Download bereit. Die neue Clusterware ist eine Komponente von Real Application Clusters – aber auch unabhängig davon verwendbar. Oracle Clusterware ermöglicht Hochverfügbarkeit von Anwendungen und Datenbanken, die in einer Cluster-Umgebung gesteuert werden, wie Oracle Single Instance Databases, Oracle Application Servers, Komponenten des Oracle Enterprise Manager, Datenbanken von Drittherstellern sowie weiteren Anwendungen.

Die Cluster-Lösungen von Oracle für Linux basieren sowohl auf Oracle Clusterware als auch auf Oracle Cluster File System Version 2. Je nach Bedarf können diese entweder zusammen oder einzeln verwendet werden. Seit seiner Markteinführung vor einem Jahr wird Oracle Unbreakable Linux bei mehr als 2.000 Kunden eingesetzt. Mehr Informationen zu Oracle Unbreakable Linux sind unter <http://www.oracle.com/technology/tech/linux/index.html> erhältlich.



Profis im Rechnungswesen

Erfolgreich mit Merck!

Das heißt: Sie beherrschen Ihr Metier. Sie sind besonders stark im Team. Sie sind zuverlässig und setzen sich ein.

Merck ist ein weltweit erfolgreiches Pharma- und Chemieunternehmen und bietet talentierten und unternehmerisch handelnden Mitarbeitern interessante Perspektiven!

Verstärken Sie die buchhalterische Abwicklung bei Merck Shared Services Europe in Darmstadt.

Internationale/r Bilanzbuchhalter/in
Sie übernehmen die komplette Betreuung unserer europäischen Gesellschaften aus buchhalterischer Sicht. Als Basis dient Ihnen ein BWL-Studium oder eine kaufmännische Ausbildung mit entsprechender Weiterbildung.

Buchhalter/in Accounts Payable
Buchhalter/in Accounts Receivable
Sie wickeln komplexe Geschäftsprozesse ab und kümmern sich – je nach fachlicher Spezialisierung – um Fragen aus dem Kreditoren- oder Debitorenkontokorrent (inkl. Kreditmanagement). Eine kaufmännische Ausbildung und spezielle Kenntnisse der Kontierung bringen Sie mit.

Referent/in Ergebnisrechnung
Sie erstellen die monatliche Ergebnisrechnung von Gesellschaften der Merck-Gruppe im europäischen Ausland. Ein BWL-Studium oder eine kaufmännische Ausbildung mit entsprechender Weiterbildung sind dafür Voraussetzung.

Für alle Positionen erforderlich: Buchhal-

terische Gesamtkenntnisse, sicherer Umgang mit Oracle Finance sowie mehrjährige, der Tätigkeit entsprechende Berufserfahrung. Ihr Zahlenverständnis ist ausgeprägt. Sehr gute Sprachkenntnisse in Französisch (Muttersprache wäre ideal) und Englisch sind unbedingt erforderlich. Die Bereitschaft zu Auslandsaufenthalten rundet Ihr Profil ab.

Details hierzu im Internet.

Sie wollen weiter als andere? Dann gestalten Sie die Zukunft mit Merck!

Merck KGaA
HR-Direktberatung
Telefon (06151) 72 33 38

Bitte bewerben Sie sich online unter



Sizing und Architektur einer 30 bis 60 TB großen Datenbank

Autor: Björn Bröhl, OPITZ Consulting GmbH

Hat man es hauptsächlich mit der Administration von bestehenden Datenbanken zu tun, die gegebenenfalls nur migriert werden müssen, macht man sich wenig Gedanken über das grundlegende Sizing einer neuen Datenbank. Um hingegen eine große Datenbank effektiv zu betreiben, sind vorab viele Dinge zu bedenken.

Das Landesamt für Vermessung und Geoinformation des Landes Bayern (LVG) stand vor der Aufgabe, die Architektur und das Sizing für eine hochverfügbare neue Datenbank zu erstellen. Der Umfang der Datenbank sollte mindestens 30 TB betragen. Gleichzeitig sollte das System so ausgelegt werden, dass bei einem Datenvolumen von bis zu 60 TB keine Performance-Probleme zu erwarten sind.

Der Hintergrund: Das LVG in München ist unter anderem dafür zuständig, Karten und Luftbilder des Landes Bayern anzufertigen und zu verwalten. In diesem Zusammenhang gewinnt man zum Beispiel regelmäßig Luftbilder und Laser-Scans für Höhenprofile durch Überflug. Diese Daten werden anschließend bearbeitet und in verschiedenen Auflösungen und Detail-Tiefen

abgelegt. Die Karten sind für verschiedene öffentliche und nicht-öffentliche Anwendungen verfügbar. Ein Beispiel für einen öffentlichen Zugriff ist der Bayern-Viewer unter <http://www.geodaten.bayern.de/Bayern-Viewer>.

Bisher erfolgte die Ablage der Daten wie Karten, Luftbilder, Luftbildkarten, Geländehöhen etc. in verschiedenen Datenbanken, die meist dateibasiert waren. Im Rahmen neuer Services und Vorhaben sollten nun möglichst alle Daten in einer zentralen, hochverfügbaren Datenbank konsolidiert werden. Eine besondere Herausforderung dabei ist, dass viele Informationen noch nicht endgültig verfügbar sind: Zum Zeitpunkt der Planung noch keine Aussagen über das Nutzerverhalten getroffen werden, ebenso wenig über die Anwen-

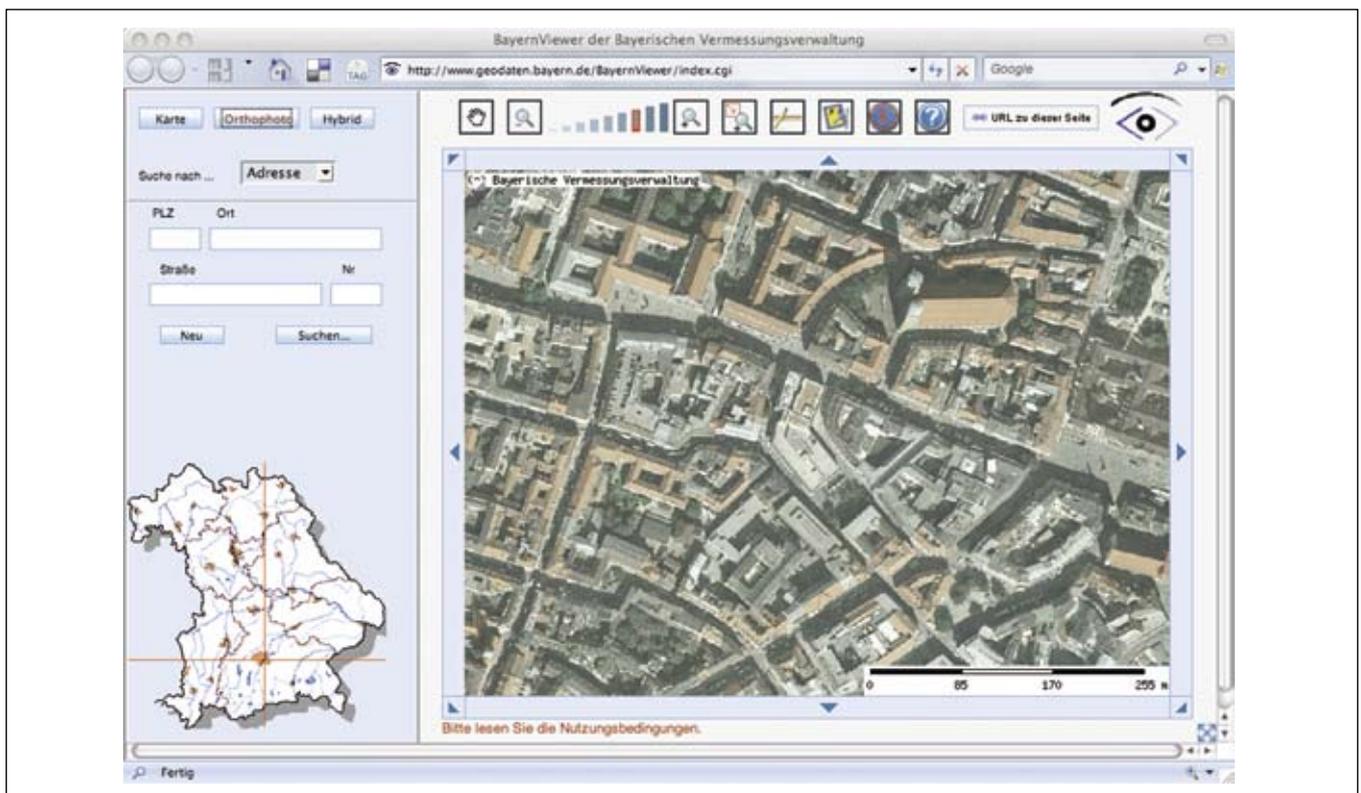


Abbildung 1: Der Bayern-Viewer

dungen, die auf die Daten zugreifen. Auch die genaue Datenmenge war uns zu diesem Zeitpunkt noch nicht bekannt. Man ging zunächst von einem 30 bis 60 TB großen Datenbestand aus und rechnete mit einem jährlichen Wachstum der Daten in Höhe von rund 3 TB.

Die Vorgehensweise im konkreten Projekt

Um trotz fehlender Informationen zu einem validen Sizing und einer funktionalen Architektur zu gelangen, wurden die Projekt-Phasen folgendermaßen aufgeteilt:

1. Sammlung der benötigten Basis-Informationen wie zum Beispiel Quellsysteme, bestehende Datenbanken, Serversysteme, Datenmengen etc.
2. Durchführung eines Workshops zur Planung und Dimensionierung eines Storage-Systems
3. Durchführung eines Workshops zur Bestimmung der zu verwendenden Hochverfügbarkeits-Technologie zum Betrieb der Oracle-Datenbank und des Application Servers
4. Erstellung eines realisierungsfähigen Konzeptes für Storage-System, Datenbank und Application Server

Zur Bewertung der Anforderungen an ein Storage-System wurde ein Storage-Workshop durchgeführt, in dessen Rahmen neben der Präzisierung des Ist- und des Soll-Zustands auch die Darstellung möglicher Storage-Architekturen betrachtet wurde, unter anderem auch hinsichtlich K-Fall-Szenarien und Backup- und Recovery-Konzepten. Diese Architekturen wurden direkt aus dem „Oracle-Blickwinkel“ bezüglich möglicher Zertifizierungen und bestehender Referenzsysteme bewertet. Für die Realisierung beim LVG wurde ein System empfohlen, das auf kostengünstigen SAN- und bereits vorhandenen NAS-Systemen basiert, sowie auf einer Storage-Virtualisierung auf Hardwareebene (Appliance). Die Verwendung einer Storage-Virtualisierung deckt die Hauptanforderungen hinsichtlich Verfügbarkeit und Erweiterung des Systems ab.

Im Anschluss an den Storage-Workshop fand ein Oracle HA-Workshop statt. Hier betrachtete man verschiedene Hochverfügbarkeitstechnologien (von Oracle und anderen Herstellern) und bewertete diese im Rahmen der Anforderungen. Im Anschluss wurde ein Gesamtkonzept erarbeitet und abgestimmt – bestehend aus Hardware- und Software-Empfehlungen sowie wichtigen Hinweisen für die Vorgehensweise bei der Umsetzung.

Als optimale Architektur, die alle gestellten Anforderungen erfüllen konnte, wurde ein Oracle Real Application Cluster ausgewählt. Dieses bestand aus drei Knoten mit SuSE Linux SLES10 64 Bit. Außerdem be-

nötigten wir einen Oracle Application Server Cluster, ebenfalls mit drei Knoten.

Allgemeine Vorgehensweise

Neben den beschriebenen Aspekten dieses konkreten Projekts soll der Beitrag auf die grundsätzliche Vorgehensweise für das Sizing eines neuen, komplexen Systems eingehen. Folgende Fragestellungen können Anhaltspunkte für das weitere Vorgehen liefern (Bottom-Up-Methode):

- *Was soll auf dem neuen System alles betrieben werden (Oracle Datenbank, Oracle Application Server ...)?*
Definition der grundlegenden Anforderungen
- *Soll ein bestehendes System migriert werden?*
Dann existieren möglicherweise Erfahrungswerte (Kennzahlen)
- *Müssen Hochverfügbarkeit und Sicherheit berücksichtigt werden?*
Neu-Definition der grundlegenden Anforderungen

Hat man diese erste Hürde genommen, nähert man sich dem Problem des richtigen Sizings mit folgenden Aktivitäten:

- *Definition der Anforderungen*
Ziele festlegen wie einzusetzende Versionen, Verfügbarkeit, Security etc.
- *Bestimmung der Architektur*
Client/Server, Web, Cluster etc.
- *Berechnung der Kennzahlen*
Wie viele Anwender arbeiten mit dem System?
- *Erstellung eines Architekturentwurfs*
grobes Design erstellen
- *POC / Benchmark zur Überprüfung*
Verfeinerung der Architektur

Im nächsten Schritt können die so gewonnen Informationen zur Festlegung oder Klärung weiterer Eckpunkte verwendet werden. Folgende Punkte sind hier exemplarisch erwähnt:

- Auswahl des zu verwendenden Betriebssystems unter Berücksichtigung daraus resultierender Einschränkungen wie zum Beispiel existierende Zertifizierungen etc.
- Auswahl der Umgebung (32 oder 64 Bit)
- Auswahl möglicher Anbieter von Hard- und Software

Technische Aspekte des Sizings

Aus technischer Sicht besteht meist das größte Interesse an der Bestimmung der Kennzahlen. Diese können sein:

Arbeiten mit großen Datenbanken

- CPUs (Taktfrequenz und Anzahl)
- Speicherbedarf im Arbeitsspeicher
- Speicherbedarf auf Festplatte
- O-Transaktionen
- Netzwerk-Traffic

Mögliche Kennzahlen lassen sich beispielsweise durch Betrachtung und Analyse eines bestehenden Systems ermitteln. Hierzu existieren eine Menge unterschiedlicher Tools.

The screenshot shows a tool interface with three main sections: 'I/O der Datenbank', 'I/O des Filesystems', and 'I/O der Tablespaces'. Each section contains a table with performance metrics.

Instanzname	Size (MB)	No. Blocks Read (1000 #)	No. Blocks Written (1000 #)	Time to Read 1 Block (ms)	Time to Write 1 Block (ms)	Average Time to Read/Write 1 Block (ms/Block)	Average Speed to Read/Write 1 Second (KB/s/Sec)
APRZ	1018,17	18,97	-73	61,60	60,14	61,87	16,70

File System Name	Size (MB)	Number Blocks Read (1000 #)	Number Blocks Written (1000 #)	Time to Read 1 Block (ms)	Time to Write 1 Block (ms)	Average Time to Read/Write 1 Block (ms/Block)	Average Speed to Read/Write 1 Second (KB/s/Sec)
C:\ORACLE\PRODUCT11.1.0\ORADATA\APRZ	1018,17	18,97	-73	61,60	60,14	61,87	16,70

Tablespace Name	Size (MB)	Number Blocks Read (1000 #)	Number Blocks Written (1000 #)	Time to Read 1 Block (ms)	Time to Write 1 Block (ms)	Average Time to Read/Write 1 Block (ms/Block)	Average Speed to Read/Write 1 Second (KB/s/Sec)
SYSTEM	240	-13	1000,00	66,70	61,87	64,28	15,53
SYSAUX	240	6,74	40	62,41	60,14	61,27	16,29
SYSTEM	240	7,84	17	61,67	60,14	60,90	16,24
APRZ	1018	1,74	0	6,00	6,67	6,33	158,00
PLUG_2	1,04	-13	0	6,0	6,33	6,17	158,67
PLUG_1	10,04	-14	0	10,73	6,67	8,70	16,60
ORDD	0	-13	0	0	6,67	6,67	16,14

Abbildung 2: Ein Tool von Opitz Consulting zur Analyse bestehender Oracle-Datenbanken

Gibt es noch kein bestehendes System, das analysiert werden kann, greift man am besten auf Erfahrungswerte für vergleichbare Systeme zurück. Wichtige Hinweise dazu können folgende Quellen liefern: Anbieter einer Standardsoftware, Installation-Guides, Whitepapers, Mitarbeiter mit Erfahrungen aus anderen Projekten etc. Für erste Ideen kann in einfacheren Fällen der kostenlose Service www.sizinglounge.com hilfreich sein.

Hat man die Schritte bis zu diesem Punkt gemeistert, können an dieser Stelle bereits folgende (Zwischen-)Ergebnisse präsentiert werden:

- Übersicht der Anforderungen (Lastenheft)
- (Grob-)Entwurf einer Architektur
- Rahmenparameter für das Sizing

Bevor es an die Bestellung der benötigten Komponenten geht, empfiehlt es sich – sofern die Zeit und das Projektbudget dies ermöglichen –, einen Proof-of-Concept (POC) oder einem Lasttest durchzuführen. Erst danach kann man sich sicher sein, dass die ausgewählten Komponenten und Architekturen den Anforderungen entsprechen. Diese zusätzliche Überprüfung wird auch Right-Sizing genannt, was bedeutet, dass man weder zu viele noch zu wenige, sondern eben genau die richtige Anzahl an Ressourcen eingeplant hat.

LVG ist auf dem richtigen Weg

Zurück zum Projekt: Aktuell befindet sich die LVG in der Planung und Umsetzung einer Vorstufe. In dieser soll mit einer kleineren Ausbaustufe der geplanten Architektur gestartet werden. Dies ist ein möglicher Kompromiss zu einem POC oder Benchmark, der es jedoch nicht ermöglicht, die endgültige Ausbaustufe unter voller Last zu testen. Vielmehr nähert man sich in Etappen der endgültigen Architektur. Voraussetzung hierfür ist allerdings, dass das geplante System skalierbar ist.

Fazit

Planung, Auswahl von Komponenten, das Sizing und auch die Durchführung von Benchmarks oder POC kosten Zeit und Geld. Befindet man sich in einer Projekt-Situation, in der extrem kurz definierte Realisierungszeiten eingehalten werden müssen, hat man oft keine andere Wahl als eine Architektur oder das Sizing „auf einem Blatt Papier“ unter Verwendung von Erfahrungswerten oder aus einem „Bauchgefühl“ heraus zu erstellen. Meist rächt sich dies aber im späteren Betrieb – und zwar spätestens, wenn man erkennen muss, dass die Leistung eines Systems nicht ausreichend ist. Daher empfiehlt es sich fast immer (auch aus Kostengründen), im Vorfeld ein wenig Zeit in Überlegungen zu investieren, um später einen reibungslosen Betrieb oder ein skalierendes System zu ermöglichen.

Kontakt:

Björn Bröhl
bjoern.broehl@opitz-consulting.de

SIG Development

Die DOAG veranstaltet am 5. Juni 2008 in der DOAG Konferenz-Lounge in Berlin und am 19. Juni 2008, im Lindner Congress Hotel in Düsseldorf die SIG Development mit dem Thema „Data Services mit Oracle: SDO, TopLink, Coherence, ADF Data Binding, ...“

Weitere Infos und Anmeldung unter <http://www.doag.org/sig/development/>

Komprimierung in der Oracle Datenbank

Autoren: Ulrike Schwinn und Thomas Niewel, ORACLE Deutschland GmbH

Durch eine Vielzahl neuer Anwendungen nimmt die Menge an gespeicherten Informationen rapide zu. Die Reduzierung des zu speichernden Datenvolumens mittels effizienter Datenkomprimierungsmechanismen bewirkt dabei nicht nur eine signifikante Einsparung des Plattenspeichers und des damit einhergehenden Energieverbrauchs, sondern auch eine effizientere Memory-Nutzung und eine Erhöhung der Performance.

Der wachsende Austausch von immer größer werdenden Datenmengen, wie beispielsweise Multimedia-Daten, über das Internet in Verbindung mit der Weiterentwicklung von Breitbandtechniken verstärkt diesen Trend zusehends. Noch vor wenigen Jahren hatten die größten bekannten Datenbanken ein Volumen von einigen GB, im Jahr 2008 ist das Volumen dieser Datenbanken im TB-Bereich angelangt.

Schon seit einigen Jahren ist deshalb die Komprimierung ein wichtiger Bestandteil der Oracle Datenbank und wird beständig weiterentwickelt. Dies zeigt sich besonders im neuesten Release 11g mit Einführung der Option Advanced Compression. Diese ermöglicht jetzt die Komprimierung unabhängig vom Anwendungs-Workload durchzuführen. Sie ist zusätzlich um die Bereiche unstrukturierte Daten, Backup-Daten und Netzwerk-Komprimierung für Data Guard Installationen erweitert. Dieser Artikel gibt einen Überblick über alle zur Verfügung stehenden Komprimierungsverfahren für Tabellen und Indizes und bereichert ihre Handhabung mit Tipps und Tricks.

Komprimierung von Tabellen mit strukturierten Daten

Bereits mit Version 9.2 war es möglich, relationale Tabellendaten zu komprimieren. Dabei werden Mehrfach-Einträge im Datenblock nur einmal gespeichert. Die sich wiederholenden Werte werden in einer Symbol Table auf Blockebene gespeichert und durch einen Pointer im Datenteil des Blocks adressiert. Dieser Komprimierungsalgorithmus ist sehr effizient, weil die dafür benötigte CPU-Last sehr gering ist.

Durch die Tabellenkomprimierung (Table Compression) kann eine größere Anzahl Zeilen in einem Block gespeichert werden, daher werden weniger „Buffer Get“-Operationen beziehungsweise I/Os durchgeführt. Diese Tatsache führt dazu, dass sich nicht nur der Speicherplatzbedarf auf der Platte und im Memory verringert, sondern sich auch die Performance der Tabellenzugriffe erhöht. Darüber hinaus bestehen keine funktionalen Einschränkungen bei Nutzung von Datenbank-Funktionalitäten wie Oracle Streams Replication.

Seit Version 9.2 kann Table Compression speziell im Data-Warehouse-Bereich für Bulk-Load-Operationen genutzt werden. Dazu gehören:

- Direct Path Load
- CREATE TABLE AS SELECT
- Paralleler INSERT oder serieller INSERT mit APPEND-Hint

Mit Oracle 11g steht diese Funktionalität mit neuer beziehungsweise erweiterter Syntax weiterhin zur Verfügung (siehe Listing 1).

```
CREATE TABLE sales_history(...) COMPRESS
FOR DIRECT_LOAD OPERATIONS
```

Listing 1: 11g-Syntax zur Einstellung der Tabellenkomprimierung

Die Schlüsselwörter COMPRESS FOR DIRECT_LOAD OPERATIONS schalten die 9.2.-Komprimierung für Bulk-Load-Operationen ein.

In Oracle 11g werden mit der Option Advanced Compression alle DML-Operationen unterstützt (siehe Listing 2). Das bedeutet, dass die Einschränkung auf Bulk-Load-Operationen wegfällt, und Komprimierung ohne Rücksicht auf die Art der Ladevorgänge garantiert werden kann.

```
CREATE TABLE sales_history(...) COMPRESS
FOR ALL OPERATIONS
```

Listing 2: 11g-Syntax zur Einstellung der Tabellenkomprimierung

Die Blöcke werden bei dieser Komprimierungsform im Batch-Mode komprimiert und nicht bei jeder Schreiboperation. Dies führt dazu, dass ein neuer Block so lange unkomprimiert bleibt, bis die Daten die PCTFREE-Grenze erreicht haben (siehe Abbildung 1).

Arbeiten mit großen Datenbanken

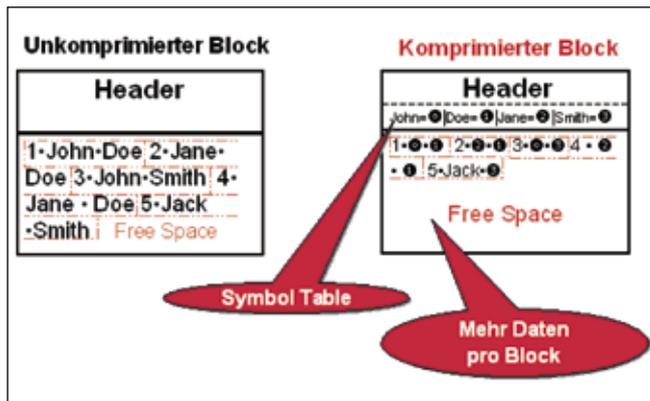


Abbildung 1: Komprimierter Block nach Erreichen der PCTFREE-Grenze

Generell wird Table Compression entweder auf der Tablespace-, Tabellen- oder Partitions-Ebene mit dem entsprechendem CREATE-Kommando eingeschaltet. Auf der Tabellen- beziehungsweise Partitionsebene werden dann die neuen Tabellenblöcke ohne zusätzlichen Reorganisationsaufwand beziehungsweise zusätzliche Operationen komprimiert abgespeichert. Möchte man hingegen die Inhalte von existierenden unkomprimierten Tabellen in komprimierte Tabellen umwandeln, lassen sich die Daten mit dem Kommando ALTER TABLE MOVE COMPRESS FOR beziehungsweise ALTER TABLE MOVE PARTITION COMPRESS FOR in einer Einschnitt-Operation umschichten und gleichzeitig komprimieren. Falls dies online ohne Sperren erfolgen soll, kann man mit dem Package DBMS_REDEFINITION arbeiten.

Je nach Art der Daten, wie der Anteil an redundanten Informationen, Datenbank-Blockgröße und Ladevorgang, können die Komprimierungsraten stark variieren. So ist es vorteilhaft, mit einer großen Datenbank-Blockgröße beziehungsweise mit vorsortierten Daten zu arbeiten, um die Komprimierungsrate zu erhöhen. Die Praxis zeigt, dass im Data-Warehouse-Bereich bei Tabellen mit hohen Redundanzwerten achtfache Komprimierungsraten erreicht werden können. Das bedeutet, dass die unkomprimierte Tabelle achtmal größer als die komprimierte Tabelle ist. Im letzten Whitepaper der Winter Corporation (siehe „Oracle’s Top Ten Features For Large Scale Data Warehousing“ vom März 2007) wird Table Compression zu den TOP-10-Features gezählt und als „essential feature for a data warehousing platform“ bezeichnet. Dort wird sogar von einer Beispielapplikation berichtet, die eine zwölfwache Komprimierungsrate erreichte.

Komprimierung von Indizes

Die Index Key Compression ist bereits ab Oracle 8i verfügbar und steht für B*tree Indizes und Index Organized

Tables (IOT) zur Verfügung. Das Prinzip beruht auf der Eliminierung von sich wiederholenden Schlüsselwerten (Präfixe) eines „nonunique single column“-Index beziehungsweise eines „multicolumn“-Index. Zusammengesetzte Schlüssel in einem Index werden dabei in einen Präfix- und einen Suffix-Anteil unterteilt, wobei der Suffix-Anteil den eindeutigen Teil des Index-Schlüssels repräsentiert. Wenn sich Schlüsselwerte im Präfix-Teil des Indexes wiederholen, so werden diese Werte nur einmal gespeichert und vom Suffix referenziert. Präfix und Suffix befinden sich dabei grundsätzlich im gleichen Datenblock. Dies bewirkt eine Reduzierung der Index Leaf Pages und damit der Anzahl der I/O-Operationen bei einem Index-Zugriff. Die Komprimierung kann beim Erzeugen des Index oder mit einem ALTER INDEX REBUILD-Kommando eingestellt werden (siehe Listing 3). Die Option COMPRESS gibt dabei die Anzahl der Präfixspalten an, die komprimiert werden.

```
CREATE INDEX t1_ci_1 ON t1(col1, col2, col3, col4) COMPRESS 2;
ALTER INDEX t1_ci_1 REBUILD COMPRESS 2;
```

Listing 3: Erzeugen eines komprimierten Index mit Präfixspalten col1 und col2

Die Höhe der Komprimierungsrate des Index kann stark variieren und ist unter anderem abhängig von der richtigen Anzahl der komprimierten Spalten, den Werte-Ausprägungen sowie der Anordnung der Spalten im Index. Falls das Umsortieren der Spalten im Index möglich ist, kann dies zu höheren Komprimierungsraten führen. Um zu beurteilen, ob und welche Indexkomprimierung für die entsprechenden Indexkandidaten sinnvoll ist, eignet sich folgendes Vorgehen:

- Analysieren des Index mit ANALYZE INDEX <indexname> VALIDATE STRUCTURE
- Verifizieren der Spalten OPT_CMPT_COUNT für die optimale COMPRESS-Einstellung und OPT_CMPR_PCT_SAVE für die prozentuale Platzeinsparung aus der Tabelle INDEX_STATS

Listing 4 zeigt ein Ergebnis aus der INDEX_STATS Tabelle. Bei der Einstellung COMPRESS 3 kann eine Einsparung von 55 Prozent für den Index erzielt werden.

```
SELECT opt_compr_count, opt_compr_pct-  
save FROM index_stats;  
OPT_CMPR_COUNT OPT_CMPR_PCTSAVE  
-----  
3 55
```

Listing 4: Ein Ergebnis aus der INDEX_STATS-Tabelle

Komprimierung im Index findet auch in den Bitmap-Indizes statt, die speziell in Warehouse-Anwendungen bei Indizes mit geringer Kardinalität zum Einsatz kommen. Bitmap-Indizes speichern eine Bitmap für jeden Index-Schlüssel statt einer Liste von ROWIDS. Jedes Bit in dieser Bitmap adressiert eine ROWID und somit eine Zeile in der Tabelle. Bitmap-Indizes benötigen keine Komprimierungsalgorithmen, da sie einen sehr geringen Speicherbedarf haben. Je nach Art der Anwendung, insbesondere bei lesenden Zugriffen, ist die Verwendung von Bitmap-Indizes von Vorteil.

Komprimierung von Tabellen mit unstrukturierten Daten

In den vergangenen Jahren hat die Verbreitung von unstrukturierten Daten massiv zugenommen. Unstrukturierte

rierte Daten vom Datentyp XML, CLOB und BLOB sind in der Regel sehr speicherintensiv. Aus diesem Grunde ist hier eine Komprimierung sehr sinnvoll. Mit 11g wurden Oracle SecureFiles als eine optimierte Speicherform dafür eingeführt. Eine Eigenschaft ist die Möglichkeit der Komprimierung von unstrukturierten Daten, die in der Advanced-Compression-Option enthalten ist. Dabei sind folgende Einstellungen der Komprimierung möglich:

- **DEDUPLICATE:**
LOBs mit identischem Inhalt werden physikalisch nur einmalig gespeichert. Diese Einstellung ist sinnvoll bei der Nutzung von großen LOBs, die mit `DISABLE STORAGE IN ROW` abgespeichert werden
- **COMPRESS HIGH** (beziehungsweise **MEDIUM**)
Reduzierung des Speicherbedarfs von LOBs durch Komprimierung. Diese Komprimierung wird durch einen Standardalgorithmus durchgeführt und kann wahlweise mit einer hohen beziehungsweise mittleren Komprimierungsrate durchgeführt werden. Die LOB-Komprimierung mit Parametereinstellung **HIGH** hat dabei einen höheren CPU-Bedarf als die Komprimierung der LOBs mit der SecureFile Standardkomprimierung **MEDIUM**



Und was kann Ihre Datenbank?

Eine niedliche, kleine Datenbank ist eine schöne Sache. Doch im Lauf der Zeit wird das Ding groß und hungrig, es frisst immer mehr von Ihrer Zeit (und wenn Sie nicht aufpassen, noch mehr). Deshalb sollten Sie bei Oracle Datenbanken jeder Größe von Anfang an auf Nummer Sicher gehen. Mit einem Tool, das Tausenden von Nutzern in puncto Bedienungsfreundlichkeit, Durchstrukturierung und Übersichtlichkeit zu einer unschätzbaren Hilfe geworden ist – und jetzt in der ausgereiften, noch griffigeren Version 7.3 vorliegt. KeepTool 7.3 – und das Biest kann wachsen!

Hora: Datenbank Administration und Entwicklung
ER Diagrammer: Datenbank Design
PL/SQL Debugger: Programm Test

Weitere Infos: www.keeptool.com

keeptool
Tools für Oracle Datenbanken

“Hora should be on your short list of tools to consider. It fulfills the needs of DBAs, developers, data operators, report writers, and power browsers, all at a compelling price point.”

Bill Coulam, OraFaq

Arbeiten mit großen Datenbanken

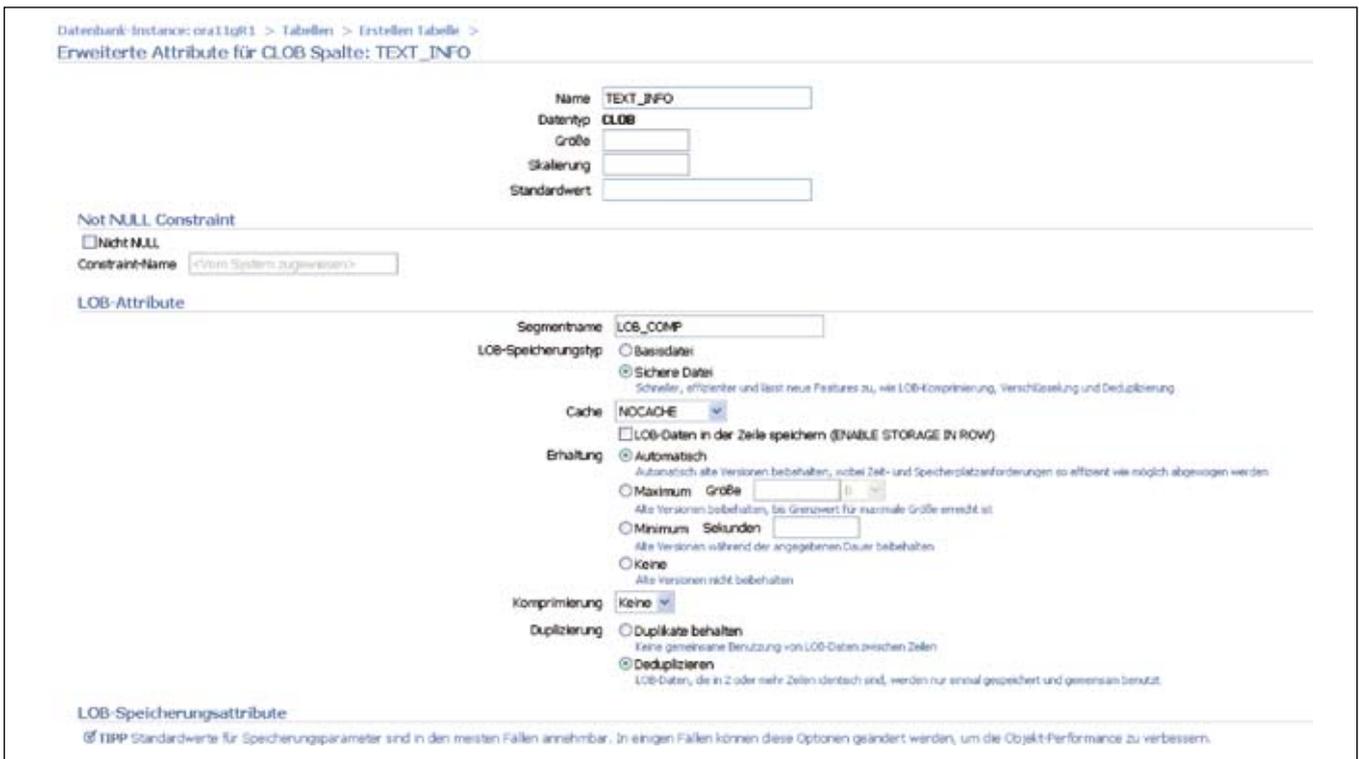


Abbildung 2: SecureFile mit den Einstellungen DEDUPLICATE und COMPRESS HIGH

Die LOB-Komprimierung ist dabei unabhängig von der Tabellenkomprimierung und wird beim `CREATE TABLE` oder `ALTER TABLE` separat über die SecureFile-LOB-Storage-Klausel angegeben (siehe Listing 5). Der Einsatz der Komprimierungseigenschaften ist auch ohne Einschränkungen mit anderen Oracle Technologien wie Oracle Text kombinierbar.

```
CREATE TABLE nachrichten_text (dok_id
NUMBER, ...text_info CLOB)
LOB (text_info) STORE AS SECUREFILE lob_
comp
(INDEX i_lob_comp DEDUPLICATE COMPRESS
HIGH
DISABLE STORAGE IN ROW)
```

Listing 5: LOB-Storage-Klausel für DEDUPLICATE und COMPRESS HIGH

Nur SecureFiles – die sogenannten 10g LOB Speicherformate – und keine BasicFiles können mit den oben erläuterten Komprimierungsklauseln versehen werden. Eine Migration von BasicFile-LOBs lässt sich dabei online mit dem Package `DBMS_REDEFINITION` durchführen.

Alle Einstellungen für Tabellen-, Index- beziehungsweise LOB-Komprimierung lassen sich auch leicht mit der Enterprise Manager Console erzeugen. Der Screen-

shot in Abbildung 2 zeigt die Nutzung an einem Beispiel.

Fazit

Die Komprimierung auf Tabellenebene ist seit geraumer Zeit eines der Key Features für den Data-Warehouse-Bereich. Unabhängig von der Anwendungsart sollten darüber hinaus die Möglichkeiten der Index-Komprimierung nicht vergessen werden. Mit der Advanced-Compression-Option in 11g eröffnen sich zusätzlich neue Möglichkeiten der Komprimierung. Unabhängig vom Applikations-Workload kann die Komprimierung garantiert werden. Das bedeutet, dass nicht nur Warehouse-Anwendungen, sondern auch Anwendungen mit gemischten Workloads oder sogar OLTP-Anwendungen die Datenbank-Komprimierung nutzen können. Darüber hinaus bietet die neue Unterstützung von unstrukturierten Daten über die sogenannte SecureFiles-Komprimierung auch Anwendungen, wie typische Content-Management- oder XML- Anwendungen, die bisher nicht von der Komprimierung profitieren konnten, neue Einsatzmöglichkeiten.

Kontakte:

Ulrike Schwinn
ulrike.schwinn@oracle.com
Thomas Niewel
thomas.niewel@oracle.com

Leitlinien zur Entwicklung und zum Betrieb eines Multi-Terabyte-Data-Warehouses

Autor: Dr. Kurt Franke, debitel AG

In einer großen Datenbank gibt es einige Punkte zu beachten, die bei kleineren Datenmengen kaum erwähnenswert sind. Dieser Artikel stellt einige Möglichkeiten vor, die sich im langjährigem Betrieb und in der Weiterentwicklung eines Data Warehouses bewährt haben, das inzwischen – mit massiv eingesetzter Daten- und Index-Komprimierung – eine Größe von nahezu 7 TB erreicht hat.

Bei einer großen Anzahl von Tabellen kann die Übersichtlichkeit durch eine geeignete Strukturierung erhöht werden. Gerade in einem Data Warehouse sind hohe I/O-Lasten sowohl bei der Integration neuer Daten als auch bei den Zugriffen zu Auswertungszwecken vorhanden. Standard-Mechanismen wie Anlegen zusätzlicher geeigneter Indizes oder Erhöhung und Optimierung des Cache-Buffers bewirken allenfalls geringe Verbesserungen, weil im Regelfall die gesamte Datenmenge nach einem bestimmten Kriterium (meist eine Zeiteinheit) betrachtet wird. Dies verlangt angepasste Vorgehensweisen sowohl bei der Definition der Strukturen als auch bei den Methoden, die zur Bereitstellung der Daten genutzt werden.

Die Vielzahl komplexer Abhängigkeiten der einzelnen Verarbeitungen voneinander erfordert eine geeignete Steuerung zusammen mit den dazu notwendigen Metadaten. Letztendlich darf nicht unbeachtet bleiben, dass in einem Data Warehouse Fachbereichsbenutzer Auswertungen laufen lassen, während parallel Verbreitungen ablaufen können oder müssen und eventuell auch noch administrative Tätigkeiten stattfinden. Durch eine geeignete Entkopplung und Benutzer-Limitierung sollte vermieden werden, dass beispielsweise durch ein fehlerhaftes manuell erstelltes SELECT-Statement eine gleichzeitig laufende Verarbeitung wegen Ressourcen-Mangel wesentlich verzögert wird oder sogar abbricht.

Strukturierte Anordnung der Datenbank-Objekte

Bei Computern ist man es gewohnt, Dateien in einen System aus Foldern und Unterfoldern strukturiert abzulegen, um damit eine gewisse Übersichtlichkeit auch bei einer großen Anzahl von Dateien zu bewahren. Das ist so in einer Datenbank nicht möglich. Alternativ können jedoch verschiedene Schemata quasi als Folderersatz dienen, wobei allerdings eine Limitierung auf eine Strukturierungsebene gegeben ist. In einem Data Warehouse lassen sich sofort einige solche Unterbereiche nennen: Basis-Daten, Analyse-Daten, Referenz-

Daten, Aggregationen, Staging-Bereich, Datenpflegebereich etc. – insbesondere auch ein separates Schema für Arbeitstabellen. Eine Aufteilung in entsprechende Schemata erhöht damit die Übersichtlichkeit und verringert für Wartung aufzuwendende Zeiten sowie auch Einarbeitungszeiten für Entwickler und Administratoren. Eventuelle zusätzliche Aufwände werden durch wenige kleine Utilities vermieden.

In einer Tabelle erfasst man alle zum Zentralbereich DWH_NAMECOMMUNITY gehörigen Schemata. Ein „BEFORE CREATE“-Trigger auf Datenbank-Ebene verhindert, dass gleichnamige Tabellen (nicht Views) in der Name-Community angelegt werden können. Kurzperiodische Jobs (in Abständen von wenigen Minuten) erteilen direkte, noch nicht vorhandene SELECT-Privilegien an die anderen Mitglieder der Name-Community und erzeugen private Synonyme in den jeweils anderen Schemata. Privilegien für schreibende Zugriffe werden grundsätzlich nur nach Bedarf explizit zugeteilt. Dies bringt den zusätzlichen Vorteil, dass Auswirkungen eventueller Programmierfehler grundsätzlich auf einen Teilbereich begrenzt sind.

Verwendung von Tablespaces

„LOCALLY MANAGED“-Tablespaces sind inzwischen Standard – es gibt sie seit Version 8i. Durch die verwendeten Bitmaps statt der vorher üblichen Dictionary-Einträge ist die Extent-Verwaltung so performant geworden, dass man sich über Anzahlen von Extents keine Gedanken machen muss. Das erlaubt den breiten Einsatz der Tablespace-Variante mit „UNIFORM SIZE“-Extents. Die Größe sollte mindestens so sein, dass die maximale I/O-Call-Datenmenge erreicht wird (siehe Abschnitt „Strategien zum Umgang mit großen I/O-Mengen“), um unnötige I/O-Calls zu vermeiden, weil diese Extent-übergreifend nicht möglich sind.

Andererseits ist es auch nicht notwendig, wesentlich größere Extents zur Reduzierung deren Anzahl zu verwenden. Man kann also die Extent-Größe für Daten-

segmente größer als die maximale I/O-Call-Datenmenge so wählen, dass auch möglichst wenig ungenutzter Platz verschwendet wird. Dies gilt insbesondere, wenn für alle große Tabellen jeweils ein eigener Tablespace eingesetzt wird. Dabei lässt sich auch gezielt die maximal mögliche Datenblock-Größe von 32 kB verwenden (seit Oracle 9) – eine Verbesserung beim Einsatz von Daten-Komprimierung (siehe Abschnitt „Komprimierung“). Einfache Verschiebungen von Objekten mit TABLE-MOVE beziehungsweise INDEX-REBUILD werden zwischen Tablespaces mit unterschiedlicher Blockgröße jedoch nicht unterstützt, sodass bei eventuellen Reorganisationen mit Änderungen der Blockgröße die Objekte explizit neu erzeugt werden müssen.

Für die Speicherung kleinerer Segmente, oder wenn immer neue noch nicht bekannter Größe darin abgelegt werden sollen, etwa in speziellen Schemata für Fachbereichs-User, empfiehlt sich hingegen die Verwendung von „SYSTEM MANAGED“-Extents. Man sollte dabei aber auf Blockgrößen von 32 KB verzichten, weil die dann verwendeten Größen (System-Extent-Größen sind abhängig von der Blockgröße) gerade bei kleineren Objekten zu einem enormen Speicher-Overhead führen.

Beide Arten von Tablespaces können mit automatischem Segmentspace-Management erzeugt werden, womit die Freelist-Verwaltung der darin liegenden Objekte durch Bitmaps ersetzt wird. Dies wirkt sich in einem Data Warehouse, in dem hauptsächlich auf „DIRECT LOAD INSERT“ gesetzt wird, jedoch kaum aus. Andererseits führt dies dazu, dass das Default-Kompression-Flag des betreffenden Tablespaces nicht mehr geändert werden kann, weshalb man den Tablespace für Fachbereichs-User einmal komplett neu anlegen musste, nur um dieses Flag einzuführen.

Aber auch „DICTIONARY MANAGED“-Tablespaces können durchaus noch einen sinnvollen Einsatz finden, solange der SYSTEM-Tablespace selbst noch in diesem Modus betrieben wird. Insbesondere kann man es für Teilbereiche einer Datenbank durchaus als Feature betrachten, dass in diesen der mit Oracle 10 eingeführte Recyclebin-Mechanismus (UNDROP TABLE) nicht unterstützt wird, was ja sonst nur durch Abschalten auf Session-Ebene zu erreichen ist, nicht jedoch beispielsweise für alle Objekte eines bestimmten Schemas. In solchen Tablespaces ist dann niemals die PURGE-Erweiterung beim DROP erforderlich, um eine Tabelle endgültig zu löschen; es funktioniert einfach wie vor der Einführung des Recyclebin-Mechanismus.

Arbeitstabellen, die nur während der Laufzeit einer Verarbeitung sinnvolle Daten enthalten, werden bei erfolgreichem Abschluss einer Verarbeitung durch ein Truncate-Statement geleert. Da immer nur ein kleiner Teil aller möglichen Verarbeitungen gleichzeitig laufen wird, bedeutet dies, dass auch immer nur ein Teil die-

ser Tabellen gleichzeitig ihren üblichen Platzbedarf hat. Also sollte man solche Tabellen in einem gemeinsamen Tablespace zusammenfassen, der dann auch nur den maximal gleichzeitig erforderlichen Platz umfassen muss.

Ein solcher Tablespace sollte für die maximale I/O-Call-Datenmenge als UNIFORM-Extentsize konfiguriert sein, um optimale I/O-Performance zu erreichen und trotzdem bei leeren Tabellen möglichst wenig Platzbedarf vorhalten zu müssen. Damit dies für zukünftige Objekte auch sichergestellt ist, sollte bei neu erzeugten Tabellen grundsätzlich automatisch verifiziert werden, dass deren INITIAL und NEXT Storage-Parameter mit denen des Tablespaces übereinstimmen. Dies kann nicht direkt in einem Trigger geschehen, weil sogar in einem „AFTER CREATE“-Trigger die Tabellen noch nicht in dba_segments sichtbar sind. Also bleibt nur die Möglichkeit eines kurzperiodischen Jobs zur Überprüfung. Im Trigger kann eine zusätzliche Information wie os_user der erzeugenden Session in einer Tabelle hinterlegt werden. Der Job hat dann die Möglichkeit einer gezielten Benachrichtigung des Creators im Falle der Nicht-Übereinstimmung, beispielsweise per E-Mail.

Da eine solche Tabelle immer als Arbeitstabelle in einem speziellen Tablespace angelegt wird, kann diese auch sofort wieder gefahrlos via DROP aus der Datenbank entfernt werden. Diese Maßnahme stellt sicher, dass die Größen-Empfehlung identisch zum Tablespace auch von allen Entwicklern/DBAs befolgt wird. Bei 1000 Arbeitstabellen (Anmerkung des Autors: bei uns werden im DWH-Bereich mehr als 1200 verwendet, und im Staging-Bereich mit komplett eigenständiger automatischer Verwaltung eher noch mehr) bedeutet das bei Verhinderung von 100 MB INITIAL Extents eben 99 GB Ersparnis an Speicherplatz, ohne dass dafür besonderer Aufwand erforderlich ist, wovon wegen der Nicht-Gleichzeitigkeit der Verwendung eben auch der größte Teil davon nicht im Tablespace vorgehalten werden muss.

Da alle diese Arbeitstabellen immer mit DIRECT LOAD INSERT befüllt werden, haben diese auch grundsätzlich das COMPRESS-Flag gesetzt, was auch schon im Tablespace als Default vorhanden ist. In Zusammenwirkung all dieser Maßnahmen lässt sich sagen, dass bei den Arbeitstabellen dadurch eine Platzersparnis von über ein 1 TB möglich wurde (Vergleich mit nicht sofort geleerten Arbeitstabellen).

Eine Trennung von Tabellen- und Index-Tablespaces wurde früher empfohlen, um eine bessere Performance zu erreichen. Bei heutigen SAN-Systemen in SAME-Technologie ist dies kein Grund mehr für eine derartige Aufteilung. Allerdings steht außer Zweifel, dass eine solche Aufteilung eine bessere Übersichtlichkeit bedeutet. Zudem kann ein Tablespace, der nur Indizes enthält, einfach neu erzeugt werden, falls Probleme mit

der Hardware auftreten. Man kann also den vorigen Zustand durch Neuerzeugung der Indizes auf dem neuen Tablespace wieder herstellen, ohne auch nur das Wort RECOVER zu denken.

Als Daten-Dateien sollte man immer eine Variante wählen, die eine Form von ASYNC-I/O erlaubt und das Kernel-Buffering umgeht. Dazu eignen sich Character-Devices oder auch sogenannte Quick-I/O-Files, wie sie von Veritas unterstützt werden, und die auch nur eine Form von Character-Devices darstellen, oder das von Oracle angebotene ASM, sofern man schon Oracle 10 einsetzt. Zusätzlich zur Vermeidung von Overhead durch Kernel-Buffering sind meist auch erst bei solchen Technologien größere I/O-Mengen pro Call einsetzbar, während bei normalen Dateien of kleinere Limits gelten.

Der UNDO-Tablespace sollte hinreichend groß sein, um eine undo_retention-Zeit abzudecken, die idealerweise größer sein sollte als alle im Regelfall vorkommenden Verarbeitungszeiten. Damit kann man die früher häufigen Snapshot-too-old-Fehler nahezu komplett vermeiden. Das Unternehmen des Autors setzt dafür 280 GB bei einer undo_retention von 2,5 Tagen ein. Wegen der UNDO-Vermeidung durch die meist verwendeten „DIRECT LOAD INSERT“-Statements (im folgenden Abschnitt beschrieben) reicht dieser Platz im Regelfall für den angegebenen Zeitraum aus. Alle moderneren Verarbeitungen benötigen hier deutlich unter einem halben Tag, lediglich bei zwei älteren, noch nicht auf Massenverfahren migrierten, wird die undo_retention-Zeit insgesamt überschritten, aber eben nicht durch die einzelnen Cursor-OPEN-Zeiten innerhalb dieser Verarbeitungen. Deswegen wäre dieser Abschnitt auch schon fast vergessen worden, weil Snapshot-too-old-Fehler heute kein Thema mehr sind.

Strategien zum Umgang mit großen I/O-Mengen

Selbstverständlich ist für eine Datenbank, die große I/O-Mengen bewältigen muss, eine dafür geeignete Hardware notwendig, die große Datenmengen pro einzeltem I/O-Call unterstützt – 1 MB pro Call sollte heutzutage Standard sein. Dadurch wird insgesamt weniger CPU-Zeit pro I/O-Durchsatz benötigt, weil für die gleiche Datenmenge weniger I/O-Calls und damit weniger ressourcenintensive Switches in den Kernelmode erforderlich sind. Damit wird auch der Grenzbereich später erreicht, der zu einer Zunahme von I/O-Waits führt und damit die Performance noch weiter reduzieren würde.

Früher war es oft ein enormer Vorteil, physikalische Platten exklusiv so für Redolog-Dateien zu verwenden. Eine Platte wurde zeitweise entweder nur beschrieben (Logging aus der DB) oder später wiederum nur gelesen (Archivierung), um dadurch die erforderlichen Positionierungen des Schreib-Lese-Kopfs auf der Platte

zu minimieren. Bei heutigen SAN-Systemen in SAME-Technologie bringt dies jedoch keinen entscheidenden Vorteil mehr. Falls diese nicht ausreichend ist, gibt es noch die Möglichkeit, für die Redolog-Dateien auf Festkörperspeicher-Technologie aufgebaute Pseudo-Disks einzusetzen.

Die Datenbank-Parametrisierung muss stimmig zur eingesetzten Hardware sein. In den derzeit gängigen Oracle-Versionen 9 und 10 ist das eigentlich nur noch `db_file_multiblock_read_count`, der auf einen Wert gesetzt wird, der für die kleinste verwendete Blockgröße in der Datenbank noch Read-Calls mit maximal von der Hardware unterstützten Größe zulässt. In den 8i-Versionen gab es noch entsprechende DB-Parameter für `hash` und `sort`, die jedoch seither nur noch als Hidden Parameter mit führendem Unterstrich existieren und nicht mehr modifiziert werden sollten.

Da bei Joins von Tabellen der Hash-Mechanismus gegenüber Nested-Loops immer dann überlegen ist, wenn große Datenmengen in mehreren beteiligten Tabellen betrachtet werden müssen, ein Fall, wie er in einem Data Warehouse häufiger vorkommt, ist der DB-Parameter `hash_join_enabled` auf TRUE zu setzen, falls die Datenbank noch nicht mindestens mit der Version 10 betrieben wird. Dort ist das dann ein Hidden Parameter, der als Standard auf TRUE gesetzt ist.

Möglichkeiten zur Reduzierung der I/O-Mengen

Performantes Bereitstellen von I/O in hinreichender Bandbreite ist natürlich immer notwendig, aber letztendlich ist es doch erforderlich, geeignete Möglichkeiten zur I/O-Reduktion einzusetzen – die kostengünstigere Lösung. Hierzu sind verschiedene gleichzeitig anwendbare Verfahren geeignet. Bei den meisten dieser Verfahren ist damit auch gleichzeitig eine Reduzierung des Speicherplatzbedarf gegeben – überhaupt lässt jede Speicherplatzreduzierung die Erwartung entstehen, dass damit auch eine I/O-Reduktion einhergeht.

a) Partitionierung

Hier bietet sich zuallererst die Partitionierung von Tabellen und Indizes an. Bei geeigneter Wahl der Partitionierungskriterien lässt sich so schon vom Optimizer eine Eingrenzung der zu lesenden Daten erreichen (Partition Pruning). Damit sind auch sehr hohe Reduktionsfaktoren zu realisieren. Wenn eine Tabelle in 100 Partitionen aufgeteilt werden kann, der Optimizer feststellt, dass nur eine davon gelesen werden muss, bedeutet das eine I/O-Reduktion auf 1 Prozent. Eine Planung muss hier je nach Daten sehr spezifisch erfolgen, meist wird jedoch irgendeine Form der Partitionierung nach geeigneten Zeiteinheiten eine angemessene Form darstellen. Wichtig ist dabei noch, dass die Befüllung

und Auswertung durchaus andere optimale Partitionierungskriterien erfordern können, sodass im Idealfall eine Kombination aus beiden angewendet wird. Bei der Befüllung oder auch Bereitstellung für andere System hat sich gerade bei größeren Datenmengen pro Lieferung schlichtweg je eine Partition pro Lieferung als am einfachsten zu handhabende und effektivste Lösung bewährt. Es sollte auch, wenn immer möglich, versucht werden, nur lokal partitionierte Indizes zu verwenden, weil durch die identische Partitionierung zur Tabelle dann während oder nach Partition-Splitting der Tabelle kein langsames UPDATE GLOBAL INDEXES oder ein langsames Komplett-Rebuild erforderlich ist. Auch Reorganisationen auf Tablespace-Ebene werden damit weniger langsam.

In einem Data Warehouse werden viele Daten nach einer gewissen Vorhaltezeit wieder aus dem System entfernt. Auch hier bringt die Partitionierung entscheidende Vorteile, weil diese Entfernung durch die DDL-Kommandos DROP beziehungsweise TRUNCATE auf Partitionsebene möglich ist. Große Mengen durch UNDO-Segmente verursachten I/O werden so einfach vermieden. Um durch die Partitionierung nicht zusätzlich administrative Aufwände zu erzeugen, ist es sinnvoll, dass alle Verarbeitungen die benötigten neuen Partitionen jeweils selbst erzeugen und auch nicht mehr benötigte automatisch wieder entfernen (Rolling-Window-Prinzip). Ab Oracle 11 kann hier für Neuentwicklungen durch Einführung der INTERVAL-Partitionierung eine gewisse Vereinfachung erwartet werden. Neue Partitionen werden dann automatisch angelegt, sobald Daten dafür in die Tabelle geschrieben werden, ohne dass sie jedoch zuvor schon Speicherplatz für ein INITIAL-Extent benötigen.

Eine ähnliche Form der I/O-Reduzierung kann auch bei der Befüllung mit Daten zu Tragen kommen. Dazu ist es erforderlich, weitestgehend mit DIRECT LOAD INSERT (INSERT /* + APPEND */) zu arbeiten und auf normale Inserts und Updates, wo immer möglich, zu verzichten. Damit wird ebenfalls das Schreiben von UNDO-Segmenten und damit wesentliche I/O-Mengen einfach vermieden. Parallel-DML sollte bei solchen „DIRECT LOAD INSERT“-Statements vermieden werden, weil jeder schreibende Parallel-Prozess dabei einen eigenen Unterbereich jenseits der High-Water-Mark zum Befüllen zugewiesen bekommt und am Ende die neue High-Water-Mark jenseits des höchsten gefüllten Blocks aller Einzelprozesse gesetzt wird. Die am Ende nicht gefüllten Teilbereiche aller anderen Parallel-Prozesse sind dann für nachfolgende „DIRECT LOAD INSERT“-Statements nicht mehr zu nutzen, was schon einen merklichen zusätzlichen Platzverbrauch bewirken kann. Diese Einschränkung der Parallelisierung gilt jedoch nicht, wenn mehrere Parallel-Prozesse so

verwendet werden, dass je einer für die Befüllung einer Zielpartition zuständig ist, und natürlich auch nicht für das datenliefernde SELECT-Statement.

Der Einsatz solcher „DIRECT LOAD INSERT“-Statements erfordert jedoch eine ganz spezielle Form von Verarbeitungen mit Massen-SQL-Statements und gegebenenfalls Arbeitstabellen in größerer Anzahl zum Festhalten der Zwischenergebnis-Werte. Die Realisierung dieser Arbeitstabellen als GLOBAL-TEMPORARY-Tables hat sich nicht als sinnvoll erwiesen, weil die Zwischenergebnis-Werte dann im Fehlerfall nicht zur Problemanalyse zur Verfügung stehen, während normale Arbeitstabellen eben nur nach erfolgreichen Verarbeitungen mit TRUNCATE geleert werden, und natürlich zu Sicherstellung korrekter Ergebnisse auch immer vorher. Solche Arbeitstabellen einschließlich etwaiger Indizes darauf werden grundsätzlich mit NO-LOGGING angelegt. Damit wird in Zusammenarbeit mit den „DIRECT LOAD INSERT“-Statements verhindert, dass RedoLog-Information für die eingefügten Daten geschrieben werden und damit zusätzlich I/O generiert wird. Die Arbeitstabellen können damit nicht recovered werden, was jedoch kein Problem darstellt, da die Daten darin nach erfolgreich abgeschlossener Verarbeitung nicht mehr benötigt werden.

b) Daten-Komprimierung

Letztendlich bleibt nach allen Ausschlussmöglichkeiten immer eine gewisse Datenmenge übrig, die tatsächlich gelesen werden muss. Die dafür erforderliche I/O-Menge kann jedoch reduziert werden, indem diese Daten in komprimierter Form gespeichert werden. Mehr noch, es wird auch weniger Buffer-Cache für die gleiche Datenmenge benötigt, weil die Daten im Cache in gleicher Form wie auf dem Speichermedium abgelegt sind. Man erhält hierdurch also einen doppelten Performancegewinn, einmal durch die reduzierte I/O-Menge bei komprimierter Ablage und zusätzlich eine I/O-Reduktion durch mehr Treffer im Buffer-Cache, weil der durch den dort benötigten geringeren Platzbedarf effektiver genutzt wird. Es gibt natürlich einen geringen Overhead durch die erforderliche Dekomprimierung, der jedoch gegenüber den Gewinnen durch die I/O-Reduktion vernachlässigbar ist. Damit das so funktioniert, wurde die Komprimierung von Oracle so implementiert, dass immer die gesamte notwendige Information innerhalb eines Datenblocks vorhanden ist.

Der Komprimierungsalgorithmus ist einfach eine spaltenübergreifende Ersetzung gleicher Werte durch einen kleinen Wert aus einer Symbol-Tabelle, die dann die Rückführung auf den Originalwert erlaubt, wobei die komplette Symbol-Tabelle eines Blocks innerhalb eben dieses Blocks gespeichert wird. Das lässt sofort erwarten, dass mit größeren Datenblöcken eine effek-

tivere Komprimierung möglich ist. Oracle 9 erlaubt es, zusätzlich zur Standard-Blockgröße einer Datenbank, einzelne Tablespaces auch mit anderen Blockgrößen zu erzeugen. Bei einer Standard-Blockgröße von 8 KB bringt ein Übergang auf Tablespaces mit einer Blockgröße von 32 KB eine Verbesserung der Komprimierung von rund 10 Prozent. Die ist nicht besonders viel und rechtfertigt im Regelfall auch keine Umstellung. Wenn jedoch aus anderen Gründen eine Reorganisation von Daten erforderlich wird, sowie auch für neue Tabellen, bietet sich der Einsatz solcher Tablespaces an, um ohne zusätzliche Aufwände diese 10 Prozent einfach mitzunehmen.

Im Normalfall kann man einen Komprimierungsfaktor zwischen 2 und 3 erwarten. Dies lässt sich jedoch noch wesentlich verbessern, wenn die Daten geeignet vorsortiert in die Tabellen eingefügt werden, weil dann eben wesentlich mehr gleiche Werte in einem Block zu erwarten sind. Damit sind dann Komprimierungsfaktoren von 5 möglich. Während die Komprimierung selbst wegen der verringerten zu schreibenden Datenmenge insgesamt nur einen geringen Performance-Verlust bewirkt, kann natürlich das Sortieren großer Datenmengen schon ziemlich ressourcenintensiv sein. Für ein optimales Ergebnis ist es wichtig, dass hier schon überprüft wird, wie eigentlich sortiert wird. Dazu wird eine einigermaßen repräsentativer Ausschnitt aus den Daten auf die Kardinalität seiner Spalten untersucht. Die Ergebnisse werden zur Festlegung der Sortierung verwendet. Im Falle von dynamisch ausgeführten Statements (EXECUTE IMMEDIATE oder dbms_sql) bietet sich auch die Möglichkeit an, die Analyse der Kardinalität zur Laufzeit auszuführen und damit eine datenoptimierte ORDER-BY-Klausel dynamisch zu erstellen.

Bisher wurde die Komprimierung nur unter dem Gesichtspunkt der Performanceverbesserung der I/O-Last betrachtet. Aber natürlich wird dadurch auch die Größe des benötigten Storage-Systems deutlich vermindert. In Anbetracht der Tatsache, dass man für den hohen I/O-Durchsatz eines DWH-Systems relativ hochwertige Hardware benötigt, lassen sich hier erhebliche Einsparungspotenziale finden.

Es gibt jedoch auch Einschränkungen bei den Einsatzmöglichkeiten: Vor Oracle 11 werden Daten grundsätzlich nur mit DIRECT LOAD INSERT komprimiert, auch wenn das entsprechende Flag des Datensegments gesetzt ist. Wenn später Updates der Daten stattfinden, führt dies zu einer Dekomprimierung der betroffenen Blöcke und sogar zu Chained-Rows, da die dekomprimierten Blöcke die darin enthaltenen Daten grundsätzlich nicht mehr komplett aufnehmen können. In solchen Fällen ist eine Daten-Komprimierung also unbedingt zu vermeiden. Man kann jedoch Tabellen mit Updates noch komprimieren, wenn sich eine Partitionierung finden lässt, die dafür sorgt, dass die erforder-

lichen Updates nicht mehr alle Partitionen betreffen. Solche nicht mehr zu ändernden Partitionen können dann nachträglich mit ALTER TABLE ... MOVE PARTITION ... COMPRESS komprimiert werden.

Die Daten-Komprimierung ist seit Version 9i Release 2 möglich, wobei es andere einschränkende Bedingungen für den Einsatz gibt. Ein ADD COLUMN auf komprimierte Tabellen funktioniert erst ab Version 10, während ein DROP COLUMN in beiden Version nicht möglich ist (zu Version 11 fehlt die Information noch). Tabellen, von denen FOREIGN KEYS auf andere Tabellen zeigen, erzeugen zwar keinen Fehler, aber die Kompression wird bei diesen Oracle-intern einfach ignoriert (als Workaround für einen Foreign-Key-Bug). Um in solchen Fällen eine Komprimierung zu erreichen, kann man entweder die FOREIGN-KEYS während des INSERT-Vorgangs abschalten oder bei partitionierten Tabellen mit EXCHANGE PARTITION arbeiten.

c) Index-Komprimierung

Auch für Indexe gibt es eine Komprimierungsmöglichkeit, sogar schon seit Oracle 8i, die jedoch anders funktioniert als die Daten-Komprimierung. Dabei werden mehrfach vorkommende Werte der Index-Keys, beginnend von ersten an – analog zum Mechanismus bei der Datenkomprimierung – durch einen Wert mit weniger Speicherbedarf ersetzt. Da Indizes von Natur aus sortiert sind, ist die Reihenfolge der Daten in der Tabelle hier nicht von Bedeutung für den Komprimierungsgrad. Wie man aber sofort erkennen kann, hat die Kardinalität der Index-Keys einen wesentlichen Einfluss. Bei einfachen UNIQUE-Keys ist gar keine Komprimierung möglich, weil es keine mehrfachen Werte gibt. Von der Syntax her ist eine derartige Anweisung auch überhaupt nicht möglich.

Bei Indizes mit gleichen Keys in unterschiedlicher Reihenfolge komprimieren diejenigen besser, bei denen die Spalten mit geringerer Kardinalität führend angeordnet sind, was ja nicht unbedingt dem üblichen Index-Design entspricht, weil durch die umgekehrte Anordnung bei unvollständiger Key-Angabe die kürzeren Range-Scans zu erwarten sind. Insgesamt lässt sich sagen, dass diese Komprimierung wesentlich weniger effektiv ist als die Daten-Komprimierung – ein Komprimierungsfaktor von 1,5 ist hier schon sehr gut. Dies führt in vielen Fällen dazu, dass es Indizes gibt, die mehr Platz benötigen als die zugrunde liegenden Tabellen.

Bei der Index-Erstellung oder beim Rebuild ist auch zu beachten, dass ein paralleler Vorgang auf einem einzelnen Segment immer Space-Overhead erzeugt. Um dies zu vermeiden, empfiehlt sich eine Parallelisierung nur dann, wenn mehrere Partitionen existieren und für jede dieser Partitionen wiederum nur ein Parallelprozess zuständig ist.

d) Index-Vermeidung

Wie oben angemerkt, sind die üblichen Statements in einem Data Warehouse so angelegt, dass meist die gesamte Datenmenge nach einem Standard-Kriterium, üblicherweise ein Zeitraum, gelesen werden muss. Wie oben gezeigt, ist es sinnvoll, Tabellen genau nach solchen Kriterien zu partitionieren, sodass von vornherein die korrekte Datenmenge zum Lesen vom Optimizer ausgewählt wird. Hier stellt sich jetzt die Frage, ob man überhaupt einen Index benötigt. Falls er nicht zur Sicherstellung der Datenintegrität erforderlich ist, sollte in solchen Fällen immer auf den Index verzichtet werden. Lediglich dann, wenn auch Einzelwert-Abfragen erfolgen sollen, ist eine Anlage sinnvoll. Dies ermöglicht einerseits Platzersparnis jenseits aller Komprimierungsmöglichkeiten, verbessert aber auch die Performance bei den Befüllungsoperationen, weil eben das immer mit einer mehr oder weniger aufwändigen Sortierung verbundene Erstellen der Indizes und natürlich auch die zum Schreiben notwendige I/O-Last entfällt.

Backup

Auch wenn sich bei separierten Index-Tablespaces ein Teil der Datenbank einfach rekonstruieren lässt, ist ein Backup-Verfahren grundsätzlich unabdingbar. Heutzutage ist der Einsatz von RMAN die Methode der Wahl, während andere Verfahren meist nicht auf die Oracle-Datenbank abgestimmt sind oder eben entsprechend Implementierungsaufwand erfordern. Bei RMAN mit ARCHIVELOG-Backup kann man eben dann auf jeden beliebigen Zeitpunkt ein Recovery durchführen, während dies bei einer Ebene tiefer ansetzenden Verfahren, wie beim Kopieren einer abgesplitteten Mirror-Komponente, nicht ohne weiteres möglich ist. Die oben vorgeschlagene Verwendung separater Tablespaces für alle großen Tabellen bietet auch hier einen Vorteil, indem bei Bedarf ein relativ schnelles Recovery einzelner Tabellen ermöglicht wird.

Man handelt sich mit RMAN allerdings den Nachteil ein, dass während eines Backup-Laufs, sei es nun ein reiner ARCHIVELOG-Backup oder ein kompletter FULL-DATABASE-Backup, die Maschine doch einer relativ hohen I/O-Last ausgesetzt ist, was dann alle anderen Vorgänge auf der Datenbank deutlich weniger performant laufen lässt. Hier wirkt sich auch die Befüllung aller Arbeitstabellen im NOLOGGING Mode äußerst vorteilhaft aus – da zur Sicherstellung des laufenden Betriebs die Archive-Logs mehrmals täglich gesichert und gelöscht werden müssen, würde eine größere Menge hier die Zeiten mit verringerter Performance in der Datenbank entsprechend ausweiten. Eine Verdopplung der Zeiten für Verarbeitungen oder Auswertungen während Backup-Aktivitäten ist da nichts Besonderes. Man kann sich

dadurch behelfen, in dem man die doch einige Stunden dauernden Full Backups nur an den Wochenenden laufen lässt. Seit Einführung des Block-Change-Trackings mit der Version 10 existiert eine sinnvolle Möglichkeit, inkrementelle Backups auch zu verwenden, sodass mit deren täglicher Anwendung auch ein schnelleres Recovery im Desasterfall möglich wird, ohne erst mehrere TB an Archive-Logs zu verarbeiten. Da zuvor auch für inkrementelle Backups die gesamte Datenbank gelesen werden musste, ist diese Option auch unter der Berücksichtigung von Performance und Zeit uninteressant.

Abhängigkeiten

Um die komplexen Abhängigkeiten zwischen Hunderten verschiedener Verarbeitungen befriedigend zu handhaben, ist eine geeignete Steuerung erforderlich. Diese muss unbedingt auf den in Tabellen vorliegenden dynamischen und statischen Metadaten beruhen, um eine hinreichende Transparenz und Flexibilität zu ermöglichen. Das Kodieren und damit Verstecken solcher Abhängigkeiten in obskuren Skripten ist hierzu absolut unzureichend. Die dynamischen Metadaten ermöglichen gleichzeitig auch jederzeit eine Statusabfrage mittels Standard-SQL sowie eine entsprechende grafische Visualisierung. Ein eigener Scheduling-Mechanismus in der Datenbank, der direkt auf diesen Metadaten aufbaut, ist hier die optimale Lösung. Mitteilungen von außen können einfach über eine weitere Abhängigkeit abgebildet werden, ebenso wie spezielle Freigaben für Verarbeitungen. Es muss hier auch absolut sichergestellt sein, dass nicht automatisch korrigierbare Fehlersituationen zum Abbruch einer Verarbeitung führen und damit jegliche davon abhängige Verarbeitungen überhaupt erst nicht gestartet werden. Der Aufwand für manuelle Korrekturen und Eingriffe steigt sonst nahezu ins Unermessliche.

Gerade eine solche Steuerung macht es auch einfach, größere monolithische Verarbeitungen zu vermeiden, weil keinerlei Ablauflogik dort direkt implementiert werden muss, und auch keinesfalls werden darf, wenn flexible Umstellungsmöglichkeiten bei notwendigen Änderungen der Abhängigkeiten wegen Geschäftsprozess-Änderungen existieren sollen. Dann sind lediglich die jeweiligen Metadaten zur Steuerung geeignet zu modifizieren. Bei hinreichend kurz laufenden Verarbeitungen ist es auch vollkommen unnötig, sich Gedanken über eine Wiederaufsetzbarkeit nach Abbruch im Fehlerfall zum machen – die komplette Verarbeitung wird nach Bereinigung der Fehlersituation einfach neu gestartet. Dies entlastet ebenfalls den Verarbeitungscode von unnötigen Steuerungs-Fragmenten und erlaubt dem Entwickler, sich auf die eigentliche Datenaufbereitung zu konzentrieren.

Allein durch die Abhängigkeiten wird bereits geregelt, dass nur jeweils ein geringer Teil aller Verarbeitungen

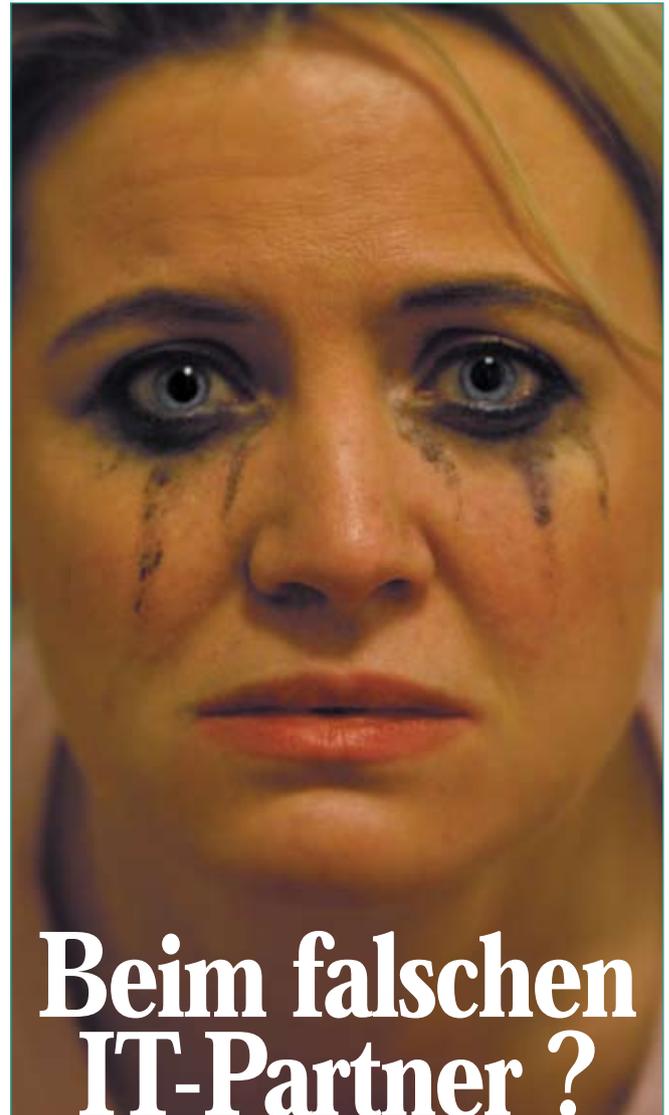
gleichzeitig läuft. Trotzdem kann es vorkommen, dass gleichzeitig zwei voneinander unabhängige Verarbeitungen laufen, die etwa eine gleichartige Aggregation für Woche beziehungsweise Monat ausführen und deshalb auch mit demselben Code-Segment arbeiten. Deshalb werden sie auch die gleichen Arbeitstabellen verwenden, sodass es hier ohne spezielle Maßnahmen zu Datenvermischungen kommen würde. Üblicherweise werden in solchen Fällen partitionierte Arbeitstabellen verwendet und die richtige Partition über den Parameter, also Woche beziehungsweise Monat, ausgewählt. Alternativ kann über eine Ressourcen-Zuordnung in solchen Fällen ein gleichzeitiges Laufen verhindert werden.

Eine andere Form von Abhängigkeit stellt die gemeinsame Nutzung von TEMPORARY-Tablespace für Sortier- und Hash-Operationen dar, wenn dies dazu führt, dass die jeweilige Ressource TEMPORARY-Tablespace nicht mehr ausreichend ist. Einfaches Vergrößern schützt dabei jedoch nicht vor fehlerhaften manuell erstellten Statements (zum Beispiel kartesisches Produkt wegen fehlender WHERE-Bedingung). Deshalb ist es sinnvoll, für alle automatischen Verarbeitungen, alle Fachbereichsbenutzer, sowie für Entwickler/DBAs jeweils eine eigenen TEMPORARY-Tablespace vorzusehen. Damit kann ein Fachbereichs-User oder ein DBA nicht mehr versehentlich eine laufende Verarbeitung durch komplette Füllung des TEMPORARY-Tablespaces zum Abbruch bringen.

Selbstverständlich sollte auch sichergestellt sein, dass einzelne Benutzer nicht beliebig viele Sessions und damit auch Parallelprozesse erhalten können und damit die komplette Maschine soweit überlasten, dass kein sinnvolles Arbeiten mehr möglich ist. In dba_profile bietet sich hierfür eine hervorragende Möglichkeit zur Limitierung an. Hier hat man ebenso die Option, Sessions nach einer bestimmten Idle-Time automatisch beenden zu lassen. Die meisten derartig beendeten Sessions wurden von den jeweiligen Benutzern sowieso vergessen. Wichtig ist dies einmal zur Freigabe von solchen vergessenen Ressourcen, aber auch zur Freigabe irgendwelcher Locks, was gerade beim Einsatz von Toad nicht zu vernachlässigende Gesichtspunkte sind.

Versionierung der Daten

In einem Data Warehouse sind nicht wie in einem operativen System nur die aktuellen Varianten der verschiedenen Datenarten gespeichert, sondern gerade bei Stamm- und Referenz-Daten sind auch frühere Versionen verfügbar. Jeder Satz erhält einfach zusätzlich einen Gültigkeits-Zeitbereich, womit auch automatisch die korrekte Reihenfolge definiert ist. Weitere Informationen wie eine Verlinkung durch Ablegen des Vorgänger- und/oder Nachfolger-Primary-Key-Werts in



Beim falschen IT-Partner ?

Einfach wechseln !

Mit uns sichern Sie sich bedarfsgerechte Beratung und Schulung für Ihren Projekterfolg. Sie profitieren von der PROMATIS Technologie- und Geschäftsprozess-Kompetenz und unserer internationalen Erfahrung. Unsere bewährten Vorgehensmodelle und Softwarekomponenten für Web-Portale, Workflow und Content Management sorgen für wirtschaftliche Oracle® Lösungen.

PROMATIS



Knowledge Powered Business Processes

PROMATIS software GmbH

Tel.: +49 7243 2179-0 · Fax: +49 7243 2179-99

www.promatis.de · hq@promatis.de

Ettlingen/Baden · Hamburg · Berlin

Arbeiten mit großen Datenbanken

einer zusätzlichen Spalte können die Performance bestimmter Abfrageformen deutlich erhöhen. Da es bei der Datenbank vor Oracle 11 hierzu keine Unterstützung gibt, ist eine geeignete lokale Implementierung zur Speicherung dieser historisierten Daten erforderlich.

Bei kleineren Datenmengen sind Trigger zur Historienverwaltung die Methode der Wahl. Seit es INSTEAD-OF-Trigger für Views gibt (ab Oracle 8.0), empfiehlt es sich, genau diesen Mechanismus einzusetzen und dabei die DML eben nur über eine View durchzuführen. Dies hat den entscheidenden Vorteil, dass alle im Trigger notwendigen Informationen jederzeit aus der Tabelle gelesen werden können, ohne dass man sich über das Mutating-Problem (ORA-04091) Gedanken machen muss. Derartige Trigger sind standardisierter Code, das immer gleichartige Operationen ausführt, egal auf welcher Tabelle-View-Kombination. Lediglich deren Namen sowie die Namen der Spalten und deren Anzahl sind unterschiedlich. Deshalb ist es eine einfache Übung, den entsprechenden Trigger-Generator als PL/SQL-Prozedur zu implementieren. Als Parameter müssen Tabellen- und View-Name sowie die Namen der Zusatzspalten für die Historisierung angegeben werden – die restliche Information wird einfach aus dem Data-Dictionary abgefragt.

Größere Datenmengen sind so natürlich nicht performant zu verarbeiten, weil ein Trigger die Daten eben Satz für Satz durcharbeitet. In diesen Fällen ist die komplette Kodierung der Historisierung als Massen-Statement dann sinnvoll. Betroffen sind hiervon jedoch nur wenige Stammdaten-Tabellen, während sämtliche Referenztabellen – und das sind Hunderte – mit dem Trigger-Mechanismus auskommen. Nichtsdestotrotz sollte auch auf solchen speziell befüllten großen Tabellen der Trigger-Mechanismus implementiert werden, um manuelle Korrekturen/Pflege zu ermöglichen, ohne dass sich die betreffende Person zuvor jedes Mal den kompletten History-Mechanismus aneignen muss.

Fazit

In einer großen Datenbank und ganz besonders in einem Data Warehouse gibt es doch einige, für kleinere Datenbanken unübliche Vorgehensweisen, die aber aus Kosten- und Performance-Gründen durchaus angebracht sind, auch wenn dazu ein etwas spezielleres Know-how erforderlich ist. Zu erwähnen bleibt, dass die besondere Form der Massendaten-Programmierung natürlich etwas anspruchsvoller ist als simples Cursor-Looping und üblicherweise auch nicht im Erfahrungsschatz von Entwicklern zu finden ist, die bisher nur mit kleineren Datenbanken zu tun hatten.

Zudem sollten DWH-Entwickler weitreichende Kenntnisse von Partitionierung, Kompression etc. haben, was ja oft eher dem DBA-Bereich zugeordnet wird. Beim Einsatz externer Entwickler sollte deshalb immer ein hinreichender Einarbeitungszeitraum berücksichtigt werden. Aber letztendlich ist es diese Methode, die die Aufbereitung bestimmter Daten in einem vorgegebenen Zeitrahmen mit gegebener Hardware oder manchmal unabhängig von jeder aktuell verfügbaren Hardware überhaupt erst ermöglicht. Eine Reduzierung der Verarbeitungszeit von fünf Tagen mit Cursor-Loops-Verarbeitung auf zwei Stunden nach Umstellung auf Massen-Statements ist durchaus als Standardbeispiel zu betrachten, wobei natürlich bei den Massen-Statements auch Parallelisierung eingesetzt wird, sodass während dieser Zeit auch eine höhere Systemlast generiert wird.

Natürlich gibt es auch in einem Data Warehouse eine große Zahl kleinerer Tabellen, bei denen – solange keine anderen größeren beteiligt sind, beispielsweise bei deren Befüllung – auch mit den üblichen Methoden vorgegangen werden kann. Man muss nicht grundsätzlich alles mit hochperformanten Massendaten-Verfahren bearbeiten, sondern nur bei Bedarf. Die Kunst in einem Data Warehouse liegt darin, rechtzeitig vor der Realisierung – also idealerweise schon bei der Planung, zu erkennen, wo spezielle Performance-Anforderungen notwendig sein werden. Dann kann schon an dieser Stelle ein realistischer Zeitplan für eine Entwicklung dafür sorgen, dass es nicht zu ungeplanten Verzögerungen durch solche Performance-Anforderungen kommt. Nichtsdestotrotz ist ein Testsystem in der Größenordnung des Produktionssystems unverzichtbar, wenn zutreffende Aussagen über das zukünftige Laufzeitverhalten in der Produktionsumgebung getroffen werden sollen. Idealerweise ist dies auch mit realen Produktionsdaten befüllt, weil nur dann die Daten in den gleichen Verteilungen wie im Produktionssystem vorliegen und dies entscheidenden Einfluss auf die Performance haben kann. Letztendlich kann auch nur bei Verfügbarkeit echter Daten all das erforderliche Work-Arounding für fehlerhafte Ausprägungen implementiert werden.

Kontakt:

Dr. Kurt Franke
Kurt.Franke@de.debitel.com

Weitere Informationen zu den Themen der DOAG News finden Sie im Mitgliedsbereich <http://www.doag.org/>

Alternativen zum Hash-Join beim Zugriff auf sehr große Datenmengen

Autor: Daniel Günther, SHS-Viveon AG

In Data-Warehouse-Umgebungen gilt der Hash-Join als performantester (Equi)Join zwischen kleinen Dimensions- und großen Fakten-Tabellen. Der Artikel erklärt die Funktionsweise der Strategien, beschreibt die Performance-Tests und leitet Empfehlungen für den Praxis-Einsatz ab.

Die Performance des Hashjoins wird maßgeblich durch die zur Verfügung stehende Hash-Area-Size beeinflusst, die bei manuell verwalteter PGA auf maximal 2 GB je Session konfiguriert werden kann. Dies ist vor allem für Dimensions-Tabellen aus dem Retail- beziehungsweise Telekommunikationssektor zu wenig, sodass es zu langen Laufzeiten beim Ausführen von Joins kommt. Zwei Optimierungsstrategien können in bestimmten praxisrelevanten Szenarien die Performance des Hash-Joins beim Anreichern und Aggregieren von Data-Warehouse-Daten deutlich schlagen:

- Hash-Cluster-Single-Tables
- PL/SQL-Tabellen

Optimierungsansätze

Der Oracle Optimizer entwickelt einen Execution-Plan, der die Dimensions-Tabellen mit der Fakten-Tabelle via Hash-Join verbindet. Im ersten Moment erscheint es wenig sinnvoll, gegen diese Standardstrategie einen Nested-Loop oder Sort/Merge-Join zu benutzen. Der klassische Nested-Loop schneidet wegen der vielen Index-Zugriffe schlecht ab. Beim Merge-Join müssen die kleinen Dimensions-Tabellen genau wie die große Fakten-Tabelle entsprechend des Join-Kriteriums sortiert werden. Doch auch der Hash-Join hat seine Nachteile. Insbesondere die starke Nutzung des TEMP-Tablespace fällt hier auf.

Zur Erinnerung: Beim Hash-Join werden die Datensätze der kleineren Tabelle mittels Hash-Funktion auf dem Join-Kriterium in Hash-Buckets eingelesen. Die Gesamtheit der Hash-Buckets ist die Hash-Tabelle. Im nächsten Schritt wird die große Tabelle gelesen und wiederum der Hash-Wert des Join-Kriteriums ermittelt. Ist dieser in der Hash-Tabelle vorhanden, erfüllt der Datensatz die Bedingung und wird zurückgegeben. Für weiterführende Informationen ist die Metalink Note 41954.1 empfohlen.

Die Hash-Tabelle wird in der PGA (im dedicated Server-Mode) gehalten und mittels des Parameters hash_

area_size bei manueller PGA-Verwaltung konfiguriert. An dieser Stelle entsteht das Performance-Problem mit Hash-Joins. Die Hash-Area-Size kann nur mit maximal 2 GB konfiguriert werden, wovon die Oracle Datenbank gemäß Dokumentation etwa 20 Prozent für die interne Verwaltung benötigt. Insbesondere bei großen Datenbanken wie im Data-Warehouse-Umfeld reicht das nicht aus. Einzelne Buckets der Hash-Tabelle müssen in den Temp(orary)-Tablespace ausgelagert und zu einem späteren Zeitpunkt in die PGA zurückgeladen und verarbeitet werden.

Die Verwendung der automatisch verwalteten PGA ist mit dem verwendeten Oracle Release 10.2.0.3 nicht möglich, da es einen Bug bei der Verwendung von PGA_AGGREGATE_TARGET in Verbindung mit Solaris 10 gibt (Metalink Note 460424.1). Die Nutzung von PGA_AGGREGATE_TARGET könnte das Problem entschärfen, wenn PGA_AGGREGATE_TARGET deutlich größer als 40 GB konfiguriert wird, da für die Hash-Area-Size per Default 5 Prozent zur Verfügung gestellt werden.

Die Performance des Hash-Joins hängt von verschiedenen Bedingungen ab, beispielsweise von den Mengen an Datensätzen in den Tabellen. Der Artikel beschäftigt sich in erster Linie mit der Abhängigkeit der Performance von der Hash-Area-Size und soll verschiedene Ideen aufzeigen, wie und unter welchen Bedingungen die Ausführungszeit durch die Hash-Area-Size limitierter Joins verkürzt werden kann.

Hash-Cluster-Single-Table und Nested Loop

Bei einer Hash-Cluster-Single-Table werden die Datensätze beim Einfügen gemäß einer Hash-Funktion in die Datenbank-Blöcke verteilt. Beim Suchen kann mit der gleichen Hash-Funktion der Block ermittelt werden, in dem der gesuchte Datensatz gespeichert ist. Im Gegensatz dazu werden Datensätze in eine Heap-Table ohne jegliche Ordnung in den nächsten freien Block eingefügt. Für eine schnelle Suche nach einem einzelnen Satz ist der Aufbau eines Index notwendig.

Arbeiten mit großen Datenbanken

Der Vorteil, Dimensionen in einer Hash-Cluster-Single-Table zu speichern, liegt im Vergleich zur Heap-Table plus Unique-Index beziehungsweise IOT darin, dass beim Suchen nach exakt einem Wert nur ein Logical-I/O benötigt wird. Bei IOTs werden mindestens zwei, bei der Kombination Index und Heap-Table mindestens drei Logical-I/Os je Zugriff auf einen einzelnen Satz benötigt.

Die Nutzung der Hash-Cluster-Single-Table eröffnet die Möglichkeit, die Fakten-Tabelle performant mittels Nested-Loop zu joinen. Konstruktionsbedingt wird für den Join via Nested-Loop keine separater Memory-Bereich in der PGA benötigt. Die Hash-Area-Size verliert ihre limitierende Wirkung.

Für den Aufbau einer Hash-Cluster-Single-Table legt man zuerst ein Cluster an mit den Attributen, über die später gesucht werden soll. Im Beispiel (siehe Abschnitt «Testmodell») ist das der Primär-Schlüssel der Dimensions-Tabelle. Wichtig sind die Angaben von Size und Hash-Keys sowie die Schlüsselwörter SINGLE TABLE. Mit Hash-Keys wird die Anzahl der eindeutigen Werte in der Dimensions-Tabelle festgelegt. Über Size wird der Platz in Bytes definiert, der pro Key abgelegt wird. Ist das Cluster-Atribut numerisch und gleichverteilt, kann man mit der Angabe HASH IS <Clusterattribut> die interne Hash-Funktion umgehen. So gelingt der Zugriff noch schneller. Die Datensätze werden dann gemäß dem Wert im Cluster-Atribut im Cluster verteilt:

```
CREATE CLUSTER c1_VARCHAR2(id_varchar2
VARCHAR2(50))
  SIZE 100
  SINGLE TABLE
  HASHKEYS 10000000
  PCTFREE 1
  STORAGE (BUFFER_POOL KEEP);
```

Bei Anlage des Clusters wird sofort der benötigte Platz allokiert. Nach Anlage des Clusters wird die Dimensions-Tabelle in den Cluster gelegt.

```
CREATE TABLE DIM_VARCHAR2_CLUST
( DV_ID    VARCHAR2(50) NOT NULL PRI-
MARY KEY
  ,DV_DESC VARCHAR2(100)
  ,DV_GROUP VARCHAR2(100))
CLUSTER c1_varchar2(DV_ID);
```

Bei der Verwendung der Statements aus Tabelle 4 sollten die angegebenen Execution-Pläne genutzt werden. In der linken Spalte sind Execution-Pläne des Hash-Joins, in der rechten die des Nested-Loop aufgeführt. Wichtig ist, dass beim Nested-Loop per TABLE ACCESS HASH auf die Dimensions-Tabelle zugegriffen wird. Da

	Hash Join / Heap Table	Nested Loop / Hash Cluster Single Table
Ins	<pre> Id Operation Name ----- 0 INSERT STATEMENT 1 1 LOAD AS SELECT TARGET 2 HASH JOIN RIGHT OUTER 3 TABLE ACCESS FULL DIM_VARCHAR2_HEAP 4 TABLE ACCESS FULL FCT_DATA</pre>	<pre> Id Operation Name ----- 0 INSERT STATEMENT 1 1 LOAD AS SELECT TARGET 2 NESTED LOOPS OUTER 3 TABLE ACCESS FULL FCT_DATA 4 TABLE ACCESS HASH DIM_VARCHAR2_CLUST</pre>
Agg	<pre> Id Operation Name ----- 0 INSERT STATEMENT 1 1 LOAD AS SELECT TARGET 2 HASH JOIN RIGHT OUTER 3 TABLE ACCESS FULL DIM_VARCHAR2_HEAP 4 TABLE ACCESS FULL FCT_DATA</pre>	<pre> Id Operation Name ----- 0 SELECT STATEMENT 1 HASH GROUP BY 2 NESTED LOOPS OUTER 3 TABLE ACCESS FULL FCT_DATA 4 TABLE ACCESS HASH DIM_VARCHAR2_CLUST</pre>

Tabelle 1: Executionpläne für Hash-Join und Nested-Loop / Hash-Cluster-Single-Table

für jeden Datensatz aus der Fakten-Tabelle durch den Table-Access-Hash ein Logical-I/O notwendig wird, ist drauf zu achten, dass die Blöcke der Dimensions-Tabelle bereits im Buffer-Cache vorliegen und keinen teuren Physical-I/O verursachen.

Leider hat die Nutzung eines Hash-Clusters auch Nachteile. Nach seiner Anlage ist der Cluster statisch. Ändert sich die Anzahl der Keys, muss der Cluster neu aufgebaut werden. Das Einfügen der Datensätze dauert länger als das Einfügen in eine Heap-Table, da der Datenbank-Block erst via Hash-Funktion ermittelt werden muss.

PL/SQL-Table-Lookup

PL/SQL-Tabellen sind eindimensionale Arrays skalarer oder strukturierter Datentypen. Die Elemente können entweder mit ganzen Zahlen (BINARY_INTEGER) oder mit Zeichenketten (VARCHAR2) indiziert werden. Die PL/SQL-Tabelle liegt in der UGA. Sie ist im Gegensatz zu Heap-Table und Hash-Cluster nicht persistent. Beim Suchen nach einem Datensatz wird mit der EXIST-Funktion das Vorhandensein eines bestimmten Werts geprüft. Der Vorteil liegt darin, dass für diese Operation weder Logical- noch Physical-I/O notwendig sind. Es fällt lediglich CPU-Zeit zum Suchen in der PL/SQL-Tabelle an. Es liegt auf der Hand, dass auch für den PL/SQL-Table-Lookup kein Speicherplatz in der Hash-Area-Size benötigt wird.

Für den Aufbau einer PL/SQL-Tabelle ist ein Package notwendig. Darin wird eine globale Variable angelegt, die die Werte der Dimensions-Tabelle aufnimmt:

```
CREATE OR REPLACE PACKAGE dim_varchar2_
plsqtalab
AS
  TYPE dim_varchar2_plsqtalab_t IS TABLE
OF dim_varchar2_heap%ROWTYPE INDEX BY
dim_varchar2_heap.dv_id%type;
  g_dim_varchar2_plsqtalab dim_varchar2_
plsqtalab_t;
  PROCEDURE initialize;
  FUNCTION lookup(pv_id IN dim_varchar2_
heap.dv_id%type) RETURN VARCHAR2;
END dim_varchar2_plsqtalab;
/
```

Im Package-Body werden eine Prozedur zum Laden der Dimensionswerte in die PL/SQL-Tabelle und eine Funktion zum Suchen (Lookup) eines Dimensionswerts implementiert. Der Aufruf von „initialize“ am Ende des Package-Bodys sorgt dafür, dass spätestens beim ersten

Lookup – während des Initialisierens des PL/SQL-Packages – alle Werte in die PL/SQL-Tabelle geladen werden:

```
CREATE OR REPLACE PACKAGE BODY dim_var-
char2_plsqtalab
AS
  PROCEDURE initialize IS
  BEGIN
    FOR s in (SELECT dv_id
              , dv_desc
              ,dv_group
              FROM dim_varchar2_heap)
    LOOP
      g_dim_varchar2_plsqtalab(s.dv_id).
dv_id      := s.dv_id;
      g_dim_varchar2_plsqtalab(s.dv_id).
dv_desc    := s.dv_desc;
      g_dim_varchar2_plsqtalab(s.dv_id).
dv_group   := s.dv_group;
    END LOOP;
  END initialize;
  --
  FUNCTION lookup(pv_id IN dim_varchar2_
heap.dv_id%type) RETURN VARCHAR2
  IS
  BEGIN
    RETURN g_dim_varchar2_plsqtalab(pv_
id).dv_group;
  EXCEPTION
  WHEN NO_DATA_FOUND THEN
    RETURN NULL;
  END lookup;
BEGIN
  initialize;
END dim_varchar2_plsqtalab;
/
```

Die Lookup-Funktion gibt nur einen Wert und nicht den ganzen Record der Dimensions-Tabelle zurück. Tests mit Objekt-Typen, die den gesamten Record als Rückgabewert haben, waren um das zwei- bis dreifache langsamer gegenüber einer Implementierung mit skalaren Rückgabewerten. Hier muss man entscheiden, ob man mit mehreren Lookup-Funktionen, die skalare Rückgabewerte haben, oder mit einer Lookup-Funktion, die den gesamten Record als Objekttyp zurückgibt, arbeiten möchte.

Der Execution-Plan zeigt nur den Full-Table-Scan der Fakten-Tabelle. Ähnlich dem TABLE-ACCESS-HASH für jeden Satz aus der Fakten-Tabelle muss die Lookup-PL/SQL-Funktion aufgerufen werden.

Arbeiten mit großen Datenbanken

	PL/SQL Table Lookup Statements	Execution Plan
Ins	<pre> INSERT /*+ APPEND */ INTO TARGET (DV_ID, D_GROUP, F1_DATE, F2_NUMBER, F3_ VARCHAR) SELECT f.dv_id ,dim_varchar2_plsqltab.lookup(f.dv_id) ,f.f1_date ,f.f2_number ,f.f3_varchar FROM fct_data f; </pre>	<pre> Id Operation Name ----- 0 INSERT STATEMENT 1 LOAD AS SELECT TARGET 2 TABLE ACCESS FULL FCT_DATA </pre>
Agg	<pre> SELECT dim_varchar2_plsqltab.lookup(f.dv_id) ,MAX(f.f1_date) ,SUM(f.f2_number) FROM fct_data f GROUP BY dim_varchar2_plsqltab.lookup(f.dv_id); </pre>	<pre> Id Operation Name ----- 0 SELECT STATEMENT 1 HASH GROUP BY 2 TABLE ACCESS FULL FCT_DATA </pre>

Tabelle 2: PL/SQL-Table-Lookup-Statements und Execution-Pläne

Die Lage der PL/SQL-Tabelle in der UGA hat den Nachteil, dass nur die Session, in der die PL/SQL-Tabelle aufgebaut wurde, darauf zugreifen kann. Man kann diese Daten nicht mit anderen Sessions gemeinsam nutzen. So fällt der Overhead, die PL/SQL-Tabelle füllen zu müssen, mit jeder neuen Session an. Das bedeutet, dass man bei der Nutzung von PL/SQL-Tabellen die Anreicherung oder Aggregationen so optimieren muss, dass das gleiche Set an Dimensions-Tabellen für mehrere Fakten-Tabellen genutzt werden kann. Bei Tests ergab sich eine Begrenzung der PL/SQL-Tabellen auf etwa 10 GB. Oracle bestätigte dieses Limit auf Nachfrage in der Metalink-Note 6576035.993.

Testmodell

Die vorgestellten Methoden zum Joinen der Tabellen sollen anhand eines kleinen Modells untersucht werden. Es werden eine Dimensions- und eine Fakten-Tabelle verwendet. Die DV_ID ist ein VARCHAR2-Feld und wird mit konstant 20 Bytes befüllt. Die Felder DV_DESC und F3_VARCHAR sind ebenfalls vom Datentyp VARCHAR2 und werden so gefüllt, dass die unten stehenden Größen durch die Tabelle belegt werden.

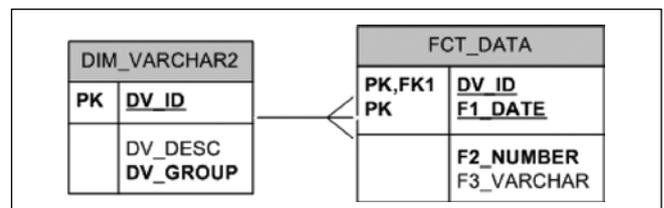


Abbildung 1: ERD-Testmodell

	Datensätze	Größe
DIM_VARCHAR2_HEAP	10 Mio	1 GB
DIM_VARCHAR2_CLUST	10 Mio	1 GB
FCT_DATA	40 Mio	8 GB

Tabelle 3: Datenmengen

Auf die Partitionierung wird hier bewusst verzichtet, da die Aggregationen und Anreicherungen im Data Warehouse auf großen Datenbeständen zumeist inkrementell, beispielsweise für einen bestimmten Monat einer monatsweise partitionierten Tabelle durchgeführt werden. Die Fakten-Tabelle spiegelt also das Daten-Volumen eines Monats oder eines Tages wider.

Messmethode

Zur Beurteilung der verschiedenen Join-Alternativen wird einerseits eine Aggregation und andererseits eine Anreicherung durchgeführt.

	Statement
Aggregation	<pre> SELECT dv.dv_group ,MAX(f.f1_date) ,SUM(f.f2_number) FROM fct_data f , dim_varchar2 dv WHERE f.dv_id = dv.dv_id(+) GROUP BY dv.dv_group; </pre>
Anreicherung	<pre> INSERT/*+ APPEND */INTO TARGET SELECT f.ds_id ,dv.dv_group ,f.f1_date ,f.f2_number ,f.f3_varchar FROM fct_data f ,dim_varchar2 dv WHERE f.dv_id = dv.dv_id(+); </pre>

Tabelle 4: Statements

Die Dimension liegt in drei Ausprägungen vor:

- Klassische Heap-Table
- Hash-Cluster-Single-Table
- PL/SQL-Table

Die Heap- und Hash-Cluster-Single-Dimensions-Tabellen werden in den KEEP-Pool gelegt, die Fakten-Tabelle verbleibt im DEFAULT-Buffer-Pool. Die Dimensions-Tabellen werden vor der eigentlichen Messung einmalig gelesen, sodass die Blöcke sich im KEEP-Pool befinden und nicht zu einem Physical-I/O führen.

Die Statements werden in unterschiedlich konfigurierten Hash-Area-Sizes seriell und in den Parallelitätsstufen 2 und 4 ausgeführt. Dabei werden Ausführungszeit und die Statistik-Werte Physical-Reads-Direct-Temporary-Tablespace, Physical-Writes-Direct-Temporary-

Table-space sowie Session-Logical-Reads gemessen. Die Statistik-Werte sind V\$MYSTAT und V\$SESSTAT entnommen. In den Ergebnissen wird der Durchschnitt der besten zwei Läufe von mindestens sechs pro Statement dargestellt.

Um die Genauigkeit der Messung nicht unnötig zu verschlechtern, werden bei der Anreicherung die Datensätze in eine mit der Option NOLOGGING erzeugten Tabelle und dem Hint APPEND eingefügt (keine Free-Block-Suche, kaum Redos).

Für den Test stand eine Oracle 10.2.0.3 EE Datenbank auf einem SUN Sparc Multiprozessorsystem mit 28 Ultra Sparc IV Prozessoren unter Solaris 10 zur Verfügung. Die Datenbank ist moderat belastet, sodass keine negativen Wirkungen durch konkurrierende Zugriffe im I/O-Subsystem, beziehungsweise in den Buffer-Caches zu beobachten waren. Das System ist mit 7 x 4 GBit/s an eine Enterprise SAN angebunden.

Ergebnisse

Wie zu erwarten war, nimmt die Laufzeit des Hash-Joins mit zunehmender Hash-Area-Size ab. Bei 2 MB Hash-Area-Size liegt die Laufzeit bei 4.135 Sekunden und nimmt bis 512 MB auf 530 Sekunden ab. Ab 512 MB bringt eine Erhöhung der Hash-Area-Size hingegen keine Laufzeitverkürzung mehr.

Sowohl Nested-Loop mittels Hash-Cluster-Single-Table als auch der Lookup via PL/SQL-Tabelle liefern unabhängig von der Größe der Hash-Area-Size annähernd konstante Laufzeiten zwischen 550 (NL/Cluster) und 600 (PL/SQL-Table-Lookup) Sekunden (siehe Abbildung 2). Nicht enthalten sind 350 Sekunden, die für den Aufbau der PL/SQL-Tabelle benötigt werden. Für das gewählte Beispiel ist die Laufzeit des Hash-Joins erst ab etwa 64 MB Hash-Area-Size besser als die des NL und des PL/SQL-Table-Lookups.

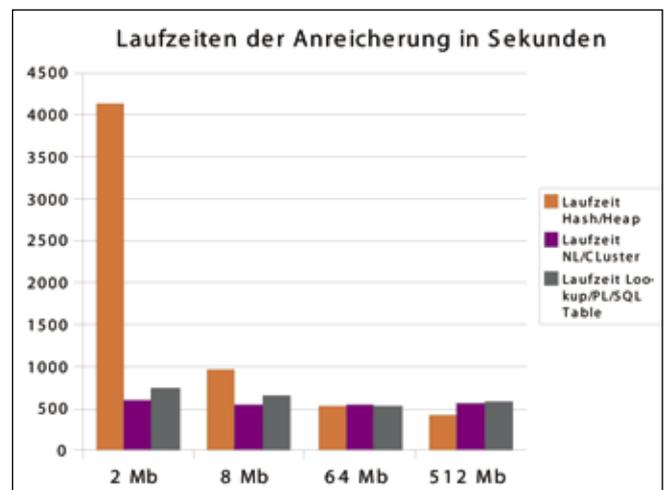


Abbildung 2: Laufzeit der Anreicherung in Sekunden

Arbeiten mit großen Datenbanken

Wie bereits beschrieben, müssen beim Joinen via Hash-Join und zu geringer Hash-Area-Size Teile der Hash-Table auf den Temporary-Tablespace ausgelagert werden. Dabei fallen teure Physical-I/Os an. So werden 5,7 Mio. Physical-I/Os bei 2 MB Hash-Area -Size zum Lesen und Schreiben in den Temporary Tablespace benötigt (siehe Abbildung 3). Erst bei einer Hash-Area-Size zwischen 512 MB und 1024 MB steht so viel Memory zur Verfügung, dass kein Physical-I/O für das Auslagern anfällt. NL und der PL/SQL-Lookup führen zu keiner messbaren Nutzung des Temporary Tablespaces.

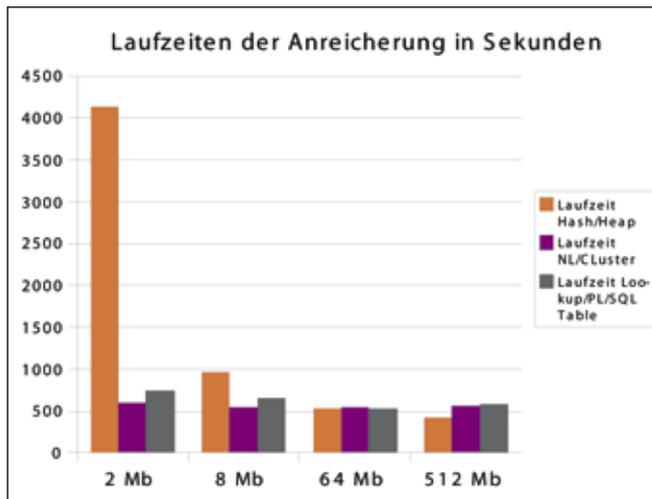


Abbildung 3: Physikal-I/O mit temporärer Anreicherung

Dem Vorteil des Nested-Loops, kein Physical-I/O zum Auslagern in den Temporary-Tablespace für den Join zu benötigen, steht eine massive Steigerung von Logical-Reads gegenüber. So benötigt der Nested-Loop etwa 41,6 Mio. Logical-Reads, unabhängig von der Hash-Area-Size. Dieser Wert zeigt, dass für jeden Satz in der Faktentabelle ein Logical-Read für das Suchen im Hash-Cluster anfällt. Im Beispiel ist die Faktentabelle mit 40 Mio. Sätzen gefüllt. Für den Hash-Join sind 1,09 Mio. und für den PL/SQL-Table-Lookup 1,02 Mio. Logical-Reads notwendig. Die Differenz erklärt sich daraus, dass der Aufbau der PL/SQL-Tabelle mit 10 Mio. Logical-Reads herausgerechnet ist. Der eigentliche Lookup in der PL/SQL-Tabelle führt zu keinem Logical-I/O.

Logical Reads Anreicherung	
Hash/Heap	1.090.758
NL/Cluster	41.626.049
Lookup/PL/SQL Table	1.024.321

Tabelle 5: Logical-Reads-Anreicherung

Die Performance-Unterschiede sind hauptsächlich auf die Kompensation von Physical-I/O gegen Logical-Reads zurückzuführen. Bei den Logical-Reads, beziehungsweise den PL/SQL-Table-Lookups sind nur CPU und Memory beteiligt. So spielen für die Performance die Auslastung des I/O-Subsystems auf der einen Seite, beziehungsweise von CPU und Memory auf der anderen Seite den entscheidenden Faktor.

Die Aggregation

Ähnlich der Anreicherung verkürzt sich die Laufzeit des Hash-Joins bei der Aggregation mit zunehmender Hash-Area-Size. Jedoch sind die Unterschiede zwischen Hash-Join und Nested-Loop/Hash-Cluster-Single-Table geringer als bei der Anreicherung (siehe Abbildung 4). Die Laufzeit des Hash-Joins nimmt von 1209 Sekunden bei 2 MB Hash-Area-Size bis auf 240 Sekunden bei 64 MB ab. Eine weitere Erhöhung der Hash-Area-Size führt zu keiner weiteren Laufzeitverkürzung. Nested-Loop und PL/SQL-Table-Lookup liefern unabhängig von der Hash-Area-Size Laufzeiten von 470 beziehungsweise 430 Sekunden. Nested-Loop und PL/SQL-Table-Lookup erreichen nur bis zu einer Hash-Area-Size von 8 MB kürzere Laufzeiten als der Hash-Join.

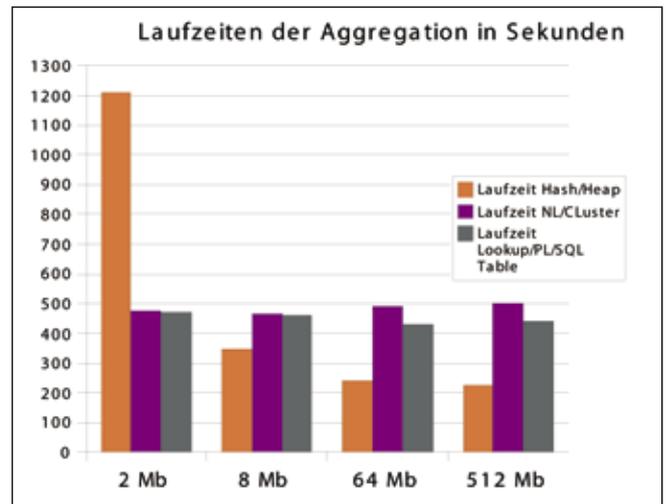


Abbildung 4: Laufzeit der Aggregation in Sekunden

Der Hash-Join beansprucht mit 1,25 Mio Physical-I/Os bei 2MB Hash-Area-Size den Temporary Tablespace nur ein Viertel so stark wie bei der Anreicherung. Deshalb ist die Performance schon bei einer deutlich geringeren Hash-Area-Size (8 MB) besser als beim Nested-Loop und dem PL/SQL-Table-Lookup (siehe Abbildung 5). Bei 512 MB Hash-Area-Size sind nur noch 31.000 Physical-I/Os notwendig, sodass es nur noch einer Erhöhung der Hash-Area-Size um wenige MB bedarf, bis der Hash-Join optimal läuft.

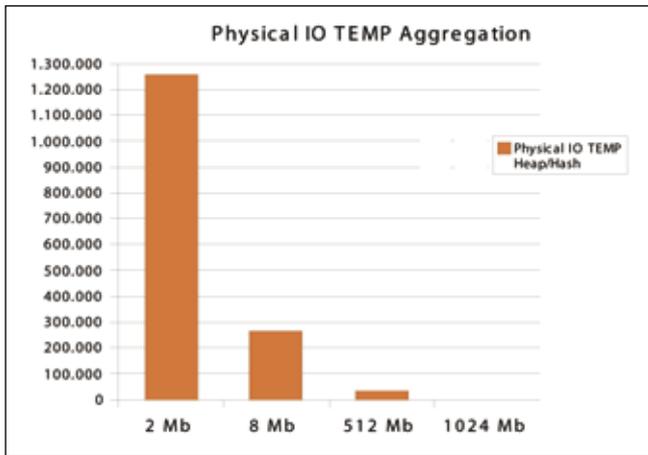


Abbildung 5: Physikal-I/O mit temporärer Aggregation

Parallele Ausführung

Hash-Join, Nested-Loop und PL/SQL-Table-Lookup sind parallelisierbar und erreichen kürzere Laufzeiten. Es gibt jedoch eine Besonderheit beim PL/SQL-Table-Lookup: Jeder Parallel-Prozess muss hier das Package initialisieren und die Dimensions-Tabelle in die PL/SQL-Tabelle laden. Es ist nicht möglich, die SQL-Tabelle in einer Session nur einmal aufzubauen und alle nachfolgenden Statements dieser Session davon profitieren zu lassen. Deshalb kann Zeit für den Aufbau der SQL-Tabelle nicht aus den Laufzeiten für die parallele Ausführung des PL/SQL-Table-Lookups herausgerechnet werden.

Wird angereichert, sinkt die Laufzeit von 2020 Sekunden bis zu einer Hash-Area-Size von 512 MB bei der Parallelitätsstufe 2 ab und bleibt dann auch bei weiter zunehmender Hash-Area-Size konstant bei 350 Sekunden (siehe Abbildung 6). Bei der Parallelitätsstufe 4 bleibt die Laufzeit bereits ab einer Hash-Area-Size von 16 MB konstant bei 330 Sekunden. Der Nested-Loop bleibt unabhängig von der Hash-Area-Size in der Parallelitätsstufe 2 bei etwa 360 Sekunden in der Parallelitätsstufe 4 bei 250 Sekunden. Die Unterschiede zwischen Hash-Join und Nested-Loop sind bei der parallelen Ausführung auch bei ausreichend großer Hash-Area-Size gering. Die Laufzeit für den PL/SQL-Table Lookup liegt unabhängig von Hash-Area-Size und Parallelitätsgrad bei 470 Sekunden.

Bei der parallel ausgeführten Aggregation kann der PL/SQL-Table-Lookup nur in der Parallelitätsstufe 2 bei der vergleichsweise geringen Hash-Area-Size von 2 MB kürzere Laufzeiten als der Hash-Join erzielen (siehe Abbildung 7). Auch der Nested-Loop erreicht nur bis 8 MB Hash-Area-Size vergleichbare Laufzeiten. Der Hash-Join liefert ab 16 MB Hash-Area-Size konstante Laufzeiten von 145 (Parallel 2) und 88 (Parallel 4) Sekunden. Für das gewählte Beispiel sind der Einsatz von Nested-Loop und PL/SQL-Table-Lookup bei paralleler Ausführung nur für eine niedrig konfigurierte Hash-Area-Size sinnvoll.

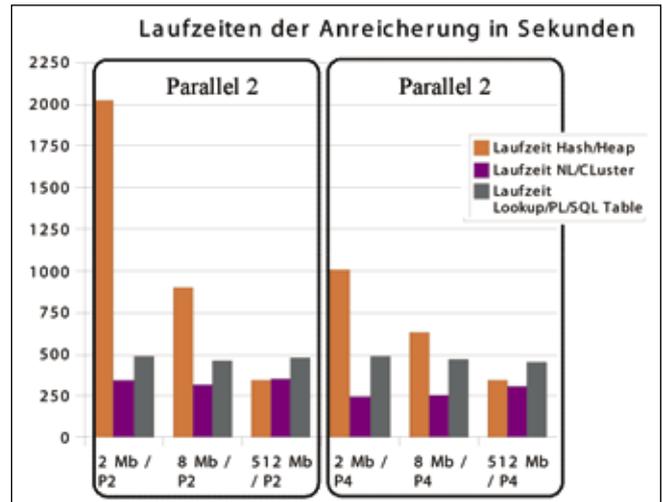


Abbildung 6: Laufzeit der Anreicherung in Sekunden

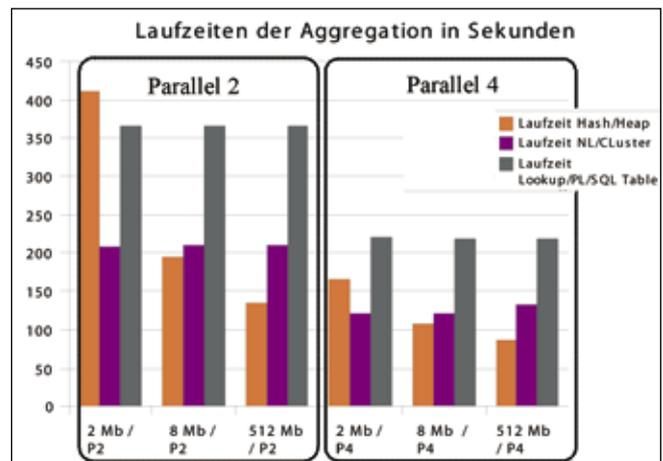


Abbildung 7: Laufzeit der Aggregation in Sekunden

Fazit

Im Wesentlichen hängt es von der Notwendigkeit ab, während des Hash-Joins in den TEMP-Tablespace auszulagern, ob mit Konstrukten wie dem Nested-Loop mit einer Hash-Cluster-Single-Table oder einem Lookup via PL/SQL-Tabelle bessere Laufzeiten zu erzielen sind. Der Nested-Loop in Verbindung mit der Hash-Cluster-Single-Table ist flexibler einsetzbar als der PL/SQL-Table-Lookup, solange man sich innerhalb der SQL-Engine bewegt. Folgende Zusammenhänge bestehen:

- Je mehr Datensätze in der Fakten-Tabelle sind, desto eher wird die Laufzeit des Hash-Joins im Vergleich zu NL/Hash-Cluster, beziehungsweise PL/SQL-Table-Lookup besser sein, da mehr Logical-Reads anfallen
- Je mehr Datensätze in der Dimensions-Tabelle sind, desto eher wird die Laufzeit von NL/Hash-Cluster beziehungsweise PL/SQL-Table-Lookup im Vergleich

Arbeiten mit großen Datenbanken

	Vorteil	Nachteil
Hash Join/Heap Table	Schnellste (Equi)Join Variante bei ausreichendem Memory	Benötigt ausreichend Hash-Area-Size
NL/Hash Cluster Single Table	<ul style="list-style-type: none"> • Ressourcenschonend, da kein zusätzlicher Memory für die Hash-Area-Size benötigt wird • Der Hash Cluster steht allen Sessions zur Verfügung 	<ul style="list-style-type: none"> • Nach Anlage ist die Größe (Hash-Keys) nicht zu ändern • Insert in den Cluster dauert länger als das Einfügen in die Heap Table
Lookup- PL/SQL-Tabellen	Es wird kein zusätzlicher Memory für die Hash-Area-Size benötigt.	<ul style="list-style-type: none"> • Nur Number (BINARY_INTEGER) und VARCHAR2 Datentypen sind indizierbar • PL/SQL-Tabellen sind nur in der jeweiligen Session sichtbar • Es können nur ca. 10GB in die PL/SQL-Tabellen je Session geladen werden • Es kann nur ein skalarer Rückgabewert der Lookup-Funktion definiert werden, sonst kommt es zu Performance-Einbußen • Programmieraufwand

Tabelle 6: Vor- und Nachteile der Join-Methoden

zum Hash-Join besser sein, da mehr Physical-I/Os für das Lesen der Dimensions-Tabelle beziehungsweise des Auslagerns der Hash-Table anfallen.

Kontakt:
Daniel Günther
daniel.guenther@shs-viveon.com



3. Juni 2008, Steigenberger Hotel Zeppelin, Stuttgart

DOAG ITIL DAY 2008

Die DOAG Deutsche ORACLE-Anwendergruppe e.V. veranstaltet am 3. Juni 2008 im Steigenberger Hotel ‚Graf Zeppelin‘ in Stuttgart den DOAG ITIL DAY 2008.

ITIL bietet die Grundlage zur Verbesserung von Einsatz und Wirkung einer operationell eingesetzten IT-Infrastruktur. An der Entwicklung von ITIL waren IT-Dienstleister, Mitarbeiter aus Rechenzentren, Lieferanten, Beratungsspezialisten und Ausbilder beteiligt. ITIL beschreibt die Architektur zur Etablierung und zum Betrieb von IT Service Management.

Der DOAG ITIL DAY 2008 richtet sich vor allem an ORACLE Anwender, die in folgenden Bereichen tätig sind:

- ▶ Prozessmanagement
- ▶ IT-Service-Management
- ▶ Business Service Management
- ▶ Serviceorientierte Architektur

▶ Alle Infos und anmelden unter www.doag.org/go/itil



Well Balanced Systems

Autoren: Marcus Bender und Kai Fischer, ORACLE Deutschland GmbH

Viele Unternehmen haben in den letzten Jahren mit einer Datenflut zu kämpfen. Ziel dieses Artikels ist es, eine Referenz-Architektur für ein „Well Balanced System“ aufzuzeigen, das leistungsstark, flexibel skalierbar und kostengünstig mit großen Datenmengen umgehen kann.

Erfahrungen aus verschiedenen Projekten zeigen, dass Kunden häufig mit den gleichen Problemen kämpfen: Verschiedene Projekte laufen auf getrennter Hardware; große Datenmengen müssen von Rechner zu Rechner kopiert werden; unterschiedliche Rechner sind auf Spitzenlast ausgelegt. Ist ein Rechner für mehrere Stunden ohne Auslastung, kann er nicht für andere Projekte eingesetzt werden. Insbesondere solche einzelnen Systeme sind nicht in der Lage, auf Businessanforderungen wie das Jahresendgeschäft zu reagieren. Hinzu kommen Storageprobleme: Es gibt sowohl Schwierigkeiten mit der I/O-Leistung als auch mit Kapazitätsengpässen. Mangelnde I/O-Leistung stellt eines der Hauptprobleme bei den meisten Projekten dar und zwar um so mehr, je größer die Datenmengen werden.

Ausgewogenheit der Komponenten

Mangelnde I/O-Performance, insbesondere der I/O-Durchsatz, ist eines der Hauptprobleme bei Projekten, die mit großen Datenmengen zu tun haben, speziell bei Data-Warehouse-Projekten. Die Ursache dafür liegt in der zunehmenden Diskrepanz zwischen der Kapazität einer einzelnen physikalischen Festplatte – verglichen mit ihrer I/O Leistung [1]. Vor zehn Jahren hat eine gängige Server-Festplatte eine Kapazität von 4 GB bei einem Durchsatz von 15 MB/sec. Heute sind Platten mit 73 GB Kapazität schon selten, häufiger findet man sie mit 146 GB Fassungsvermögen. Diese Platten haben einen Durchsatz von 45 MB/sec. Einer Kapazitätssteigerung um den Faktor 36 steht damit eine Leistungssteigerung Faktor drei gegenüber. Zukünftige Generationen wie die neue 450 GB Server-Festplatte von Seagate werden diesen Trend noch enorm verstärken. Erschwerend kommt hinzu, dass eine einzelne CPU mittlerweile so stark ist, um I/O-Aufträge zu generieren, die mehrere Platten zu 100 Prozent auslastet.

Die stark anwachsenden Datenmengen unterstreichen die Wichtigkeit, auf die Ausgewogenheit der Komponenten zu achten. Dieses als „Amdahl's Law“ [2] bekannte Prinzip zeigt, dass zwischen der CPU-Geschwindigkeit, der Speicher-Bandbreite und der I/O-Performance Ausgeglichenheit herrschen muss. Als

Faustregel gilt: Möchte man die Leistungsfähigkeit einer modernen CPU unter Oracle ausnutzen, dann muss man zwischen 100 und 150 MB/sec. Durchsatz bei 1500 bis 2000 IOs pro Sekunde zur Verfügung stellen. Die Platten sollten aus Performance-Gesichtspunkten als Raid 0+1 anstatt Raid 5 aufgesetzt werden [3]. Demnach benötigen acht moderne CPUs etwa 800 bis 1200 MB/sec. reinen Plattendurchsatz.

Der I/O-Cache ist im Gegensatz zu BI&W-Projekten bei OLTP-Projekten sehr wichtig, weil hier häufig so viele Daten bewegt werden, dass ein I/O-Cache nicht ausreicht, diese anhaltend zu speichern. Bei neun von zehn Kundenprojekten wird diese wichtige Faustregel nicht eingehalten. Mittlerweile bieten jedoch fast alle Hardware-Anbieter einen leistungsfähigen, kleinen „Entry Level Storage“ beziehungsweise „Midrange Storage“ an. Werden mehrere dieser Storage-Systeme gekoppelt, bieten sie eine ähnliche oder sogar höher I/O-Leistung als High-End-Storage Systeme und das zu deutlich günstigeren Kosten.

Im ersten Schritt ist es möglich, mit einer geringen Investition zu starten. Die Referenz-Architektur ist in der Stufe 1 so ausgelegt, dass alle Komponenten, einschließlich CPU, Fibre Channel Controller (FCAL), Switches und Platten-Einheiten mitwachsen können (siehe Abbildung 1). Mit insgesamt vier modernen CPUs und 16 GB Hauptspeicher hat man bereits ausreichend Leistung, um Projekte mit einem Mindestdurchsatz von 400 MB/sec. und 1 bis 2 TB an Daten zu unterstützen.

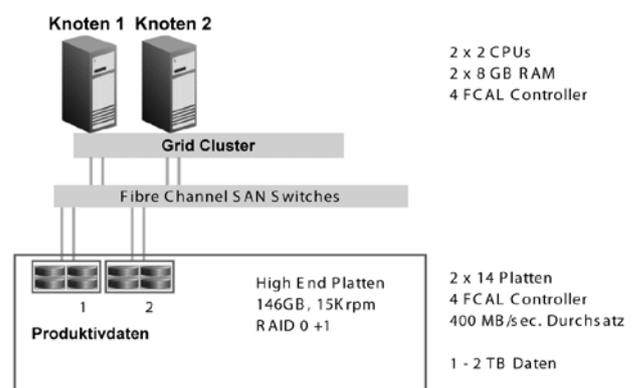


Abbildung 1: Referenz-Architektur, Stufe 1

Arbeiten mit großen Datenbanken

Modularer und flexibler Aufbau

Eine weitere wichtige Anforderung für ein Well Balanced System besteht darin, dass CPU-, RAM- und Storage mit den Projektanforderungen skalierbar sind, und zwar zu möglichst geringen Kosten. Der Anwender muss in der Lage sein, mit einer kleinen Investition das Projekt zu starten und bei Bedarf aufzurüsten – bis in einen Bereich von mehreren Tausend Benutzern und Datenbank-Größen von 5, 10 oder 50 TB. Bei großen SMP-Systemen, die in der Lage sind, derartige Leistung abzudecken, liegt das Problem bei der Anfangsinvestition für das große Kabinett einschließlich Stellflächen- und Wartungskosten. Sehr kostengünstig und trotzdem sehr leistungsfähig sind hingegen kleine Rechner mit maximal vier oder acht CPUs, beziehungsweise zwei und vier Doppelcore-CPU's. Mit einer Technologie wie Oracle Grid Computing und Real Application Clusters kann man diese Rechner zu einem leistungsstarken und zudem noch ausfallsicheren Cluster anwachsen lassen.

In Stufe zwei werden alle Komponenten der Referenz-Architektur verdoppelt (siehe Abbildung 2). Mit acht CPUs, 32 GB Hauptspeicher und 800 MB/sec. Durchsatz können Projekte bis 3 TB effektiv unterstützt werden. Die einzelnen Plattensysteme sind zudem in der Lage, zusätzliche Platten aufzunehmen, um die Kapazität und I/O-Leistung noch weiter zu steigern. Durch Volume Manager wie Veritas oder Oracle ASM ist es möglich, Daten online auf neu hinzukommende Plattensysteme zu verteilen (Online Rebalancing). So werden die Platten-Ressourcen sowohl hinsichtlich Kapazität, als auch bei der I/O-Leistung optimal genutzt. Entgegen dem Markttrend sollten für die Referenz-Architektur allerdings keine Blades mit einer oder zwei CPUs eingesetzt werden, sondern mittelgroße Knoten mit vier oder acht CPUs, denn bestimmte Rechenoperationen, wie Join-, Sortierungs- oder Gruppierungs-Operationen erfordern neben der horizontalen Skalierungsmöglichkeit auch eine gewisse Leistungstärke pro Knoten.

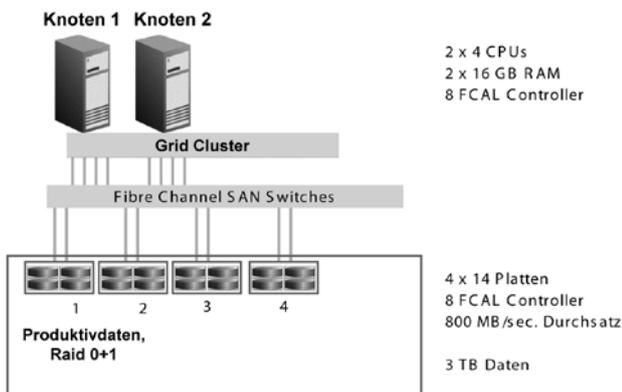


Abbildung 2: Referenz-Architektur, Stufe 2

Der wichtige Kostenaspekt

Aus Kostensicht bietet der Aufbau eines Grid Clusters zwei Vorteile: Man kann sehr günstige Hardware nutzen und trotzdem der hohen Anforderung nach Ausfallsicherheit und Skalierbarkeit nachkommen. Entry-Level Storage mit 28 Platten und einem Durchsatz von rund 250 MB/sec. pro Storage. Einheit ist heute schon für 10.000 bis 20.000 Euro zu haben. Quad-Boards mit vier Cores sind je nach Hersteller, RAM-Kapazität und Rabatten ebenfalls sehr günstig und liegen zwischen 20.000 und 40.000 Euro.

In der optimalen Ausnutzung der Hardware liegt ebenfalls Einsparungspotenzial [4]: Hardware, die auf Lastspitzen ausgerichtet ist, hat bei einer wöchentlichen Betrachtungen Leerzeiten, innerhalb dieser keine optimale Nutzung vorliegt. Benötigt ein Projekt aus speziellen Gründen Ressourcen über die Lastspitze hinaus, ist es schwer beziehungsweise unmöglich, diese zur Verfügung zu stellen. Ein Cluster, aufgebaut aus mehreren kleineren Komponenten, ist flexibler und kann als Infrastruktur für mehrere Projekte genutzt werden. Zum Beispiel benutzen zwei große Projekte einen Vier-Knoten-Cluster, Projekt A zwei Knoten, Projekt B zwei Knoten. Projekt A ist ein Data Warehouse und sehr ETL-intensiv, Projekt B ist ein OLTP-Projekt und hat häufig Probleme, Rechenleistung zwischen 8 und 10 Uhr bereitzustellen. In einem Grid Cluster ist es möglich, Projekt A während der Nacht Ressourcen für den ETL-Prozess von Projekt B zur Verfügung zu stellen und umgekehrt morgens einen Knoten für die OLTP-Verarbeitung von Projekt A an Projekt B abzutreten. Im Jahresendgeschäft könnte Projekt B dann alle vier Knoten nutzen.

Die Referenzarchitektur ist in der Stufe drei eine Konfiguration, die dieses Szenario optimal unterstützt (siehe Abbildung 3). Hier wirken 16 leistungsstarke CPUs und 64 GB Hauptspeicher mit einem I/O-Gesamtdurchsatz von 1,6 GB/sec. Das System ist in der Lage, 6 TB an Daten zu verarbeiten, bei Erweiterung der acht Plattensysteme um weitere 14 Platten kommt man auf 12 TB. Damit ermöglicht man ein Sizing der Knoten nicht auf Lastspitzen der einzelnen Projekte, sondern auf eine optimale Gesamtnutzung aller Ressourcen. Kommt ein Projekt hinzu oder steigt die Anforderung eines Projekts, wächst die Anzahl der Server bedarfsgerecht. Bei horizontaler Skalierung profitieren die anderen Projekte von den zusätzlichen Ressourcen.

Zusätzliche Anforderungen

In dieser Referenz-Architektur sind Anforderungen zum Life Cycle Management, schnelles Backup & Recovery großer Datenmengen von mehreren Terabytes

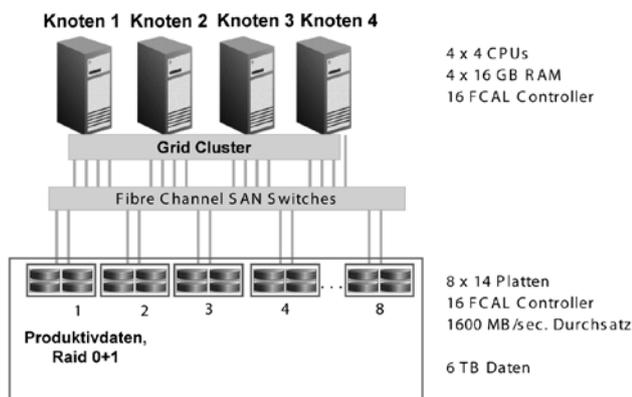


Abbildung 3: Referenz-Architektur, Stufe 3

und Grundkonzepte für eine professionelle Projektunterstützung mit Entwicklungs-, Test- und Produktionsrechner abgedeckt. Für Daten, die in einem Life-Cycle-Management-Konzept nicht so häufig benötigt werden und für Backup & Recovery hat man heutzutage die Möglichkeit, sehr günstige Storage, ausgestattet mit 500/750 GB SATA-Platten, in die Konzepte einzubeziehen (siehe Abbildung 4). 28 Platten in einem RAID-5-Verbund ergeben so eine Nettokapazität von etwa 14 TB. Mit mehreren solcher Storage-Einheiten lassen sich auch große Datenbanken direkt auf Platten zwischenspeichern, was sowohl schnelles Backup als auch schnelles Recovery ermöglicht – schneller als mit Bandsicherungskonzepten. Ein zusätzliches Sichern auf Band wird dennoch aus Disaster-Gründen empfohlen. Allerdings benötigt man für dieses Konzept keine teuren Hochleistungsbandstationen, sondern kann günstige Tape-Stationen nutzen. Mit einem Zwei-Stufenkonzept sichert man zuerst schnell auf Platten und dann im Hintergrund von den Platten in einem längeren Zeitraum auf Band, ohne die Produktiv-Umgebung zu belasten.

Die Referenz-Architektur beinhaltet ein dreistufiges Projekt-Konzept. Die Entwicklungs-Umgebung besteht aus einem kleinen Ein- oder Zwei-CPU Rechner. Hinzu kommt eine Test- und Referenz-Umgebung, für Tests zur Software-Entwicklung, Performance-Tests und Tests von Betriebssystem- und Software-Patches. Die Test-Umgebung sollte so nahe wie möglich an CPU- und I/O-Leistung an die Produktivumgebung angepasst sein. Durch den Einsatz gleicher Hardware-Komponenten sind Entwicklungsrechner und Testrechner sehr flexibel einsetzbar – beispielsweise für Notfälle oder als Leistungsreserven bei speziellen Business-Anforderungen.

Für anspruchsvolle Projekte mit großen Datenmengen investieren viele Kunden in leistungsstarke Großrechner, die sowohl teuer in der Anschaffung als auch im Support sind. Einzelne Projekte werden mit Hardware bestückt, ohne die Möglichkeit zu besitzen, kostbare Leistungsreserven in Form von CPU und I/O

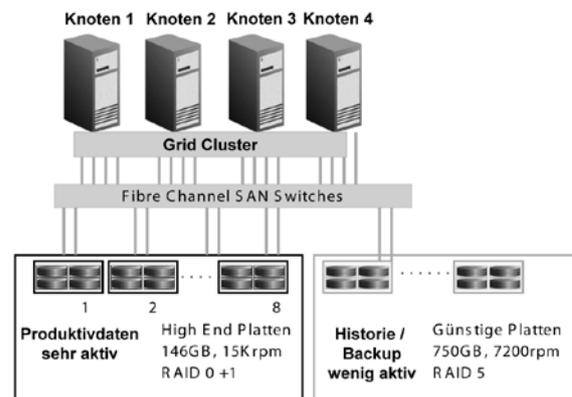


Abbildung 4: Life Cycle Management und Backup & Recovery

anderen Projekten schnell und flexibel zur Verfügung zu stellen. Zudem steht vielen Projekten eine ungenügende I/O-Leistung zur Verfügung, die gerade bei Data-Warehouse-Projekten wichtig ist. Veränderungen, um Missstände zu beheben oder wachsenden Projekt-Anforderungen gerecht zu werden, sind bei vielen Hardware-Architekturen im laufenden Projekt schwierig.

Fazit

Die Referenz-Architektur für „Well Balanced Systems“ unterstützt den Weg vieler Kunden zu einer Homogenisierung der IT-Infrastruktur – und das zu einem sehr guten Preis-/Leistungsverhältnis für Anschaffung und Support. Die Architektur ist so aufgebaut, dass sie sehr flexibel mit den Leistungsanforderungen der Projekte und großen Datenmengen wachsen kann und zudem ein ausgewogenes Verhältnis von CPU- und I/O-Leistung zur Verfügung stellt. Mehrere Projekte können die Referenz-Architektur als Grid-Infrastruktur nutzen und so von vorhandenen Leistungsreserven profitieren.

Literatur

- [1] Hsu, W.W., Smith, A.J., Characteristics of I/O traffic in personal computers and server workloads, IBM System Journal, Volume 42, 2003
- [2] Gustafson, J.L. Reevaluating Amdahl's Law, CACM, 31(5), 1988
- [3] Chen, P.M., Raid: High-Performance, reliable secondary storage, ACM Computing Surveys (CSUR), 1994
- [4] Thome. B., Enterprise Grid Computing gives you better information at a lower cost, Enterprise Java Research Library (EJRL), 2005

Kontakte:

Marcus Bender
marcus.bender@oracle.com
Kai Fischer
kai.fischer@oracle.com

Kampf dem Schweizer Käse – oder wozu reorganisieren?

Autor: Patrick Schwanke, Quest Software GmbH

Das Reorganisieren einer Tabelle ist eine relativ aufwändige Operation, die zudem meist mit einer Auszeit des Systems einhergeht. Dennoch gibt es gerade in Datenbanken häufig gute Gründe dafür.

Bei der Reorganisation steht der Speicherplatz-Gewinn in großen Umgebungen wie ERP-Datenbanken im Vordergrund. Archivierungsläufe, die alte Daten löschen, hinterlassen häufig löchrige Tabellen und damit viel ungenutzten Speicherplatz, der durch eine Reorganisation wieder zugänglich gemacht werden kann. Anhand eines konkreten Projekts wird beispielhaft vorgestellt, wie in einer SAP-Datenbank bis zu 700 GB Speicherplatz eingespart und damit Kosten reduziert werden konnten.

Gerade in großen ERP-Datenbanken ist das Phänomen der „Schweizer-Käse-Tabelle“ weit verbreitet: Durch wiederholte Löscho- und Archivierungs-Jobs sind in den Tabellen und Indizes viele verstreute, kleine und große Datenlöcher entstanden. Nach und nach wird dies aus mehreren Gründen unangenehm:

- Offensichtliche Verschwendung von Speicherplatz, die durch die häufige Verwendung von Spiegelplatten und/oder Standby-Systemen noch multipliziert wird
- Die Performance lesender Zugriffe wird schlechter, da beispielsweise bei nur halb gefüllten Blöcken doppelt so viel I/O erfolgen muss, um dieselben Daten zu lesen
- Die Effizienz des Buffer-Caches wird schlechter, da auch hier die Blöcke nur halb gefüllt sind, also ein großer Teil des Caches aus leerem Raum besteht. Dies kann nur durch einen entsprechend größeren Cache, also mehr Hauptspeicher ausgeglichen werden
- Index-Zugriffe werden ungünstiger, da schlecht gefüllte Indizes tendenziell tiefer sind und daher ebenfalls mehr I/O-Aufwand nach sich ziehen als gut gefüllte Indizes

Der erste Aspekt ist aus Kostensicht recht gut zu greifen und dadurch auch bei allen Arten von ROI-Analysen im Vorfeld einer Reorganisations-Entscheidung beliebt. Aber auch die anderen Aspekte sollten nicht unterschätzt werden, gerade in kritischen Anwendungen, bei denen es auf Performance und Antwortzeiten

ankommt. Manchmal lässt sich durch langfristige Monitoring-Daten (etwa im Rahmen von SLA-Messungen) ein entsprechender Negativ-Trend in der Antwortzeit bestimmter Abfragen ablesen. Eine Reorganisation der zugrunde liegenden Tabellen ist hierbei wirkungsvoller als der beliebte Schnellschuss, im System mehr oder schnellere Hardware einzusetzen.

Die Qual der Wahl

Es gibt mittlerweile eine Vielzahl von Reorganisations-Methoden in der Oracle-Welt. Bei der Klärung der Frage, welche davon sind in welcher Situation sinnvoll sind, liegt der Fokus auf Online-Methoden („Live-Methoden“), die komplett ohne Auszeit des Systems oder zumindest nur mit einer sehr kurzen Unterbrechung auskommen. Neben SQL-basierten Methoden wie Move Table, Index-Rebuild, Segment-Shrink oder Online-Redefinition sind auch Tool-basierte Wege wie beispielsweise LiveReorg und Shareplex der Firma Quest Software möglich. Dabei zeigt sich, dass es keinen eindeutigen Favoriten gibt, sondern dass es stark auf die jeweilige Situation und auf die Randbedingungen ankommt.

Alte Bekannte und neue Wege

Seit langem gibt es die beiden Methoden „Move-Table“ und „Online Index-Rebuild“.

```
ALTER TABLE auftraege MOVE [TABLESPACE
neuer_ts] [ONLINE];

ALTER INDEX auftraege_pk REBUILD [ONLI-
NE];
```

Allerdings gehören beide nur begrenzt in diesen Bereich, denn Index-Rebuild reorganisiert nur Indizes oder Index-Partitionen und keine Tabellen, während Move-Table zwar Tabellen neu aufbaut, aber dies nur in einem seltenen Spezialfall auch tatsächlich online schafft. Normalerweise ist beim Move-Table die Tabelle für die gesamte Zeit des Neuaufbaus gesperrt, nur

bei Index-organisierten Tabellen kann dies durch die ONLINE-Klausel verhindert werden. Beide Methoden sind – sofern die Online-Klausel genutzt wird – an die Oracle Enterprise Edition gebunden. Außerdem benötigen beide temporären Plattenplatz, da das neue Objekt parallel zum noch bestehenden alten Objekt aufgebaut wird. Erst am Ende steht der entsprechend gewonnene Speicherplatz zur Verfügung.

Neue Wege eröffnen die Methoden Online-Redefinition (seit Oracle 9i) und Segment-Shrink (seit Oracle 10g). Online-Redefinition war die erste echte Oracle-eigene Online-Methode. Sie basiert auf dem PL/SQL-Package DBMS_REDEFINITION. Auch hier wird zunächst eine neue Tabelle parallel zur alten aufgebaut. Es ist jedoch nur eine wenige Sekunden dauernde Auszeit in Form einer Tabellensperre zum Ende der Reorganisation hin nötig, wenn die alte und neue Tabelle intern ausgetauscht werden. Seit Oracle 10gR2 kann dies bei partitionierten Tabellen auch partitionsweise geschehen. Dies ist wichtig, da gerade bei großen Tabellen mit mehreren 100 GB der Neuaufbau der kompletten Tabelle erhebliche Last und Redolog-Mengen auf dem produktiven System erzeugen kann.

```
exec DBMS_REDEFINITION.CAN_REDEF_
TABLE(,DEMO','T')
CREATE TABLE t_new AS SELECT * FROM t
WHERE 0=1;
exec DBMS_REDEFINITION.START_REDEF_
TABLE(,DEMO','T','T_NEW')
DECLARE
    num_errors PLS_INTEGER;
BEGIN
    DBMS_REDEFINITION.COPY_TABLE_DEPEN-
DENTS(
    ,DEMO','T','T_NEW',DBMS_REDEFINITI-
ON.CONST_ORIG_PARAMS
    ,TRUE, TRUE, TRUE, TRUE, num_errors);
END;
/
SELECT object_name, base_table_name,
ddl_txt
FROM dba_redefinition_errors;
exec DBMS_REDEFINITION.FINISH_REDEF_
TABLE(,DEMO','T','T_NEW');
```

Listing 1: Die Online-Redifition-Methode

Die Segment-Shrink-Methode ist die einzige Methode, die „In-place“ reorganisiert, also keine temporäre Kopie der Tabelle erzeugt. Sie funktioniert intern wie die bekannte Windows-Defragmentierung, das heißt, die Zeilen werden physisch zusammengeschoben. Dadurch

entsteht, ausgehend von einer „Schweizer-Käse-Tabelle“, zunächst ein großer, zusammenhängender Freiplatz am physischen Ende der Tabelle. Um diesen wieder an den Tablespace zurückzugeben, muss zum Schluss die sogenannte High-Water-Mark der Tabelle nach unten korrigiert werden. Dafür wird eine kurzzeitige Tabellensperre benötigt. Mit der COMPACT-Option wird dieser Schritt übersprungen. Dann ist die Operation zwar komplett online, der freie Speicherplatz verbleibt aber im Segment und steht dort beispielsweise für zukünftige Daten-Inserts zur Verfügung.

```
ALTER TABLE auftraege SHRINK SPACE [COM-
PACT];
```

Beide Methoden haben einige Restriktionen. Insbesondere werden Spalten der Typen LONG und LONG RAW nicht unterstützt. Bei Segment-Shrink muss die Tabelle außerdem in einem Tablespace mit automatischer Segmentplatzverwaltung (ASSM) liegen und die Row-Movement-Eigenschaft der Tabelle aktiviert sein. Während die Tablespace-Art nicht veränderbar ist, kann Row-Movement nachträglich wie folgt aktiviert werden:

```
ALTER TABLE auftraege ENABLE ROW MOVE-
MENT;
```

Auch diese vorbereitende Änderung benötigt eine kurze Exklusivsperrung auf der Tabelle.

Beide Methoden stehen mit allen Editionen der Oracle Datenbank zur Verfügung. Bei Online-Redefinition ist die Situation etwas unklar, da die Dokumentation und die View V\$OPTION Gegenteiliges behaupten. Ein Praxistest zeigt aber, dass die Methode auch auf einer Standard-Edition-Datenbank funktioniert.

Einerseits hat Segment-Shrink einen deutlichen Vorteil hinsichtlich der Einfachheit und Flexibilität: Ohne PL/SQL-Aufrufe können gezielt einzelne Tabellen, Partitionen, Subpartitionen, LOB-Segmente, Indizes, Index-Partitionen oder IOT-Overflow-Segmente reorganisiert werden. Andererseits ist es aber durch den In-place-Mechanismus immer auch eine „Operation am offenen Herzen“, und der Vorgang ist mit einem erhöhten Risiko verbunden.

Tool-basierte Methoden

Einige Tools sollen hier noch kurz vorgestellt und eingeordnet werden: LiveReorg von Quest Software reorganisiert Tabellen inklusive aller Indizes. Es ähnelt stark der Online-Redefinition-Methode, da auch hier eine temporäre Kopie der Tabelle und ihrer Indizes erzeugt und zum Schluss Original- und Klon-Tabelle vertauscht werden, wobei eine wenige Sekunden dauernde

Arbeiten mit großen Datenbanken

Sperre anfällt. Bei partitionierten Tabellen ist wiederum ein partitionsweises Vorgehen möglich. Ein nicht unwesentlicher Vorteil von LiveReorg ist aber die Unterstützung von LONG- und LONG-RAW-Spalten, die beispielsweise bei SAP-Systemen in großen Mengen anfallen.

Ist der Freiplatz über sehr viele Tabellen verteilt oder befindet sich zwischen den Tabellensegmenten (Fragmentierung auf Tablespace-Ebene), lohnt sich eventuell eher eine Reorganisation der gesamten Datenbank. Hierfür gibt es Tools wie Quest Shareplex oder iReflect von IBM (ursprünglich von DataMirror), die technisch gesehen Replikations-Lösungen darstellen. Es wird temporär eine zweite, reorganisierte Datenbank aufgesetzt, die per Replikation aktuell gehalten ist und nach einer Übergangs-/Beobachtungszeit den produktiven Betrieb übernimmt. Dies kann natürlich auch mit einer ohnehin geplanten Migration der Datenbank etwa auf eine andere Plattform, ein neues Oracle-Release oder auf Unicode-Zeichensatz kombiniert werden. Aus Synchronisationsgründen ist zu Beginn der Replikation eine kurzzeitige Sperre auf jeder Tabelle erforderlich. Zum Ende hin muss außerdem die Applikation auf die neue Datenbank „umgebogen“ werden. Beide Tools funktionieren mit jeder Edition der Oracle-Datenbank sowie Oracle-Versionen zwischen 8i und 10gR2.

Ein Beispiel aus der Praxis

In einem Projekt wurden mehrere SAP-Datenbanken von einem Hosting-Provider betrieben und unter anderem nach dem benötigten Plattenplatz in Rechnung gestellt. Da in SAP-Systemen häufig sogenannte Archivierungsläufe durchgeführt werden, die alte Daten aus dem System entfernen, lag die Vermutung nahe, dass bereits große Lücken in den Tabellen bestehen. Daraus

ergab sich der Wunsch, diesen vorhandenen Freiplatz zu konsolidieren, um Storage-Kosten einzusparen beziehungsweise sie für die nächsten Monate zumindest konstant zu halten.

Wegen der in SAP-Datenbanken zahlreich vorhandenen LONG- und LONG-RAW-Spalten kam hier LiveReorg in Frage. Auf Basis einer Analyse wurde ein entsprechender Report in Form eines Excel-Sheets erstellt, in dem alle Tabellen und Indizes aufgelistet waren, geordnet nach dem möglichen Einsparpotenzial, ausgedrückt in Gigabyte. Dabei ist es entscheidend, die weiter unten genannten Klippen zu umschiffen, um hier zu verlässlichen Aussagen zu kommen. Um eine Abwägung zwischen Aufwand und Nutzen zu treffen, wurden in diese Excel-Liste zwei weitere Spalten aufgenommen:

- Nutzen: kumulierter Anteil am Gesamt-Einsparpotenzial
- Aufwand: kumuliertes Daten-Volumen der reorganisierten Objekte

Dabei kamen die folgenden Formeln (Objekt steht hier für Tabelle, Index oder Partition, je nach Feinheit der Analysemethode) zur Anwendung:

$$\text{Kumulierter Anteil}(i) \text{ in } \% = \frac{\text{Summe}(\text{Einsparungen Objekt 1 bis } i)}{\text{Summe}(\text{Einsparungen Objekt 1 bis } n)} * 100,$$

$$\text{Kumuliertes Datenvolumen}(i) \text{ in GB} = \text{Summe}(\text{Größe Objekt 1 bis } i \text{ NACH Reorganisation}),$$

wenn insgesamt n Objekte vorliegen.

Owner	Object Name	Allocation Before	Allocation After	Size Savings	% Savings	Space Usage	HighWater Mark	Stats Date	Stats Source
Tablespaces: DEMODATEN									
DEMO	PK_POSITIONEN	4	0,31	3,69	92,19	0,27	2,95	06.03.2008	Data Dictio...
DEMO	PK_AUFTRAEGE	0,94	0,13	0,81	86,67	0,12	0,82	06.03.2008	Data Dictio...
DEMO	AUFTRAEGE_LIEFERDATUM	0,88	0,13	0,75	85,71	0,11	0,74	06.03.2008	Data Dictio...
DEMO	PK_ADRESSEN	0,25	0,06	0,19	75,00	0,03	0,16	06.03.2008	Data Dictio...
DEMO	PK_PERSONEN	0,13	0,06	0,06	50,00	0,01	0,09	06.03.2008	Data Dictio...
DEMO	PERSONEN_TEMP	8	7	1	12,50	6,96	7,23	06.03.2008	Data Dictio...
DEMO	PERSONEN	8	7	1	12,50	6,96	7,27	06.03.2008	Data Dictio...
Summary:		22,19	14,69	7,5	33,80	14,46	19,27		

Abbildung 1: Screenshot von LiveReorg

Dass bei der Aufwandsberechnung die Objektgrößen „NACH Reorganisation“ benutzt werden, liegt daran, dass dies den Aufwand zur Erstellung der temporären Objekt-Kopie darstellt. Auf Basis dieser beiden Werte kann nun eine Abwägung durchgeführt werden:

- Wie hoch ist das Einsparungspotenzial überhaupt und welche Tabellen sollten in welcher Reihenfolge reorganisiert werden, um dieses möglichst schnell zu realisieren?
- Wie viel Reorganisation lohnt sich? Beispielsweise könnte man den Vorgang nach 90 Prozent des Gesamt-Einsparungspotenzials abbrechen, wenn der Aufwand für die letzten 10 Prozent unverhältnismäßig hoch erscheint.

Die benötigten Rohdaten für diese Analyse müssen entweder aus den Optimizer-Statistiken abgeleitet oder – wie in diesem Fall – von dem verwendeten Tool extrahiert werden.

Fallstricke

Die größten Herausforderungen liegen bei der Gewinnung dieser Rohdaten für die Analyse des Einsparungspotenzials. Dabei ist das absolute Einsparpotenzial (in GB) pro Tabelle ausschlaggebend. Dies lässt sich aus der aktuellen Tabellengröße sowie dem durchschnittlichen Freiplatz pro Block berechnen. Diese beiden Zahlen können prinzipiell auch aus den Optimizer-Statistiken der Oracle Datenbank direkt abgelesen werden. Allerdings wird heutzutage meistens die DBMS_STATS-Methode für die Pflege der Optimizer-Statistiken verwendet. In diesem Fall sollte jedoch der alten ANALYZE-Methode der Vorzug gegeben werden, denn:

- Nur das ANALYZE-Kommando berechnet die entscheidenden Werte wie durchschnittlicher Freiplatz in den Blöcken (Spalte DBA_TABLES.AVG_SPACE) oder Anzahl unbenutzter Blöcke (Spalte DBA_TABLES.EMPTY_BLOCKS)
- Nur das ANALYZE-Kommando berücksichtigt die Storage-Beiträge von LONG- und LONG-RAW-Spalten. Die Benutzung von DBMS_STATS kann daher zu erheblichen Überbewertungen des Einsparungspotenzials führen, da gerade diese zumeist Speicherplatz-intensiven Daten vollkommen ausgeblendet werden, also wie Freiplatz erscheinen. Ursache hierfür ist der Oracle-Bug 1954150, der sich durch alle Versionen seit Oracle 8i hindurchzieht

Es ist zu berücksichtigen, dass der vorhandene Freiplatz in Tabellen und Indizes nicht zu 100 Prozent freigegeben werden sollte, da sonst bei zukünftigen Updates

auf vorhandene Datensätze leicht Ineffizienzen durch Row-Chaining entstehen. Der altbekannte PCTFREE-Parameter setzt hier Grenzen, das heißt, bei einem typischen Wert PCTFREE = 10 verbleibt auch nach der Reorganisation ein Freiplatzanteil von 10 Prozent pro Oracle-Block, und das ist in den allermeisten Fällen auch gut so. Die einzige Ausnahme bilden Tabellen, auf denen definitiv keine Updates mehr stattfinden und auch nie mehr Spalten hinzugefügt werden.

Fazit

Die Größe heutiger ERP-Datenbanken bewirkt, dass eine Reorganisation erhebliche finanzielle Vorteile für Endkunden bringen kann. Durch die Vielzahl verfügbarer Reorganisations-Methoden mit ihren unterschiedlichen Schwerpunkten, die aber allesamt mit minimalen Auszeiten auskommen, ist eine Reorganisation eben auch auf kritischen, hochverfügbaren Systemen verhältnismäßig einfach möglich.

Kontakt:

Patrick Schwanke
patrick.schwanke@quest.com



The advertisement features a vertical yellow bar on the left with the text "JOIN OUR TEAM!". The main content is on a white background with a red border. At the top right is the AMIS logo, with "Business Consultancy", "Application Development", "Project Management", and "DB Operation & Administration" listed below it. The central text reads: "Ihr leistungsstarker Dienstleistungspartner für Individuallösungen auf der Basis modernster Oracle- und JAVA-Technologie." followed by "Es geht um Ihr Tagesgeschäft und Ihre Prozesse!". Below this is the question "Sie sind Profi, Insider? ... und kennen diesen Weblog nicht?" and the URL "www.amis-de.com/blog". At the bottom left, contact information for AMIS Deutschland GmbH is provided: "Innungstraße 5, 50354 Hürth, www.amis-de.com, +49 (0) 22 33 - 61 19 20". At the bottom right is a photo of a smiling man. The footer contains the text "Committed to ICT. Involved in People." and the "ORACLE PARTNER" logo.

Datenbank 11g macht den FULL OUTER JOIN tauglich für das Data Warehouse

Autor: Karol Hajdu, Trivadis AG

Den FULL OUTER JOIN gibt es in Oracle Database seit der Version 9.0. Doch erst mit der Version 11g steht seinem Einsatz in der Data-Warehouse-Praxis nichts mehr im Wege.

Da die meisten Data Warehouses eine Versionierung der Stammdaten implementiert haben, und obwohl diese Versionierung mit einem FULL OUTER JOIN elegant umsetzbar ist, findet dieses Konstrukt in der Praxis keine breite Anwendung. Der Grund liegt darin, dass die Oracle-interne Ausführung eines FULL OUTER JOINS äußerst ineffizient ist. So ist der praktische Einsatz bei großen Tabellen so gut wie unmöglich. Oracle Database 11g bietet eine neue interne Implementation, die diesen Nachteil beseitigt.

Die in diesem Artikel beschriebenen Konzepte werden anhand des folgenden Beispiels erklärt: In der Staging-Area des Data Warehouses liegt die Tabelle SA_CUSTOMER. Die Spalte OLTPKey bildet den Primärschlüssel der Tabelle. Zusätzlich gibt es noch eine inhaltliche Spalte namens Status, die den Status des Kunden abbildet. Des Weiteren existiert im CORE des Data Warehouses eine Tabelle C_CUSTOMER mit der Spalte SurrogateKey als Primärschlüssel. Neben der inhaltlichen Spalte Status bestehen noch die beiden versionierungstechnischen Felder C_ValidFrom und C_ValidTo. Abbildung 1 zeigt die Tabellenstruktur und den Tabelleninhalt des Beispiels.

Der FULL OUTER JOIN

Streng genommen ist der FULL OUTER JOIN ein Mengen-Operator der relationalen Algebra. Die meisten

Table SA_CUSTOMER	
OLTPKey	Status
A	active
B	active
C	inactive

C_CUSTOMER				
Surrogate	OLTPKey	C_ValidFrom	C_ValidTo	Status
4711	A	20.11.2007	02.01.2008	inactive
5111	A	02.01.2008	01.01.9999	active
6128	B	21.01.2008	01.01.9999	inactive
734	D	21.01.2008	01.01.9999	active

Abbildung 1: Tabellen SA_CUSTOMER und C_CUSTOMER

Anbieter der relationalen Datenbank-Technologien haben ihn in den letzten zehn Jahren in ihre Datenbank-Server integriert. Im Gegensatz zu einem INNER JOIN liefert ein FULL OUTER JOIN nicht nur solche Rows aus beiden Tabellen zurück, die die gleichen Werte-Kombinationen für die Join-Spalten ausweisen, sondern auch alle restlichen Rows aus beiden Tabellen, in denen keine Übereinstimmung gefunden werden konnte (siehe Abbildung 2).

Mit einem FULL OUTER JOIN kann man zum Beispiel die Versionierung der Stammdaten im Data Warehouse elegant umsetzen. Bei der Versionierung geht es darum, die Änderungen seit dem letzten Datenabgleich zu erkennen und dieses ermittelte Delta in geeigneter Form in dem CORE Data Warehouse abzulegen.

Table SA_CUSTOMER		Table C_CUSTOMER				
OLTPKey	Status	Surrogate	OLTPKey	C_ValidFrom	C_ValidTo	Status
NULL	NULL	734	D	21.01.2008	01.01.9999	active
A	active	4711	A	20.11.2007	02.01.2008	inactive
A	active	5111	A	02.01.2008	01.01.9999	active
B	active	6128	B	21.01.2008	01.01.9999	inactive
C	inactive	NULL	NULL	NULL	NULL	NULL


```

SELECT SA.*, C.*
FROM SA_CUSTOMER SA
FULL OUTER JOIN C_CUSTOMER C
ON SA.OLTPkey = C.OLTPkey ;
    
```

Abbildung 2: Unterschied zwischen INNER JOIN und FULL OUTER JOIN

Mit einem FULL OUTER JOIN wird die Delta-Erkennung sehr einfach und lesbar umgesetzt (siehe Listing 1)

```
CREATE TABLE H_CUSTOMER_DELTA
AS
SELECT *
FROM (SELECT C.SurrogateKey
, SA.OLTPkey
, SA.Status
,CASE WHEN SA.OLTPkey IS NULL
THEN 1 ELSE 0 END AS DML_D
,CASE WHEN C.OLTPkey IS NULL
THEN 1 ELSE 0 END AS DML_I
,CASE WHEN SA.OLTPkey IS NOT NULL
AND C.OLTPkey IS NOT NULL
AND (SA.STATUS <> C.STATUS)
--Status is mandatory
THEN 1 ELSE 0 END DML_U
FROM (SELECT * FROM C_CUSTOMER
WHERE C_ValidTo = to_date(,01.01.9999',
,dd.mm.yyyy')) C
FULL OUTER JOIN SA_CUSTOMER SA
ON SA.OLTPkey = C.OLTPkey
)
WHERE DML_I+DML_U+DML_D > 0 ;
```

Listing 1: Delta-Erkennung mit Einsatz von FULL OUTER JOIN

```
SQL> SELECT * FROM H_CUSTOMER_DELTA;
SURROGA
TEKEY OLTPKEY STATUS DML_D DML_I DML_U
-----
6128 B active 0 0 1
C inactive 0 1 0
734 1 0 0
```

Listing 2: Resultat der Delta-Erkennung anhand der Beispieldaten

Der FULL OUTER JOIN war bis zur Version 10g praktisch nicht einsetzbar. Den Grund kann man relativ einfach finden, indem man sich den Execution Plan für ein Statement mit FULL OUTER JOIN im Listing 3 anschaut.

```
SQL> select * from v$version ;
BANNER
-----
Oracle Database 10g Enterprise Edition
Release 10.2.0.3.0 - Prod
PL/SQL Release 10.2.0.3.0 - Production
CORE 10.2.0.3.0 Production
TNS for Linux: Version 10.2.0.3.0 - Production
```

```
NLSRTL Version 10.2.0.3.0 - Production
SQL> SELECT C.SurrogateKey
2 , SA.OLTPkey
3 , SA.Status
4 ,CASE WHEN SA.OLTPkey IS NULL
5 THEN 1 ELSE 0 END AS
DML_D
6 ,CASE WHEN C.OLTPkey IS NULL
7 THEN 1 ELSE 0 END AS
DML_I
8 ,CASE WHEN SA.OLTPkey IS NOT
NULL
9 AND C.OLTPkey IS NOT NULL
10 AND (SA.STATUS <>
C.STATUS) --Status is mandatory
11 THEN 1 ELSE 0 END DML_U
12 FROM (SELECT * FROM C_CUSTOMER
13 WHERE C_ValidTo = to_
date(,01.01.9999', ,dd.mm.yyyy')) C
14 FULL OUTER JOIN SA_CUSTOMER SA
15 ON SA.OLTPkey = C.OLTPkey;
```

Execution Plan

Plan hash value: 3478700539

```
-----
| Id | Operation | Name |
|-----|-----|-----|
| 0 | SELECT STATEMENT | |
| 1 | VIEW | |
| 2 | UNION-ALL | |
| * 3 | HASH JOIN OUTER | |
| * 4 | TABLE ACCESS FULL | C_CUSTOMER |
| 5 | TABLE ACCESS FULL | SA_CUSTOMER |
| * 6 | HASH JOIN ANTI | |
```

Listing 3: Execution Plan für ein FULL OUTER JOIN in Oracle 9i/10g

Man sieht, dass jede der Tabellen zweimal gelesen wird. Solange die Tabellen klein sind, ist dies zu vernachlässigen. Handelt es sich jedoch um große Tabellen, komplexe oder aufwändige Joins, Groupbys oder Sorts, die den Input für den FULL OUTER JOIN aufbereiten, so ist das Resultat verheerend. Eine Verdoppelung der Arbeit hat dann einen fatalen Einfluss auf die Verarbeitungszeit. Dies trifft vor allen für Data-Warehouse-Problemmstellungen zu und erklärt somit, warum ein FULL OUTER JOIN in Oracle 9i/10g hier nur sehr selten eine

Arbeiten mit großen Datenbanken

praktische Anwendung fand. Ein Blick in den Execution Plan zeigt die Verbesserungen in Oracle 11g am FULL OUTER JOIN (siehe Listing 4).

```
SQL> select * from v$version
BANNER
-----
Oracle Database 11g Enterprise Edition
Release 11.1.0.6.0 - Production
PL/SQL Release 11.1.0.6.0 - Production
CORE      11.1.0.6.0      Production
TNS for Linux: Version 11.1.0.6.0 - Pro-
duction
NLSRTL Version 11.1.0.6.0 - Production

SQL> SELECT   C.SurrogateKey
 2  , SA.OLTPkey
 3  , SA.Status
 4  ,CASE WHEN SA.OLTPkey IS NULL
 5             THEN 1 ELSE 0 END AS DML_D
 6  ,CASE WHEN C.OLTPkey IS NULL
 7             THEN 1 ELSE 0 END AS DML_I
 8  ,CASE WHEN SA.OLTPkey IS NOT NULL
 9             AND C.OLTPkey IS NOT NULL
10             AND (SA.STATUS <>
C.STATUS) --Status is mandatory
11             THEN 1 ELSE 0 END DML_U
12 FROM (SELECT * FROM C_CUSTOMER
13        WHERE C_ValidTo = to_
date(,01.01.9999', ,dd.mm.yyyy')) C
14 FULL OUTER JOIN SA_CUSTOMER SA
15 ON      SA.OLTPkey = C.OLTPkey;
```

Execution Plan

Plan hash value: 1804996898

Id	Operation	Name
0	SELECT STATEMENT	
1	VIEW	VW_FOJ_0
* 2	HASH JOIN FULL OUTER	
3	VIEW	
* 4	TABLE ACCESS FULL	C_CUSTOMER
5	TABLE ACCESS FULL	SA_CUSTOMER

Predicate Information
(identified by operation id):

```
2 - access(„SA“.„OLTPKEY“=„C“.„OLTPKEY“)
4 - filter(„C_VALIDTO“=TO_DATE(, 9999-01-
01 00:00:00', ,yyyy-mm-dd hh24:mi:ss'))
```

Listing 4: Execution Plan für ein FULL OUTER JOIN in Oracle Database 11g

Die gute Nachricht

Oracle hat eine neue Row-Source-Operation implementiert. Sie heißt HASH JOIN FULL OUTER und liest die beiden Input-Tabellen nur einmal. Der praktische Einsatz in Data Warehouses ist daher mit Oracle 11g möglich. Doch wozu dienen die zwei Views – eine mit Namen VW_FOJ_0 und die andere ohne Namen? Diese Views sind nur temporär und im kompilierten Format vorhanden. Vermutlich dienen sie folgendem Zweck: Der Cost Based Optimizer von Oracle verwendet Predicate-Pushing und Subquery-Merging, um einen optimalen Execution-Plan zu finden, der die Daten-Einschränkungen (Filter) möglichst früh appliziert. Für einen FULL OUTER JOIN ist charakteristisch, dass diese Daten-Einschränkungen genau über der Row-Source appliziert werden, die im SELECT aufgeführt ist.

Eine frühere oder spätere Applizierung der Filterbedingung, die im Falle vom INNER JOIN die Semantik der Abfrage nicht ändert, hätte im Falle von OUTER JOIN fatale Folgen: Der Execution Plan wäre vielleicht optimal, aber er würde falsche Resultate liefern (Issues bekannt mit Oracle Database 9i Version kleiner als 9.2.0.4). Würde beispielsweise der Filter „C_ValidTo = to_date(,01.01.9999', ,dd.mm.yyyy)“ aus Listing 1 erst nach dem FULL OUTER JOIN appliziert, dann liefert die Abfrage komplett andere Resultate. Findet diese Verschiebung nicht auf Anlass des Benutzers, sondern aufgrund der Optimierung statt (also aus Benutzersicht unbewusst), sind diese Resultate falsch. Hier kommt die Aufgabe der Views zum Tragen: Die Views enthalten vermutlich solche Hints, die dem Optimizer mitteilen, kein Predicate-Pushing oder Subquery-Merging anwenden zu dürfen.

Kontakt:

Karol Hajdu
karol.hajdu@trivadis.com

Weitere Informationen zu den Themen der DOAG News finden Sie unter <http://www.doag.org/>

Oracles HARD-Initiative

Autor: Peter Heumel, HVB Information Services

Block Corruptions in Datenbank-Files nicht nur zu erkennen, sondern auch zu verhindern, indem intelligente Disk-Subsysteme die Datenbank-Blöcke vor dem Schreiben prüfen – das ist das Ziel von Oracle bei der gemeinsam mit Platten-Herstellern initiierten HARD-Initiative (Hardware Assisted Resilient Data).

Werden Oracle-Datenbanken in Rechenzentren betrieben, so liegen die zur Datenbank gehörenden Dateien häufig gespiegelt auf zwei Platten-Boxen an unterschiedlichen Standorten. Dieser sinnvolle Schutz vor Media-Fehlern und Standort-Ausfällen nützt aber nichts, wenn die Oracle-Blöcke bereits korrupt am Plattensystem ankommen. Die Ursachen für die Korruptionen sind mannigfaltig und meist – aber nicht notwendigerweise – hardwarebedingt (beispielsweise eine defekte Fibre-Channel-Karte). Fehler treten sehr selten auf, sind dann aber sehr ärgerlich. Innerhalb der letzten zehn Jahre war es im Unternehmen des Autors immerhin einmal nötig, eine Datenbank in TB-Größe komplett zu restoren und recovern, weil durch eine Spannungsschwankung im Datenbank-Server Teile der Online Redo-Logs ausgenullt worden waren.

Eine Möglichkeit zur Abhilfe für diese Art von Fehlern ist natürlich eine Standby-Datenbank-Lösung. Allerdings ist diese für große Datenbanken wie Data Warehouses sehr teuer, da man mehrere TB Plattenplatz zusätzlich benötigt, der in Rechenzentren nicht gerade billig ist. Eine andere Möglichkeit ergibt sich aufgrund der HARD-Initiative (Hardware Assisted Resilient Data) von Oracle.

Die Idee ist sehr einfach: Oracle-Blöcke werden vor dem Wegschreiben (etwa durch den DBWR) mit einer Checksumme versehen. Das Plattensystem prüft diese Checksumme, bevor die Daten tatsächlich auf Platte geschrieben werden. Ist sie falsch, kann das Disk-Subsystem das I/O abweisen. Somit wird verhindert, dass fehlerhafte Blöcke überhaupt erst geschrieben werden. Das Datenbank-System bekommt für den System-Call (pwrite, aiowrite unter Unix) einen I/O-Fehler gemeldet und kann das I/O wiederholen.

Möglich wird dies durch die HARD-Initiative, in deren Rahmen Oracle den Plattensystemherstellern Validierungsalgorithmen für Oracle-Blöcke offengelegt hat. Weitere Festlegungen, wie I/O-Calls, besagten, diese immer nur als Vielfaches der jeweiligen Blockgröße zu verwenden.

Die Software für die Prüfung läuft auf dem Disk-Subsystem und heißt je nach Hersteller des Plattensystems unterschiedlich. Die einzelnen Produkte unterscheiden sich, abgesehen von der Abhängigkeit

von bestimmten Plattenboxen, auch darin, welche der HARD-Validierungstests implementiert wurden. Denn der bereits erwähnte Test zu den Block-Checksummen ist nur einer von mehreren. Eine Übersicht der einzelnen Hersteller und ihrer Produkte steht unter www.oracle.com/technology/deploy/availability/htdocs/HARD.html. Drei der momentan sieben HARD-Tests sind:

- Die Prüfung der Oracle-Block-Checksumme
- Die Prüfung der Database-Block-Adresse (DBA)
- Die Prüfung der I/O-Größe

Prüfung der Block-Checksumme

Um ein Missverständnis auszuräumen: Die HARD-Initiative war nicht die Ursache dafür, die Oracle Block-Checksumme einzuführen. Die von Oracle im September 2001 ins Leben gerufene Initiative hat sich vielmehr die damals bereits vorhandenen Block-Checksummen zunutze gemacht. Eingeführt wurden sie bereits mit der Version 7.2 für Daten- und Online-Redoblöcke. Damals noch zwei Parameter – die Checksummen konnten getrennt für Daten- und Online-Redoblöcke aktiviert werden –, wird heute durch einen einzigen Parameter (DB_BLOCK_CHECKSUM) aktiviert, der defaultmäßig angeschaltet ist. Dieser Parameter bewirkt zweierlei, falls er auf TRUE (TYPICAL in Oracle 10g) gesetzt ist:

- Vor dem Wegschreiben auf Platte wird eine 16 Bit lange Checksumme generiert und in den Block gesetzt. Zusätzlich wird ein Bitflag gesetzt, dass für diesen Block die Checksumme berechnet wurde.
- Wird ein Oracle-Block wieder von Platte gelesen, so prüft die Oracle-Instanz, ob für diesen Block das Bitflag gesetzt ist. Ist dies der Fall, wird die Checksumme geprüft.

Abweichend davon wird in HARD-Implementierungen am Platten-Subsystem die Checksummen-Generierung vorausgesetzt und das Flag nicht geprüft.

Die Kontrollabbildung ist denkbar einfach: Ein XOR in 16-Bit-Schritten über alle Daten des Datenblocks, das heißt, die ersten 16 Bit des Blockes XOR mit den zwei-

Arbeiten mit großen Datenbanken

ten 16 Bit XOR mit den dritten 16 Bit usw. muss 0x00 (bei 16 Bit 0000000000000000) ergeben.

Wo die Checksumme im Block enthalten ist, ist für die Prüfung (Kontrollabbildung) irrelevant. Im Block Dump für Data-Files wird die Checksumme übrigens explizit ausgewiesen (chkval):

```
alter system dump datafile 5 block 18;
Start dump data blocks tsn: 3 file#: 5
minblk 18 maxblk 18
buffer tsn: 3 rdba: 0x01400012 (5/18)
scn: 0x0000.000696e6 seq: 0x01 flg: 0x06
tail: 0x96e60601
frmt: 0x02 chkval: 0x94a9 type:
0x06=trans data
```

Die Checksumme ist hier 0x94a9 und das Bitflag ist gesetzt (in flg: 0x06 enthalten).

Auf jeden Fall berechnet Oracle die Checksumme und setzt sie in den im Hauptspeicher befindlichen Block, bevor beispielsweise der DBWR den Block wegschreibt. Wenn nun unterwegs zum Platten-System etwas passiert – der I/O Stack hat oft viele Schichten und Komponenten – dann ist das Plattensystem mit einer HARD-Implementierung in der Lage, den Fehler zu erkennen. Der Controller des Platten-Systems kann die Kontrollabbildung (16 Bit breites XOR) ausführen und das I/O zurückweisen, falls die Checksumme nicht stimmt. Die zusätzliche Prüfung vor dem Schreiben auf dem Platten-Subsystem, die zur Prüfung durch die Datenbank Instanz beim Wiedereinlesen von Platte hinzukommt, macht nun HARD aus.

Um dies zu bewerkstelligen, benötigt das Plattensystem allerdings Metadaten, es muss wissen, welche Bereiche auf den physikalischen Platten zur Datenbank gehören, welche Oracle-Blockgröße die dort abgelegten Daten eines Tablespace haben und ob es sich um Daten-/Redolog- oder Control-Files handelt. Folglich benötigen HARD-Produkte der Plattenhersteller eine Software, das aus dem Data Dictionary der Oracle Datenbank die benötigte Information ausliest und in das Repository des Platten-Systems bringt. Ähnlich einem RMAN-Resync muss man bei Struktur-Änderungen in der Datenbank (neue Tablespace, zusätzliches Raw Device für einen Tablespace etc.) einen solchen Resync mit dem Platten-System ausführen, damit die neuen Metadaten übertragen werden und somit auch die Data-Files eines neuen Tablespaces überwacht werden.

Dadurch, dass die Metadaten im Repository des Plattensystems liegen, sind Data-, Redolog- und Control-Files der Datenbank auch unabhängig von der Datenbank-Instanz und dem Datenbank-Server geschützt, wie folgender Test mit Unix dd zeigt:

```
dd if=block_18.blk of=/dev/vx/rdisk/raw5
bs=8192 seek=18 count=1 conv=notrunc
write: I/O error
1+0 records in
1+0 records out
```

Man kann versuchen, in ein durch HARD-Software geschütztes Daten-File, einen Datenblock (block_18.blk) mit korrupter Checksumme zu schreiben. Im Truss sieht man:

```
write(4, „0602\0\001 @\012\00696E6“..,
8192)
Err#5 EIO
```

Die Datenbank-Files sind also vor jeglichem schreiben dem I/O mit falschen Checksummen geschützt – unabhängig von der Oracle-Software. Nun sind die 16 Bit langen Checksummen kein besonders starker Schutz (nur ein einziger bis zu 16 Bit langer Fehler wird mit Sicherheit erkannt), aber hier helfen weitere Prüfungen.

Wie bereits erwähnt, stützt sich ein weiterer HARD-Test auf die vier Byte lange Database-Block-Adresse (DBA), die in jedem Datenblock enthalten ist. Im Block Dump oben: rdba: 0x01400012. Die ersten 10 Bits geben die Filenummer (hier 0000 0001 01, = 5), die restlichen 22 Bits die Blocknummer an (= 18). Das Plattensystem kann nun für jeden I/O prüfen, ob die Daten auch für die richtige Stelle im jeweiligen Data-File sind. Falls die Database-Block-Adresse nicht zum Offset des I/O-Calls passt (pwrite, aiowrite), wird das I/O zurückgewiesen.

Der dritte HARD-Test bezieht sich auf die Größe des I/Os. Oracle hat sich im Rahmen der HARD-Initiative darauf festgelegt, dass jeglicher I/O immer als Vielfaches der jeweiligen Blockgröße eines Data- oder Redo-Log-Files abgesetzt wird (beim Aufruf der System-Calls für das Schreiben). Hat ein Tablespace eine Block-Size von 8 KB, so wird durch den dritten HARD-Test ein I/O von beispielsweise 10 KB abgewiesen.

Fazit

Allein diese drei HARD-Tests sind bereits sehr mächtig und bieten einen guten Schutz für Daten-, Redo- und Control-Files. Für die Data Warehouses mit TB-Größe setzt das Unternehmen des Autors seit mehreren Jahren eine HARD-Implementierung auf den Platten-Subsystemen ein. Ein Ausnullen von Teilen der Online-Redologs, wie sie es einmal erleben mussten, ist dadurch nicht mehr möglich.

Kontakt:

Peter Heumel
peter.heumel@hvbis.com

Langzeitspeicherung und Trendanalyse von Statistik- und Performance-Daten mit einem RRD-Tool

Autor: Félix Castillo Sánchez

Bei der Verwaltung großer Datenbestände ist eine Trendanalyse oft Grundlage für die Planung weiterer Ressourcen. Mithilfe des frei verfügbaren RRD-Tools lassen sich nicht nur Kapazitätsinformationen, sondern auch Performance-Werte langfristig speichern und grafisch darstellen.

RRD ist ein Akronym für Round Robin Database. Es beschreibt ein Konzept, wie über einen langen Zeitraum gespeicherte Daten effizient archiviert werden können. Ziel ist es, die Datenmenge ohne wesentlichen Informationsverlust auf ein vernünftiges Maß zu reduzieren. Das Open-Source-Programm rrdtool von Tobias Oetiker [1] implementiert genau dieses Konzept und bietet zudem eine sehr leistungsstarke Schnittstelle, um sowohl aus den gespeicherten als auch aus berechneten Daten Graphen erzeugen zu können.

rrdtool speichert in regelmäßigen Abständen die übergebenen Werte. Nach definierbaren Regeln werden diese in nachgeordnete Archive (Round Robin Archives – RRA) übertragen, indem Mindest-, Maximal- und/oder Durchschnittswerte eines Zeitraums berechnet werden. Durch diese Zusammenfassung entstehen komprimierte Datenreihen, die Größe der RRD-Datei bleibt genauso konstant wie die Zugriffszeit.

Grafische Darstellung der Daten

Das Speichern der Daten mit rrdtool ist die Grundlage für die Erstellung von Graphen, die eine Interpretation der Datenreihen ermöglichen. Abbildung 2 zeigt beispielhaft die Speicherplatz-Auslastung eines Tablespace im Zusammenhang mit der durchschnittlichen Größe der darin gespeicherten „Akten“ über einen Zeitraum von rund zwei Jahren.

Zudem ist der voraussichtliche Zeitpunkt zu erkennen, an dem die stetig wachsende Speicherauslastung den vorhandenen Speicherplatz übersteigen wird.

Eine weitere interessante Anwendung ist die Darstellung Oracle-spezifischer Performance-Werte wie aus der `v$sysstat` oder `v$system_event`. Abbildung 3 zeigt das Verhältnis zwischen logical I/Os und physischen Plattenzugriffen über einen Zeitraum von drei Wochen.

Durch die grafische Darstellung der Werte ist die rhythmische Belastung des Rechners ersichtlich. Abbil-

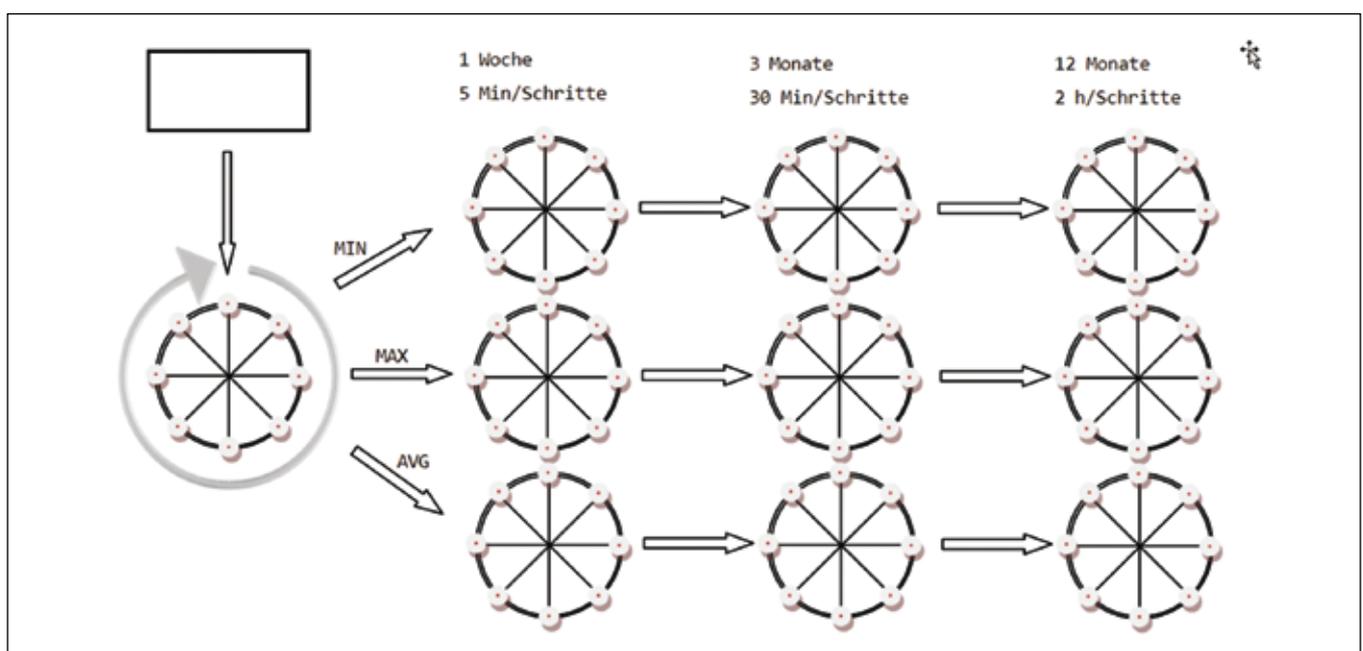


Abbildung 1: Funktionsweise von rrdtool

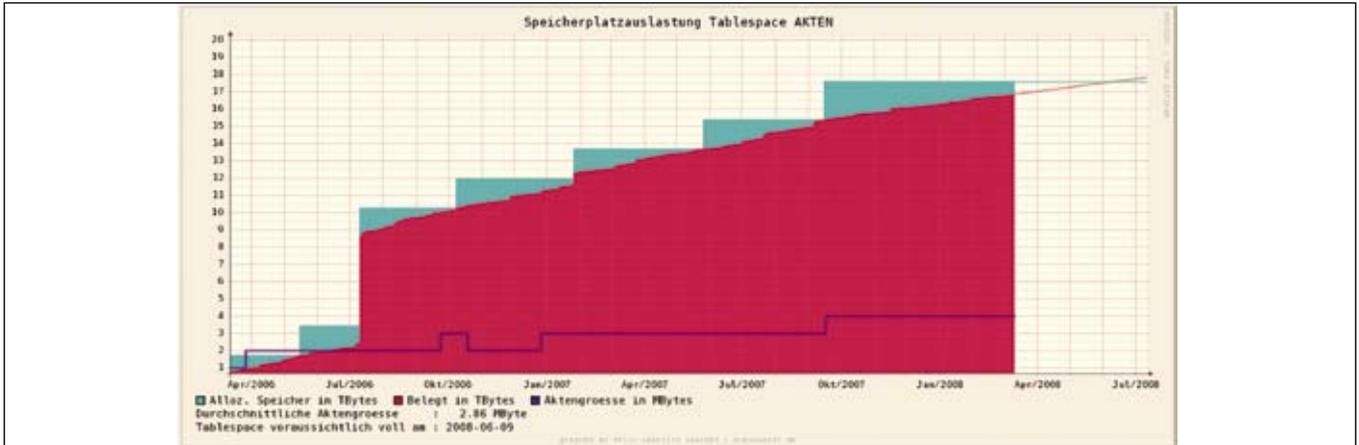


Abbildung 2: Grafische Darstellung der Speicherplatzbelegung und der durchschnittlichen Aktengröße

Abbildung 4 zeigt die aus der View `v$system_event` bekannten Werte `average_wait` und `time_waited` zusammengefasst pro Wait-Group.

Mit den gezeigten Grafiken lässt sich nachweisen, dass trotz erhöhter Wartezeit die durchschnittliche Wartezeit stabil bleibt, das heißt, dass der Rechner mit der Last auskommt und es keine Engpässe gibt.

Beispiel

Erzeugung der RRD-Datenbank:

```
rrdtool create db_alloc.rrd --step 86400 \
  DS:alloc:GAUGE:86400:0:NaN \
  DS:used:GAUGE:86400:0:NaN \
  DS:avgsz:GAUGE:86400:0:NaN \
  RRA:AVERAGE:0.5:1:730
```

`rrdtool` legt eine RRD-Datenbank `db_alloc.rrd` an. Das Aktualisierungsintervall beträgt einen Tag ($24 \cdot 60 \cdot 60$). Es wird ein Archiv angelegt, das 730 Werte (rund zwei Jahre) tagessgenau speichert.

Update der Datenbank:

Der Update-Befehl für `rrdtool` sieht sehr einfach aus: `rrdtool update N:20000:17163:4,15`.

Er speichert die Werte in der Reihenfolge der beim Create angelegten Spalten. Der Wert `N` bestimmt den aktuellen Zeitpunkt.

Erstellung einer einfachen Grafik:

Um eine Grafik mit den gespeicherten Werten zeichnen zu können, sind folgende Befehle notwendig:

```
rrdtool graph db_alloc_simple.png \
  --height 150 --width 650 \
  --start -2y \
  --title „Speicherplatzauslastung Tablespace AKTEN“ \
  DEF:alloc=db_alloc.
rrd:alloc:AVERAGE:start=-2y \
  DEF:used=db_alloc.
rrd:used:AVERAGE:start=-2y \
  DEF:avgsz=db_alloc.
rrd:avgsz:AVERAGE:start=-2y \
  CDEF:allocated=alloc,1024,/ \
  CDEF:used_space=used,1024,/ \
  AREA:allocated#48D1CC:„Alloz. Speicher in TBytes“ \
  AREA:used_space#DC143C:„Belegt in TBytes“ \
  LINE2:avgsz#480082:„Aktengröße in MBytes\n“
```

Die direkte Anwendung von `rrdtool` ist eine aufwändige Angelegenheit, daher ist die Anwendung eines Frontends erforderlich. Es existieren mehrere Tools auf dem Markt. Leider sind diese entweder nicht speziell für den Einsatz zur Speicherung Oracle-spezifischer Daten vorge-

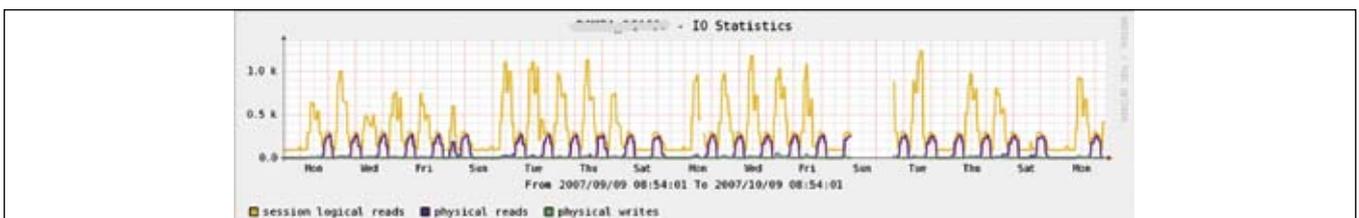


Abbildung 3: Gegenüberstellung der logischen IO-Operationen und physischen Plattenzugriffe

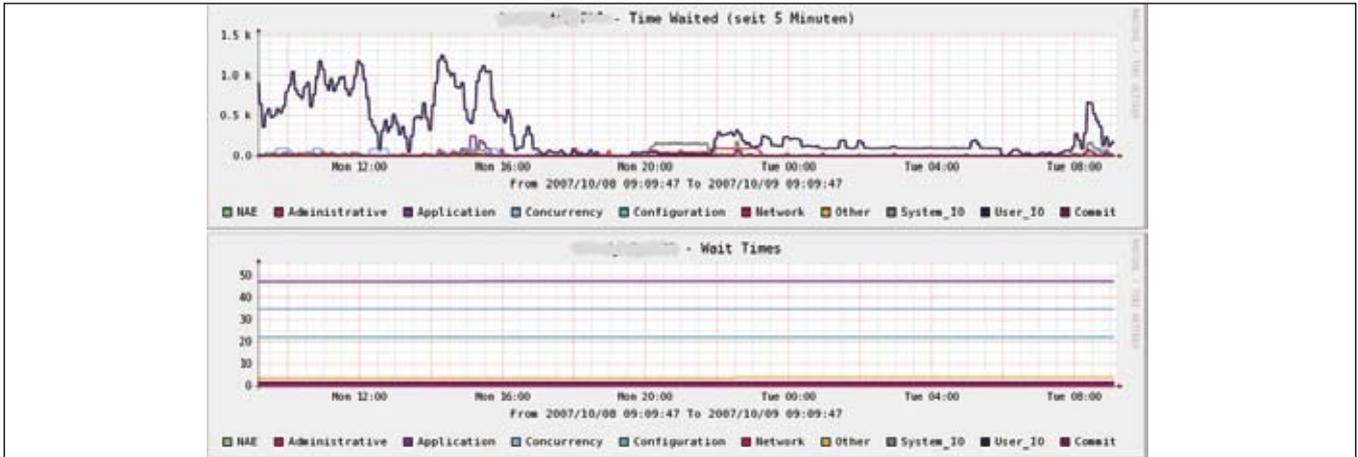


Abbildung 4: Wartezeiten und gewartete Zeit

sehen oder sie werden seit Jahren nicht mehr gepflegt. CACTI [2] bietet als eines der verbreiteten Frontends die Möglichkeit, Oracle-Datenbanken einzubinden. Nachteil ist, dass dies auf der Erstellung von Custom-Scripts basiert, die bei jedem Aufruf eine neue Verbindung zur Datenbank-Instanz aufbauen müssen. Außerdem ist eine automatisierte Administration der von CACTI überwachten Systeme nicht ohne Weiteres möglich, da es sich um ein reines Web-Frontend handelt.

Weitere Informationen

- [1] <http://oss.oetiker.ch/rrdtool/index.en.html>
- [2] CACTI – The complete rrdtool-based graphing solution, www.cacti.net

Kontakt:

Félix Castillo Sánchez
felix.castillo@oraconsult.de

DACHCOM



www.trivadis.com

Das Unternehmen
Trivadis ist ein erfolgreicher, expandierender und unabhängiger Anbieter von IT Lösungen und Services. Trivadis ist für mehr als 600 Kunden an 12 Standorten in Deutschland, in Österreich und in der Schweiz tätig. Das Dienstleistungsportfolio umfasst Anwendungsentwicklung, Consulting, Systemmanagement, Projektmanagement und Schulung. Trivadis vereint exzellentes technologisches Know-how in den Basistechnologien Oracle, Microsoft, IBM und Open Source mit einer breiten Lösungskompetenz.

Was wir Ihnen bieten
Trivadis bietet Ihnen ein vielseitiges Aufgabengebiet, viel Freiraum für die Umsetzung eigener Ideen, gute Weiterbildungs- und Entwicklungsmöglichkeiten sowie ein spannendes Know-how-Netzwerk, in dem Sie Ihr Wissen mit anderen Spezialisten austauschen können. Attraktive Anstellungsbedingungen, moderne Arbeitsmittel und ein leistungsorientiertes Einkommen runden unser Angebot ab.

Interessiert?
Dann freuen wir uns auf Ihre Bewerbungsunterlagen inkl. IT-Skill-Liste an jobs@trivadis.com. Telefonische Auskünfte erteilt Ihnen gerne Frau Janine Käser oder Doris Paul, Human Resources, Tel. 069 264 933 031.

Trivadis GmbH · Human Resources · Täferstr. 5 · CH-5405 Baden-Dättwil

Application Development
Application Performance Management
Business Communication
Business Intelligence
Managed Services
Security
Training

Trivadis beschäftigt bereits über 500 Mitarbeiter und will weiter wachsen. Zur Verstärkung unseres Teams in **Hamburg** suchen wir per sofort oder nach Vereinbarung eine initiative Persönlichkeit als:

■ Gruppenleiter (m/w)

Ihre Hauptaufgaben sind:

- Account Management / Vertrieb
 - Akquisition neuer Kundenprojekte
 - Betreuung der bestehenden Kunden
- Bereichsverantwortung
 - Umsatz-, Auftrags- und Delivery-Verantwortung
- Mitarbeiterführung
 - Führung von bis zu 15 Mitarbeitern
 - Selektion neuer Mitarbeiter
 - Mitarbeiterentwicklung
 - Einsatzplanung

Was Sie mitbringen sollten:

- Mehrjährige Berufs- und Führungserfahrung in vergleichbarer Position
- Erfolgsnachweis in der Auftragsakquisition (Consulting/Projektdienstleistung) und umfassendes Kundennetzwerk
- Gute Kenntnisse in der IT-Dienstleistung im norddeutschen Raum
- Technischer Background im Bereich der Oracle Core Technology oder in der Applikationsentwicklung
- Teamplayer, Koordinations- und Kommunikationsstärke
- Sehr gute Deutsch- und gute Englischkenntnisse

trivadis

makes IT easier. ■ ■ ■

Basel · Baden · Bern · Lausanne · Zürich · Düsseldorf · Frankfurt/M. · Freiburg i.Br. · Hamburg · München · Stuttgart · Wien

Oracle 11g versus SQL Server 2008

Autor: Timo Meyer, Steinbeis-Transferzentrum

Microsoft hatte den neuen SQL Server 2008 für Februar 2008 angekündigt. Der Artikel vergleicht Systemanforderungen und ausgewählte neue Features dieses Datenbank-Management-Systems, das zur Zeit in der Beta-Version (Community Technology Preview, CTP) getestet werden kann, mit der neuen Datenbank-Version 11g von Oracle.

Um erste Vergleichsmöglichkeiten zwischen SQL Server 2008 und Oracle 11g zu erhalten, werden zunächst in Tabelle 1 die unterschiedlichen Systemanforderungen betrachtet.

Components	Oracle 11g	SQL Server 2008
Prozessor	Minimum: 550 MHz (Windows Vista, 800 MHz) Empfohlen: Pentium IV mit 2 GHz oder höher	Minimum: 1 GHz (x86 Prozessor) oder 1.4 GHz (x64 Prozessor)
Arbeitsspeicher	Minimum: 1 GByte RAM und die doppelte Menge an Virtual Memory Empfohlen: 2 GByte RAM Optimal: 8 GByte RAM (Server Core installation) oder mehr	Minimum: 512 MByte RAM empfohlen: 2 GByte RAM oder mehr Optimal: 2 GByte RAM (Full Installation) oder 1GByte RAM (Server Core Installation) oder mehr Maximum (32-bit Systeme): 4 GByte (Standard) oder 64GByte (Enterprise and Datacenter) Maximum (64-bit Systeme): 32 GByte (Standard) oder 2TB (Enterprise, Datacenter und Itanium-basierte Systeme)
Festplattenspeicher	Basic Installation: 4,55 GByte Advanced Installation: 4,92 GByte	Minimum: 10 GByte Empfohlen: 40GByte oder mehr.
Laufwerk	Internet Download	DVD-ROM
Display	Minimum: 256 Farben Empfohlen: 65536 Farben, 1024 x 768 Auflösung.	Super VGA (800 x 600) oder mehr

Tabelle 1: Systemanforderungen von Oracle 11g und SQL Server 2008

Preislich wird der SQL Server 2008 bei rund 6.000 US-Dollar liegen und die sechs Module Database Engine, Integration Services (ETL), Reporting Services, Analysis Services (Data Mining und Multidimensional Data), Service Broker und Replication umfassen. Die Oracle Enterprise Edition kostet 40.000 US-Dollar pro Prozessor bzw. 800 US-Dollar pro Benutzer, die vier neuen selbst-adaptiven Module Real Application Testing, Advanced Compression, Total Recall und Active Data Guard können bei Bedarf einzeln hinzu gekauft werden.

Neue Features

Der SQL Server 2008 und Oracle 11g stellen innovative Module zur Verfügung, womit viele Prozesse vereinfacht, verbessert und automatisiert werden können.

Tabelle 2 stellt ausgewählte Neuerungen der beiden Datenbanken vor.

Features	Oracle 11g 	SQL Server 2008 
Sicherheit / Monitoring	Enhanced Password (Passwortverschlüsselung durch SHA-1-Algorithmus) Automatic Diagnostic Repository: (treten kritische Fehler auf, wird der DBA automatisch benachrichtigt)	Verschlüsselung der ganzen Datenbank (Verschlüsselung auf Festplatte, Entschlüsselung im Speicher) Externe Schlüsselverwaltung Daten-Auditing (Aufzeichnung von Struktur- und Datenänderungen)
Verfügbarkeit	Oracle 11g RAC parallel upgrades (Online-Upgrade ohne Downtime, seit 2002 von L. Ellison angekündigt)	Hot plugable CPU: CPUs können, bei Hardwareunterstützung, während der Laufzeit hinzugefügt werden
Performance	Automatic Memory Tuning (Automatische PGA Optimierung wurde in Oracle 9i eingeführt und erweitert in 10g für die SGA. In 11g kann sowohl PGA als auch SGA über einen einzigen Parameter optimiert werden) Datenkompression bis zu 2/3 weniger bei verbesserter Performance und Verfügbarkeit	Datenkompression durch Online-Aktivierung Backup-Kompression Performance-Daten-Aufzeichnung / Reporting Plan Guides: verpflichtende Ausführungspläne für SQL-Abfragen Ressource Governor (Festlegung von maximalem CPU-Verbrauch pro Nutzer / Nutzungsart)



Business Intelligence Lösungen, Oracle ADF, Forms/Java-Migration, Oracle WebCenter, der Aufbau von Informations- und Kommunikationsplattformen mit Oracle Portal, Hochverfügbarkeitslösungen mit Oracle RAC oder Oracle Online DBA Services ... TEAM bietet Ihnen als Oracle Certified Advantage Partner eine Rundum-Betreuung in allen Bereichen.

Sprechen Sie uns an, wenn es um Lizenzierung, Beratung, Entwicklung oder Schulung im Oracle-Umfeld geht. Beachten Sie bitte auch den aktuellen Schulungskatalog 2008.

Wir freuen uns auf das Gespräch mit Ihnen!



Oracle Online DBA Services von TEAM
 Optimieren Sie Ihren Datenbank-Support.
Jetzt Infos anfordern!

TEAM - Ihr Partner für innovative IT-Lösungen



TEAM in Paderborn ■ Fon 05254 / 8008-0 ■ Fax 05254 / 8008-19 ■ eMail: oracle@team-pb.de ■ www.team-pb.de

Management	Oracle 11g speichert und liest große Binärdateien innerhalb der Datenbank ebenso schnell wie das zugrunde liegende Dateisystem des Servers Spezielle Windows-Funktionen: Datenbank-Backup / Recovery unter Einsatz von Schattenkopien (Windows Server 2003), sowie Active-Directory- und Kerberos-Authentifizierung, Deinstallation mit verbesserter Registry-Säuberung	Declarative Management Framework (Richtlinienbasierte Verwaltung, ähnlich Gruppenrichtlinien unter Windows) Dynamic Management Views zum Anzeigen der Speicherauslastung des SQL Servers
Datenbank-Entwicklung	Neue Datentypen für Datei-Infos (RFID-Tags), medizinische Bilder (DICOM) sowie dreidimensionale Geometriedaten (SPATIAL)	Benutzerdefinierte Datentypen können bis zu 8 GB groß werden Geometriedaten (SPATIAL) Datenverwaltung ohne Internet Information Server (IIS)
Daten-Speicherung	FUNCTION BASED VIRTUAL COLUMN ist ein Tabellen-Link auf im Data Dictionary gespeicherte Funktionen Rückverfolgung von Flashback-Aktionen und Überprüfung auf fehlerhafte Backup-Blöcke	FILESTREAM-Datentyp (Referenz auf Datei im Dateisystem) Spezialtabellen für Änderungshistorie von Tabellen und benutzerdefinierte Tabellen für Funktionsparameter
Data Warehouse / ETL	Oracle Data Miner (GUI): Optimierte analytische Funktionen für OLAP und Data Warehouse, z.B. kopiert Data Mining Objekte in das Dictionary	Bessere Parallelisierung bei partitionierten Tabellen: keine Thread-Begrenzung für Abfragen STAR JOIN Unterstützung: automatische Erkennung durch Abfrage Optimierer Persistente Suche im SQL Server. Integration Services: keine erneute Abfrage bei Lookup Operationen, Cache Lookups Verbesserte Thread-Abarbeitung“ sowie „Verbesserung beim Backup von Cubes

Tabelle 2: Neue Features (Auswahl)

Die Microsoft Beta-Version (CTP) weckt das Interesse am neuen SQL Server. Dieser soll insgesamt 53 neue Features besitzen, welche die Performance und das Reporting der Datenbank durch neue Kompressionsroutinen, Indizieralgorithmen und intelligente Arbeitslastenverteilung erhöhen. Der SQL Server 2008 soll die letzte der 32-bit-Versionen sein, nachfolgende Versionen sind nur noch 64-bit. Oracle 11g überzeugt durch seine selbstadaptiven Module, die aus Automatisierungsalgorithmen und mehr als 450 neuen Features bestehen. Diese vereinfachen die Arbeitsabläufe des Grid Computings für Datenbank-Administratoren und Entwickler sehr.

Fazit

Häufig ist folgendes Szenario in der Praxis anzutreffen: Beide Datenbanken werden als Backend eingesetzt. Software-Anwendungen werden zumeist auf Oracle-Datenbanken entwickelt, weil es hier mehr Möglichkeiten zur Optimierung gibt. Die Auswertungen der Software-Anwendungen hingegen werden meistens über den SQL Server erstellt, da hier die niedrigsten Gesamtkosten anfallen und die Administration einfacher

ist. Die Zusammenarbeit in diesem Szenario geschieht durch Datenreplikation von Oracle zum SQL Server. Beide Datenbanken haben für die Zusammenarbeit in diesem Szenario revolutionäre Neuerungen entwickelt.

Quellen

SQL Server 2008

- <http://www.microsoft.com/germany/sql/uebersicht/futureversion/default.msp>
- <http://blogs.sqlserverfaq.de/Lists/Beitraege/Post.aspx?List=1d5de6c9%2Df560%2D4dc3%2Dbc29%2D434c3f3de2a6&ID=54>

Oracle 11g

- <http://www.oracle.com/technology/documentation/database11gR1.html>
- <http://www.oracle.com/technology/pub/articles/oracle-database-11g-top-features/index.html>
- http://www.dba-oracle.com/oracle11g/oracle_11g_new_features.htm

Kontakt:

Timo Meyer
meyer@stz-isd.de

Partitioning ist eine Frage der richtigen Strategie

Autor: André Schulte, Freiberufler

Das Partitioning ist für die Verwaltung großer Datenmengen seit der Version 8 eine meist unverzichtbare Option in der Oracle Datenbank. Die Kunst der Partitionierung besteht nicht im Erstellen der Partitionen, sondern in der Auswahl des richtigen Partitions-Designs. Der Artikel erläutert Wege zur besten Partitionierungs-Strategie sowie eine sinnvolle Anwendung der neuen Partitionierungsfeatures in Oracle 11g.

Partitioning ist eine Option, die zur Enterprise Edition der Datenbank erworben werden kann. Sie bietet die Möglichkeit, eine Tabelle in kleinere Einheiten mit eigenen Tablespaces aufzuteilen. Diese Änderung ist für die Applikation transparent und kann auch im Online-Betrieb mittels des DBMS_REDEFINITION-Packages durchgeführt werden. Eine gute Applikation bringt das Partitioning von Haus aus mit. Es bestimmt entscheidende Kriterien, wie Manageability, Performance, Verfügbarkeit und Information Lifecycle Management, die im Idealfall im Voraus geplant werden. Der Einsatz von Partitioning ist in unterschiedlichen Arten von Applikationen möglich. Üblicherweise profitieren OLTP-Systeme von einer Verbesserung der Manageability und Verfügbarkeit, wohingegen Data-Warehouse-Anwendungen von besserer Manageability und Performance profitieren. Zur Auswahl der richtigen Partitionierungs-Strategie sollte eine individuelle Priorisierung der zu erreichenden Ziele unter Beachtung folgender Kategorien vorgenommen werden:

- *Manageability*
Verschieben, Mergen, Löschen und Austauschen von Partitionen ermöglichen eine schnellere Ausführung von Aufgaben als Operationen, die ganze Tabellen betreffen. Hilfreich beim Laden, Löschen und Verschieben von Daten. Erstellung und Rebuild von Indizes kann auf Partitionsebene schneller als auf Tabellenebene durchgeführt werden.
- *Performance*
Der Optimizer überspringt Partitionen, die nicht durchsucht werden müssen, sogenannte Partition Pruning. bei Abfragen kann für Partitionen ein Load Balancing über physische Devices angewendet werden. Der Optimizer beschleunigt Abfragen mit Partition-Wise Joins, indem die Ausführung partitionsweise optimiert wird.
- *Verfügbarkeit*
Aufteilen einer Tabelle in kleinere Tablespaces, somit Vorteile beim Backup und Recovery. Schnelles Löschen von Daten mit der Drop-Partition-Funktion. Verschieben von Partitionen auf andere Speichermedien bei gleichzeitigem Zugriff mit Select Statements.

- *Integration von Partitioning in das Information Lifecycle Management (ILM) Konzept*

Zusammen mit dem Oracle ILM Assistant hilft Partitioning mit seinen Datenverschiebe- und Löschoptionen Daten kosteneffektiv vorzuhalten. Dabei befinden sich häufig genutzte Daten auf schnellen, teureren Platten und selten benutzte Daten auf langsamen, billigeren Medien.

Partitionierungsmethoden

Die Partitionierungsmethode gibt an, nach welchen Regeln eine Tabelle aufgeteilt wird. Das Range-Partitioning ist die häufigste Methode, die auf einer Zeit-Spalte der Tabelle basieren kann. Ein Beispiel soll dies verdeutlichen:

```
CREATE SEQUENCE SALES_SEQ;
/
CREATE TABLE SALES_PART
PARTITION BY RANGE (TIME_ID)
INTERVAL (INTERVAL '1' YEAR) STORE IN
(TAB_1, TAB_2, TAB_3)
(PARTITION P1999 VALUES LESS THAN (TO_
DATE ('01.01.1999', 'dd.mm.yyyy')) TA-
BLESPEC TAB_1)
AS
(SELECT sales_seq.nextval SALES_ID,
SALES.* FROM SH.SALES SALES WHERE
TIME_ID > TO_DATE('01.01.1999', 'dd.
mm.yyyy'))
/
```

Hier wurde die Sales-Tabelle nach dem `time_id`-Range partitioniert. Die Sequenz wird für das Füllen des Feldes `sales_id` benötigt, auf das später noch eingegangen wird. Beim Statement wird nur die erste Partition angegeben. Die anderen Partitionen werden von Oracle automatisch mittels der in 11g eingeführten Intervall-Partitionierung angelegt. Die Tablespaces der Partitionen werden nach den in der „STORE IN“-Klausel angegebenen Tablespaces im Round-Robin-Prinzip gewählt.

Nachdem die Statistiken berechnet wurden zeigt die View USER_TAB_PARTITIONS die Partitionen mit der Anzahl der Datensätze an.

```
SQL> exec dbms_stats.gather_table_stats(user, ,SALES_PART')

PL/SQL procedure successfully completed.

SELECT PARTITION_NAME, NUM_ROWS, HIGH_VALUE FROM USER_TAB_PARTITIONS WHERE TABLE_NAME = ,SALES_PART'
/
PART_NAME NUM_ROWS TABLESP HIGH_VALUE
-----
P1999      0          TAB_1    TO_DATE(,1999-01-01 00:00:00', ,SYYYY-MM-DD HH24:MI:SS', ,NLS_CALENDAR=GREGORIAN')
SYS_P48    247707         TAB_2    TO_DATE(,2000-01-01 00:00:00', ,SYYYY-MM-DD HH24:MI:SS', ,NLS_CALENDAR=GREGORIAN')
SYS_P49    232646         TAB_3    TO_DATE(,2001-01-01 00:00:00', ,SYYYY-MM-DD HH24:MI:SS', ,NLS_CALENDAR=GREGORIAN')
SYS_P50    259418         TAB_1    TO_DATE(,2002-01-01 00:00:00', ,SYYYY-MM-DD HH24:MI:SS', ,NLS_CALENDAR=GREGORIAN')
```

Eine andere Partitionierungsmethode ist das List-Partitioning (siehe Beispiel Tabelle sales_list), die als Kriterium eine Werteliste für jede Partition hat. Die Listen müssen untereinander disjunkt sein. Beim Einfügen eines Wertes wählt die Datenbank automatisch die richtige Partition aus. Es gibt die Möglichkeit, eine Default Partition zu definieren. Ohne Default Partition werden Werte, die keiner Partition zugewiesen werden können mit einem Fehler von der Datenbank abgewiesen. Beim Hash-Partitioning werden Daten mithilfe eines Hashing-Algorithmus zu einer Partition zugeordnet. Der Vorteil liegt darin, dass eine Partitionierung angewendet werden kann, obwohl kein List- oder Range-Partitionierungskriterium vorhanden ist.

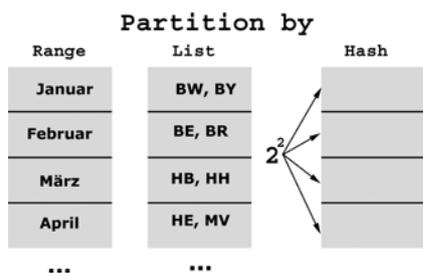


Abbildung 1: Range-, List- und Hash-Partitioning nach Monaten und Bundesländern

Das Composite Partitioning ist eine Mischung aus zwei der zuvor genannten Partitionierungsmethoden. Bis zur Version 10gR2 konnte man eine Range-Hash- oder eine Range-List-Kombination nutzen. Ein Beispiel für eine Range-List-Partitionierung wären Daten für Bundesländer, die nach Monat und Bundesland partitioniert sind. Mit der Einführung von 11g kann jetzt für eine Range- oder List-Partition eine beliebige Sub-Partition mit Range, List und Hash gewählt werden.

Laden von Daten

Ein Vorteil der Partitionierung ist das einfache Hinzufügen und Löschen von Daten. Will man in der Tabelle SALES_PART die Daten von 1999 bereitstellen, kann man diese erst in eine temporäre Tabelle vorladen und dann mit einem Partition Exchange übernehmen.

```
SQL> CREATE TABLE TMP_LOAD AS SELECT * FROM SALES_PART WHERE 1=0
/
Table created.

SQL> INSERT INTO TMP_LOAD SELECT sales_seq.nextval, SALES.* FROM SH.SALES WHERE TIME_ID < TO_DATE(,01.01.1999', ,dd.mm.yyyy')
/
178834 rows created.

SQL> ALTER TABLE SALES_PART EXCHANGE PARTITION P1999 WITH TABLE TMP_LOAD
/
Table altered.

SQL> exec dbms_stats.gather_table_stats(user, ,SALES_PART', ,P1999')

PL/SQL procedure successfully completed.

SELECT PARTITION_NAME, NUM_ROWS FROM USER_TAB_PARTITIONS WHERE TABLE_NAME = ,SALES_PART'
/
PART_NAME NUM_ROWS
-----
P1999      178834
SYS_P48    247707
SYS_P49    232646
SYS_P50    259418
```

Die temporäre Tabelle muss die gleiche Struktur wie die partitionierte Tabelle haben. Dies wird mit dem ersten

Statement erreicht. where 1=0 bewirkt, dass die erstellte Tabelle leer ist. Die Exchange-Partition-Anweisung dauert so lange wie das Vorladen von 10 Millionen Datensätzen, da nur ein Update im Dictionary erfolgt. Die Statistiken werden nur für die geänderte Partition neu berechnet.

Indizes und Partitioning

Ein Index auf einer partitionierten Tabelle kann partitioniert oder unpartitioniert sein. Folgende Möglichkeiten der Partitionierung ergeben sich.

- *local prefixed*
Der Index ist wie die Tabelle partitioniert (local) und die führenden Spalten des Indexschlüssels entsprechen dem Partitionierungsschlüssel des Index
- *local nonprefixed*
Der Index ist wie die Tabelle partitioniert, aber die führenden Spalten des Indexschlüssels entsprechen nicht dem Partitionierungsschlüssel des Index
- *global prefixed*
Der Index ist anders als die Tabelle partitioniert und die führenden Spalten des Indexschlüssels müssen dem Partitionierungsschlüssel entsprechen, da global nonprefixed Indizes nicht unterstützt werden.

Das folgende Beispiel zeigt einen local-prefixed-Index auf der Spalte `time_id`:

```
SQL> CREATE INDEX TIME_IDX ON SALES_PART
(time_id) LOCAL COMPUTE STATISTICS
/
Index created.

SQL> SELECT ui.INDEX_NAME, up.PARTITION_
NAME, up.NUM_ROWS, ui.PARTITIONING_TYPE
FROM USER_IND_PARTITIONS up, USER_PART_
INDEXES ui where up.index_name='TIME_IDX'
and ui.index_name='TIME_IDX'
/
```

INDEX_NAME	PART_NAME	NUM_ROWS	PARTITI
TIME_IDX	SYS_P101	247707	RANGE
TIME_IDX	SYS_P102	232646	RANGE
TIME_IDX	SYS_P103	259418	RANGE
TIME_IDX	P1999	178834	RANGE

Der Index ist genau wie die Tabelle Range partitioniert. Bei der Verwendung von Indizes muss man beachten, dass diese durch DDL Statements gegen die Tabelle unusable werden können, wie im Beispiel mit einem globalen unpartitionierten unique-Index auf `sales_id` veranschaulicht.

```
SQL> create unique index sales_idx on
sales_part (sales_id) global
/
Index created.

SQL> alter table sales_part drop parti-
tion SYS_P50;

Table altered.

SQL> select index_name, status from
user_indexes where index_name = ,SALES_
IDX'
/
```

INDEX_NAME	STATUS
SALES_IDX	UNUSABLE

Der Index wird erst nach einem Rebuild wieder gültig. Es gibt ab 9i die Möglichkeit, globale Indizes sowie ab 10g lokale Indizes durch die Datenbank pflegen zu lassen. Dazu muss beim Erstellen der Tabelle die UPDATE-INDEXES-Klausel angegeben werden.

Performance mit Partition-Pruning steigern

Um mit dem Partitioning die Performance zu steigern, helfen Partition-Pruning und Partition-Wise-Joins. Beim Partition-Pruning wird nur die Partition ausgesucht, die für die Abfrage relevant ist. Dazu muss das Partitionskriterium in der Where-Klausel der Abfrage vorhanden sein. Bei einem Zugriff über einen Index bewirkt das Partition-Pruning nichts. Bei Full Table Scans wird aber je nach Daten eine immense Beschleunigung erreicht.

```
SQL> explain plan for
select * from sales_part where time_
id between to_date('01.01.1998', ,dd.
mm.yyyy') and to_date('31.12.1998', ,dd.
mm.yyyy');

Explained.

SQL> select * from table(dbms_xplan.dis-
play);

PLAN_TABLE_OUTPUT
-----
| Id | Operation | Name
| Pstart| Pstop
-----
```

```
| 0 | SELECT STATEMENT | |
|   |                   |
| 1 | PARTITION RANGE SINGLE |
|   | 1 | 1 |
|* 2 | TABLE ACCESS FULL | SALES_
PART | 1 | 1 |
```

(Ausgabe vereinfacht)

Das Beispiel zeigt ein static Partition-Pruning, bei dem zur Erstellung des Ausführungsplans bereits festgelegt ist, welche Partitionen benötigt werden. Beim dynamic Partition-Pruning wird dies erst zur Laufzeit entschieden, beispielsweise bei der Verwendung von bind Variablen oder Unterabfragen. Im Ausführungsplan wird nur auf eine der drei Partitionen zugegriffen und rund zwei Drittel der Daten ausgeblendet. Gerade wenn das Partitioning nachträglich in die Anwendung eingebaut wird ist es häufig notwendig Statements für das Partition-Pruning in der Where-Klausel zu optimieren.

Virtuelle Spalten partitionieren

In 11g wurden aber noch weitergehende Konzepte umgesetzt. Findet man keinen geeigneten Partitionierungsschlüssel, kann man versuchen, diesen durch eine virtuelle Spalte zu erzeugen. Werte für virtuelle Spalten werden nicht in der Datenbank gespeichert, sondern nur die deterministische Funktion zur ihrer Berechnung. Das Partitioning auf diesen virtuellen Spalten kann mit allen drei Basis-Partitionierungsmethoden oder Composite-Partitioning erfolgen. Geht man davon aus, dass in einer ProduktID in den ersten beiden Stellen die Produkt-Kategorie codiert ist, kann man diese als virtuelle Spalte anlegen und, wie das Beispiel zeigt, nach der Kategorie die Tabelle List partitionieren.

```
create table sales_list(
prod_id number, cust_id number,
time_id date, prod_category number(2) as
(substr(prod_id, 1, 2)) virtual)
partition by List (prod_category)
(partition database values (10),
partition applications values (20),
```

```
partition middleware values (30),
partition others values (default))
/
```

Reference-Partitioning

Ein anderes neues Partitionierungsfeature in 11g ist das Reference-Partitioning. Will man zu jedem Verkauf in der Sales-Tabelle eine Lieferadresse in einer separaten Tabelle speichern, kann man diese nicht wie die Sales-Tabelle partitionieren, weil die time_id-Spalte fehlt. Mit dem Reference-Partitioning kann man die Verbindung zwischen den beiden Tabellen mit einem foreign Key-Constraint herstellen und so eine equipartitionierung zur sales_part-Tabelle erreichen.

```
create table salesdeliver(
sales_id number not null,
address varchar2(400),
constraint fkc_salesdeliver_sales
foreign key (sales_id) references
sales_part)
partition by reference (fkc_salesdeliv-
er_sales)
/
```

Hinweis: Damit das Beispiel ausgeführt werden kann, muss die sales_part-Tabelle ohne Intervall-Partitionierung angelegt werden und sales_id der primary Key sein.

Fazit

Das Partitioning ist eine vielfältige, leistungsstarke Option, die Verbesserungen von der Maintenance über die Verfügbarkeit bis zur Performance ermöglicht – und das oft ohne Applikationsänderung. Die Erweiterungen in 11g ergänzen Lücken und runden die Option ab. Mit Referenz-Partitioning, erweitertem Composite-Partitioning, Partitioning mit virtuellen Spalten und Intervall-Partitioning lassen sich jetzt Anwenderwünsche noch genauer realisieren.

Kontakt:

André Schulte
contact@andschulte.de

SIG Data Warehouse

Die DOAG veranstaltet am Dienstag, 17. Juni 2008, das Treffen der Special Interest Group Data Warehouse zum Thema Data Warehouse Operations & BI Infrastructure im Holiday Inn Munich in Schwabing.

Weitere Infos und Anmeldung unter <http://www.doag.org/sig/dwh/>

Einfaches Reporting mit OpenOffice Calc

Autorin: Dr. Heike Püttmann

Sind die Ansprüche und die Datenmengen nicht zu hoch, reicht manchmal schon ein Tabellenkalkulationsprogramm wie Excel oder Calc aus, um ein Reporting, ähnlich dem Discoverer, zu erstellen. Der Artikel stellt den Einsatz von OpenOffice 2.3.1 und Oracle 10g XE vor.

Der wichtigste Aspekt ist zunächst die Tatsache, dass direkt und innerhalb von OpenOffice über das Modul Base (ein – sehr abgespecktes – Microsoft-Access-Pendant) die Möglichkeit besteht, verschiedenste Datenbank-Verbindungen aufzubauen. Damit ist man also nicht auf die maximal 65.536 Zeilen begrenzten Tabellenblätter als Datenquellen beschränkt.

Etliche Treiber werden mit OpenOffice als Standard geliefert: Adabas, DBase, Text, JDBC etc. Ist eine Datenquelle erst einmal in OpenOffice angemeldet, kann man sowohl im Writer als auch im Calc jederzeit über den Menüpunkt Ansicht => Datenquellen auf diese zugreifen.

Ähnlich wie Microsoft Excel verfügt auch OpenOffice Calc über die Möglichkeit, Pivot-Tabellen (Kreuztabellen) aus vorhandenen Tabellen zu erstellen. Das bietet die Möglichkeit, Spalteninhalte schnell und unkompliziert auch als Spaltenbeschriftungen (= Gruppierung von Spalteninhalten) zu interpretieren. Man lehne sich zurück und führe sich den Aufwand vor Augen, diese Pivotierung in der Oracle-Datenbank selbst berechnen zu müssen ...

Daneben kann man auch Spalten-Inhalte als Seitenelemente verwenden, sodass man in den Datenblättern kann. Ändern sich die Daten in der Quelle, also in unserem Falle in der XE-Datenbank, können die geänderten Werte mit einer einfachen Aktualisierung in die Pivot-Tabelle geladen werden.

Entsprechend aufgebaute Pivot-Tabellen sind schon wesentlich aussagekräftiger als die zugrunde liegenden Original-Tabellen, manchmal aber ist eine Grafik noch aussagekräftiger. Die Grafiken (deren Definitionen) enthalten Zell-Referenzen, um den darzustellenden Inhalt abzurufen. Hier liegt nun der ganz besondere Kniff: ändern sich die Inhalte in den Tabellen beziehungsweise Pivot-Tabellen – zum Beispiel durch Blättern im Seitenelement – aktualisiert sich die Grafik ebenfalls.

In einer OpenOffice-Calc-Mappe können auch mehrere Pivot-Tabellen, auch zum selben Datenbank-Objekt, in einem Blatt vorhanden sein. So kann man sich eine bedarfsgenaue Reportingmappe zusammenstellen.

Datenbank-Verbindung in OpenOffice

Als erstes ist der passende Treiber in OpenOffice anzumelden. Als Treiber soll der mit Oracle 10g XE mitgelieferte JDBC-Treiber (C:\oracle\app\oracle\product\10.2.0\server\jdbc\libodbc14.jar-Archiv) genutzt werden. Damit OpenOffice auf diesen zugreifen kann, muss dieser mit Pfad und Namen zunächst über Extras => Optionen => Abteilung OpenOffice.org => Unterabteilung Java angemeldet werden, wobei man zuerst den Button <Class Path> und dann den Button <Archive hinzufügen> wählt. Dann wird der Pfad angegeben und mit dem Button <OK> bestätigt. Im nächsten Schritt richtet man die Datenbank-Verbindung ein, wobei der Anmelde-Dialog in OpenOffice mit Datei => Neu => Datenbank geöffnet wird. Danach folgen die Schritte

1. *Datenbank auswählen*
Verbindung zu einer bestehenden Datenbank mit Treiber Oracle JDBC herstellen
2. *Datenbank-Verbindung einrichten*
Angabe: Name der Datenbank, Rechnername und Portnummer. Mit Button <Klasse testen> Klassen-Verwendung prüfen
3. *Benutzerauthentifizierung einrichten*
Angabe: Benutzername und Aktivierung „Kennwort erforderlich“. Mit Button <Verbindungstest> die Verbindung prüfen
4. *Fertigstellen und fortfahren*
Aktivierung: „Ja, die Datenbank soll angemeldet werden“, nicht erforderlich: „Die Datenbank zum Bearbeiten öffnen“. Im folgenden Speichern unter Fenster den Name und Ort der Datenbank-Verbindung festlegen und mit <Speichern> den Vorgang beenden

Das Anmelden der Datenbank in OpenOffice bedeutet nur eine Anmeldung auf dem lokalen Rechner – nicht weltweit. Die Konfiguration wird in der im letzten Schritt angegebenen Datei abgespeichert. Bei dieser – wie bei allen anderen OpenOffice-Dateien auch – handelt es sich um gezippte Ordner und XML-Dateien im OASIS-Format, deren Inhalt Sie sich nach

Entwicklung Reports

dem Entpacken ansehen oder auch ändern können. Das Passwort ist nicht darin enthalten.

Erstellen einer Auswertung

Im folgenden Beispiel bildet die in Abbildung 1 dargestellte Datenbank-Tabelle die Grundlage für die Auswertung.

JAHR	MONAT	TAG	LT_STATION_1	LF_STATION_1	LT_STATION_2	LF_STATION_2
2002	200203	02	13,1	63	13,4	69
2002	200203	03	8,5	62	8,9	66
2002	200203	04	7,6	67	8,3	68
2002	200203	05	8,4	64	8,7	85
2002	200203	06	9,5	79	9,7	82
2002	200203	07	6,2	69	7	73
2002	200203	08	7	72	8	71

Abbildung 1: Datenbanktabelle für Reporting-Beispiel

Nach dem Öffnen von OpenOffice Calc ruft man den Datenpiloten auf (Daten => Datenpilot => Aufrufen). Im neu geöffneten Kontextfenster wird als Datenquelle „In OpenOffice.org angemeldete Datenquelle“ ausgewählt. Dann wird man in einem zweiten Fenster aufgefordert, Benutzername und Passwort einzugeben. Dann wählt man im ersten Fenster die gewünschte Tabelle (Datenquelle) aus.

Das abschließende dritte Fenster zeigt einen Layout-Entwurf sowie die Felder der zuvor ausgewählten Tabelle. Mit drag-and-drop zieht man die Felder entsprechend seinen Vorstellungen in den Layout-Bereich. Das Tabellenblatt wird nach dem letzten Betätigen von <OK> gefüllt.

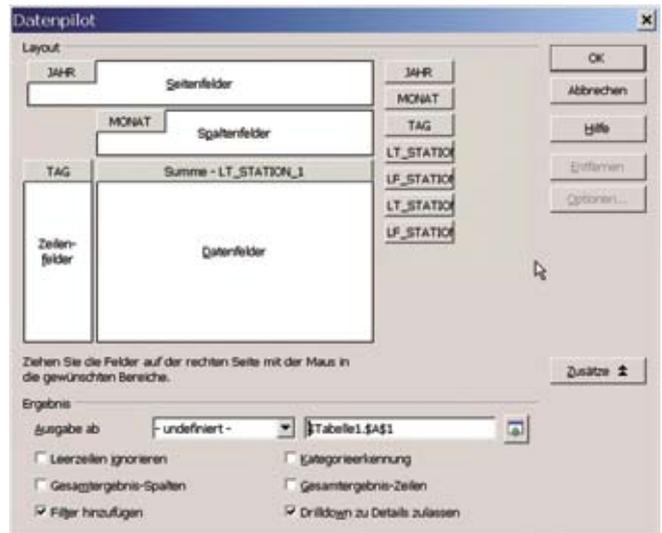


Abbildung 2: Ansicht Datenpilot in OpenOffice Calc

Klickt man mit der rechten Maustaste in den Pivot-Tabellenbereich, kann man über das Kontextmenü wieder den Datenpiloten aufrufen oder – wenn sich beispielsweise die Daten in der Datenbank geändert haben – die Daten aktualisieren.

Auf den Daten-Bereich der Pivot-Tabelle lässt sich auch ein Diagramm aufsetzen. Auf einer etwas anders formatierten Pivot-Tabelle sieht dies wie in Abbildung 3 dargestellt aus. Beim Blättern im Seitenelement [JAHR] ändert sich ebenfalls die Grafik. Eine solche Grafik-Definition ist allerdings nur dann sinnvoll, wenn der Datenbereich auch nach dem Blättern immer gleich groß bleibt. Sonst schneidet die Grafik entweder die Datenreihe ab oder es gibt eine Lücke am Ende der Reihe.

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	JAHR	2002											
2													
3	Summe - LT_STATION_1	MONAT											
4	TAG	200201	200202	200203	200204	200205	200206	200207	200208	200209	200210	200211	200212
5	D1	0	-0,4	10,9	10,3	11,8	12,4	16,5	15,5	21,6	14,7	3,2	3,1
6	D2	0	-2	13,1	9,5	14,5	18,3	20,5	18,7	21,7	13	4,4	-1,3
7	D3	0	0,1	8,5	9,3	15,5	16,1	19,8	21,1	18,3	11,4	5,3	0,1
8	D4	0	2,9	7,6	5,3	15,8	18	16,5	21,9	20	0	1,6	1,4
9	D5	0	2,8	8,4	4,7	16,4	20	18,1	21,6	20,8	0	6,9	0,3
10	D6	0	2,7	9,5	4,9	14,1	20	16,5	20,6	18,1	0	8,3	2,5
11	D7	0	5,5	6,2	3,5	8	21,3	18,9	20,7	16,1	0	11,1	3,3
12	D8	0	6,5	7	6,3	10,5	20,2	19,8	21,4	18,2	18,9	8,6	3,8
13	D9	0	6,5	8,7	9,8	12	20	19,5	22,3	16,3	19,4	9,2	4,7
14	D10	0	9,1	7,1	8,9	11,6	20,3	19,1	22,8	13,7	15,2	9,9	8,2
15	D11	0	7	7,6	7,8	17,4	23,2	20,7	23,3	15	12,3	9,1	10,9
16	D12	0	8,1	5,5	6,2	16,4	21,3	16,2	23,2	19,5	10,3	5,9	8,8
17	D13	0	8,3	9,3	6,2	18,1	20,6	0	23	14,3	9,2	6,6	5,3
18	D14	0	4,9	9,4	8,3	20,9	18,7	0	23,2	12,6	8,1	1,8	3,9
19	D15	0	2,3	10,1	6	18,3	17,2	0	20,6	12,8	7,2	6	-0,4
20	D16	0	1,7	4,6	5,1	21,3	15	0	21,3	14,9	4,6	6,4	-7,5
21	D17	0	2,5	2,1	6,2	23,3	14,9	0	21,5	15,3	7,6	5,9	-7,7
22	D18	0	5,6	3,6	8,3	22,2	13,2	18,7	19,9	16,3	12,2	3,7	-1,8
23	D19	0	4	5,9	6,2	20,1	15,6	17,2	21,2	12,9	12,2	-0,4	1
24	D20	0	0	1	5	18,6	14,5	15,1	21,9	10	8,6	-0,7	0,5
25	D21	0	0	2	5,4	15,9	15,3	18,5	22,5	9,6	3	1,9	1,7
26	D22	3,4	0	2,9	5,7	12	15,5	21,4	23,2	10,6	2	1,6	1,8
27	D23	5,7	0	1,9	6,8	10,7	15,5	19,7	23,3	12,7	5,6	1,9	2,3
28	D24	4	0	4,6	10,9	10,1	14,4	20	24,9	13,8	6,2	2,7	5,1
29	D25	2,3	11,5	6	11,1	11,4	15,5	18,8	25,2	12,4	5,9	2,7	10,8
30	D26	0,8	9,1	8,3	10,1	12,9	16,4	17,1	22,5	13,8	3,8	2,9	9,8
31	D27	-0,1	5,2	12,2	11,2	11,6	17,7	19,7	21,8	11,4	0,2	2	6,8
32	D28	1,5	8,4	8,9	13,2	11,8	19,6	18,7	22,2	9,8	0,4	3,7	5,8
33	D29	3		6	11,8	12,8	25,1	17,8	16,6	13,5	1	7,9	5
34	D30	3,4		6,7	9,9	12,8	18,4	20,3	17,9	14,8	3,7	8,5	4
35	D31	3		9,9		10,8		18	19,3		3,4		5,6

Abbildung 3: Fertige Pivot-Tabelle in OpenOffice Calc

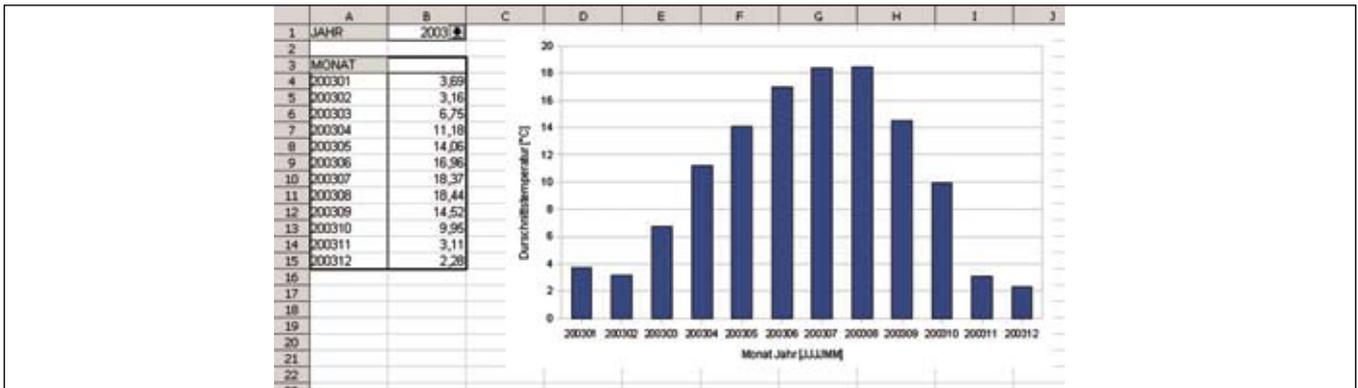


Abbildung 4: Diagramm auf Pivot-Tabellendaten

Einschränkungen

Die Reporting-Möglichkeiten mit OpenOffice Calc haben leider ein paar Mängel, die hier nicht unerwähnt bleiben sollen. Ein registrierter Fehler ist die Unfähigkeit vom Datenpiloten Views anzuzeigen – obwohl man diese in OpenOffice Base nutzen kann. Hier hilft nur das Warten auf ein neues Release.

Der angemeldete Datenbank-Benutzer sieht über die Oracle-JDBC-Verbindung alle Datenbank-Tabellenobjekte, für die er Zugriffsrechte hat. Wer sich an das Eingemachte traut, bearbeitet die Datenbank-Datei, die ja mehrere gezippte XML-Dateien enthält: mit einem Packprogramm die *.odb öffnen, in der content.xml einen Filter ähnlich Listing 1 setzen, alle Dateien wieder zippen und in *.odb umbenennen. Danach werden nur noch die Tabellenobjekte aus dem vorgegebenen Schema aufgelistet.

```
<db:table-filter-pattern>SCHEMANAME.%</db:table-filter-pattern>
```

Listing 1: Filtereintrag in content.xml der Base-Datenbankdatei

Wenn man eine Datenbankverbindung auf ein großes Data Warehouse und dann auch noch über ein

Netzwerk aufbaut, kann die Performance stark absinken. Hier ist die genannte Überarbeitung notwendig.

Fazit

Im Vergleich zu Microsoft Excel 2003 sind OpenOffice Calc-Diagramme immer Teil eines Tabellenblatts. Dies lässt verschmerzen, dass nicht wie in Microsoft Excel-Diagrammblättern Drop-Down-Listen für die Seiten-, Spalten- und Zeilenelemente der Pivot-Tabelle in der Grafik selbst enthalten sind. Schließlich liegen diese Elemente ja gleich nebenan.

Weitere Informationen

- Erläuterungen zu Pivot-Tabellen mit OpenOffice Calc: <http://www.oowiki.de/DatentabellenAuswerten/PivotTabelle>
- Oracle JDBC-Treiber: Oracle Database JDBC Developer's Guide and Reference 10g Release 2 (10.2, B14355-02)

Kontakt:

Dr. Heike Püttmann
heikepuettmann@web.de

Unsere Inserenten			
AMIS Deutschland GmbH www.amis-de.com	Seite 45	ORACLE Deutschland GmbH www.oracle.com	Umschlagseite 3
Hunkler GmbH & Co KG www.hunkler.de	Seiten 3, 69	ORDIX AG www.ordix.de	Seite 79
IDG-Verlag Redaktion Computerwoche www.computerwoche.de	Seite 71	PROMATIS software GmbH www.promatis.de	Seite 29
Keptool GmbH www.keptool.com	Seite 21	Quest Software GmbH www.quest.com	Seite 5
Merck Shared Services Europe GmbH www.msse.de	Seite 15	Trivadis AG www.trivadis.com	Umschlagseite 4, Seiten 9, 53
OPITZ CONSULTING GmbH www.opitz-consulting.de	Umschlagseite 2	Team GmbH Paderborn www.team-pb.de	Seite 55

Service-orientierte Architekturen mit Oracle – eine Vorschau auf die Oracle SOA Suite 11g

Autoren: Christian Schmülling und Ingo Meier, OPITZ Consulting GmbH

Der Artikel stellt die Bedeutung der SOA-Infrastrukturlösung Oracle SOA Suite für die Umsetzung eines fachlich motivierten, service-orientierten Ansatzes in Unternehmen vor. Darüber hinaus werden neue Komponenten und Änderungen in der Oracle SOA Suite 11g sowie ihr Einsatz an einem Praxisbeispiel erläutert.

Das in den vergangenen Jahren vieldiskutierte Thema service-orientierter Architekturen (SOA) hat zahlreiche Facetten – vom fachlichen und organisatorischen Aufbau eines Service-Portfolios über die Entwicklung von IT-Service-Komponenten bis hin zum Betrieb service-orientierter Unternehmensanwendungen auf entsprechenden Infrastruktur-Lösungen.

Fachlich betrachtet ist der Kern einer SOA die Kapselung wieder verwendbarer Funktionalität in gut dokumentierten Services. Diese Funktionalität wird von Service-Anbietern zur Verfügung gestellt und kann entsprechend von interessierten Service-Konsumenten genutzt werden. Über die Nutzung einzelner Services hinaus ist auch die logische Verknüpfung mehrerer Services zu wertschöpfenden Geschäftsprozessen im Unternehmen ein wichtiger Aspekt im Zusammenhang mit einer SOA: Services werden in diesem Szenario als Bausteine betrachtet, die einzelne Schritte der Geschäftsprozesse unterstützen.

Will man diesen fachlichen Ansatz auf die IT eines Unternehmens übertragen, erfordert dies einerseits Technologien für die Entwicklung service-orientierter Software und andererseits eine Infrastruktur, die Deployment, Betrieb und Verwaltung der Komponenten

ermöglicht. Für die Entwicklung service-orientierter Anwendungen haben sich die Web-Services-Technologien weitgehend durchgesetzt. In diesem Umfeld stehen zahlreiche Standards zur Verfügung, die vom Nachrichtenversand zwischen Services (SOAP, XML) über die Beschreibung und das Auffinden von Services (WSDL, UDDI) bis hin zur Nutzung von Services in Geschäftsprozessen (BPEL) alle erforderlichen technologischen Schichten umsetzen (vgl. Abbildung 1).

Oracle SOA Suite – eine SOA-Infrastrukturlösung

Diese weitgehend standardisierten Technologien nutzt auch die Oracle SOA Suite (aktuell in der Version 10g verfügbar). Sie bietet ein umfangreiches Portfolio an Produkten, die die verschiedenen Implementierungsaspekte einer SOA unterstützen:

- Erstellung, (Wieder-)Verwendung und Verwaltung von Services (JDeveloper, Enterprise Manager, UDDI)
- Sicherheit von Services (Web-Services-Manager)
- Routing und Transformation von Nachrichten zwischen Services und Anwendungen (Enterprise Service Bus / Mediator)
- Orchestrierung von Services zu Prozessen (BPEL Process Manager)
- Integration von Anwendungen (Adaptoren in ESB und BPEL)
- Einbeziehung von Personen in serviceorientierte Anwendungen (Human Workflow)
- Auslagerung von Geschäftsregeln (Business Rules Engine)
- Ermittlung und Visualisierung von Kennzahlen (KPI) direkt aus der Laufzeitumgebung (Business Activity Monitoring)

Mit der demnächst erscheinenden Version 11g der Oracle SOA Suite wird es eine Reihe interessanter Änderungen und Erweiterungen dieses Produktportfolios geben, auf die dieser Artikel anhand eines praktischen Beispiels eingeht.

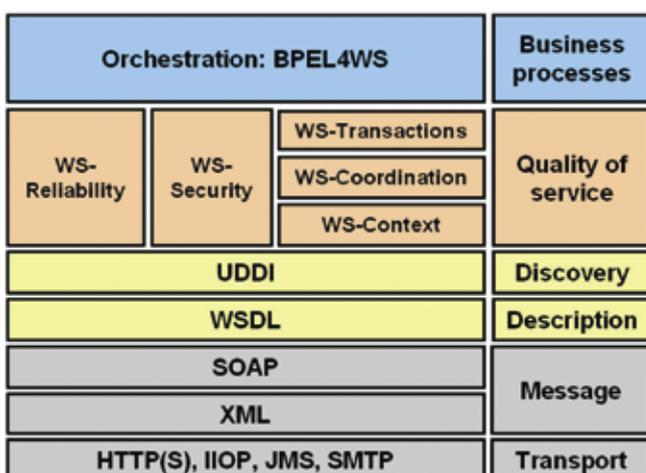


Abbildung 1: Web-Services-Technologien und ihre Einsatzgebiete

Architektur der Oracle SOA Suite 11g

Das Herzstück in der Architektur der SOA Suite 11g ist der Enterprise Service Bus (siehe Abbildung 2). Seine wesentliche Aufgabe besteht in der Transformation und Weiterleitung von Nachrichten, die durch die verschiedenen Service-Komponenten fließen. Darüber hinaus bietet er auch die Möglichkeit, über eine Vielzahl an Adaptoren andere Systeme wie Datenbanken, Messaging-Systeme, ERP-Anwendungen und viele weitere zu integrieren. Die übrigen Service-Komponenten wie BPEL, Human Workflow, Business Activity Monitoring werden um den ESB gruppiert und dienen entweder als Erzeuger oder Konsumenten von Nachrichten.

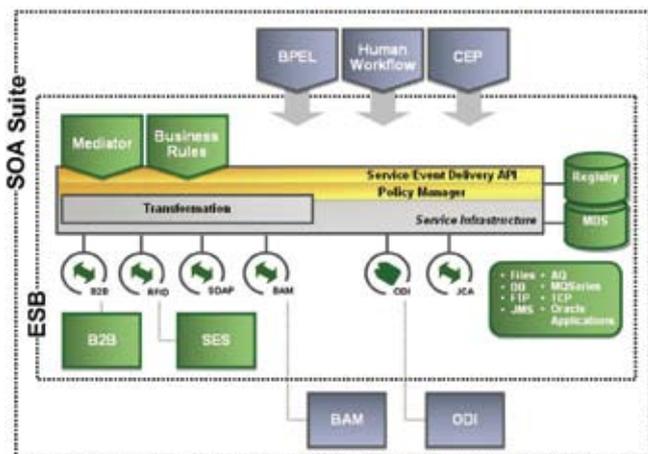


Abbildung 2: Architektur der Oracle SOA Suite 11g

Die Service Component Architecture (SCA)

Bei der Entwicklung service-orientierter Anwendungen kommt in der Version 11g das Konzept der Service Component Architecture (SCA) zum Einsatz, das im Kern ein von verschiedenen Organisationen und Herstellern vorgeschlagenes Modell zur Entwicklung service-orientierter Anwendungen beschreibt. SCA liefert dem Entwickler einerseits einen groben Überblick über die erstellten Service-Komponenten und deren Beziehungen untereinander, andererseits ermöglicht die Betrachtung mehrerer Service-Bausteine als Business Application (SCA-Kompositum) ein vereinfachtes Deployment der Artefakte in die Laufzeit-Umgebung. Abbildung 3 zeigt die Entwicklung einer SOA-Anwendung mit einem sol-

chen SCA-Kompositum im Oracle JDeveloper, der zentralen Entwicklungsumgebung für die SOA Suite 11g. Dabei werden alle Service-Komponenten als Bausteine in der Mitte des SCA-Bilds angelegt und miteinander verbunden. Musste für die Entwicklung einiger Service-Komponenten (zum Beispiel Business Rules) in der Version 10g noch eine browserbasierte Oberfläche genutzt werden, kann mit dem JDeveloper 11g die Entwicklung sämtlicher Service-Komponenten in einer einheitlichen Entwicklungsumgebung durchgeführt werden. In der linken Spalte des SCA-Kompositums werden die nach außen zur Verfügung gestellten Schnittstellen, die zum Aufruf der Service-Anwendung dienen, modelliert. In der rechten Spalte werden externe Referenzen – also beispielsweise andere Services – abgebildet, auf die die Service-Komponente zur Erbringung ihrer Dienstleistung zugreift.

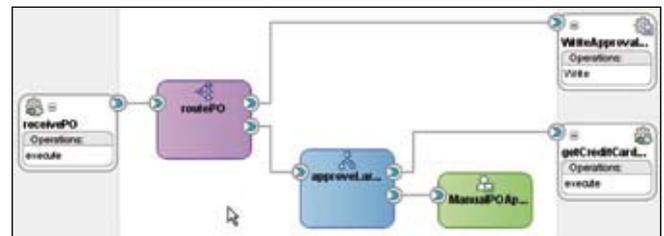


Abbildung 3: Entwicklung eines SCA-Kompositums mit Oracle JDeveloper 11g

Das in Abbildung 3 gezeigte SCA-Kompositum nimmt eingehende Nachrichten in einem Mediator Service (routePO) entgegen. Dieser Service leitet die Nachricht unter anderem an einen BPEL-Prozess (approveLargeOrder) weiter. Die Konfiguration eines Mediator Service (siehe Abbildung 4) ermöglicht das Weiterleiten (Routing) einer eingehenden Nachricht an einen oder mehrere Endpunkte. Die Inhalte können zuvor gefiltert werden und ermöglichen so ein inhaltsabhängiges Routing. Außerdem können Nachrichten-Formate im Mediator von einer eingehenden XML-Struktur auf eine abweichende ausgehende XML-Struktur transformiert und Protokoll-Umwandlungen – etwa von SOAP/HTTP auf JMS – durchgeführt werden. Routing, Filterung und Transformation von Nachrichten sind die wesentlichen Aufgaben eines Enterprise Service Bus (ESB), der Laufzeit-Komponente, die die Mediator-Services ausführt.

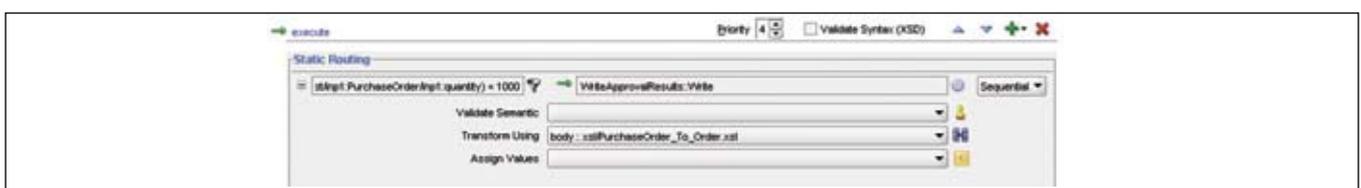


Abbildung 4: Konfiguration eines Mediator-Services mit Oracle JDeveloper 11g

Wie bereits angedeutet, leitet der Mediator-Service eine eingehende Nachricht an den BPEL-Prozess approveLargeOrder weiter. Mit BPEL, einer XML-basierten Prozessautomatisierungssprache können Web Services in einem Prozessfluss orchestriert werden. Im JDeveloper kann diese Orchestrierung grafisch erfolgen.

Nach der Konfiguration und Implementierung aller beteiligten Service-Komponenten können diese als SOA-Kompositum gemeinsam per Deployment aus dem JDeveloper in die Laufzeit-Umgebung übertragen werden. Dieses, gegenüber dem einzelnen Deployment der Komponenten vereinfachte Verfahren, stellt einen weiteren Vorteil des Anwendungsentwicklungsmodells SCA dar.

Sind die Komponenten aus dem Entwicklungstool auf die Laufzeit-Umgebung übertragen, erfolgt der Zugriff mit einer zentralen Web-Oberfläche auf die SCA-Komponenten. Der neue Oracle Enterprise Manager löst damit die verschiedenen Web-Oberflächen der Oracle SOA Suite 10g ab und bietet eine zentrale Sicht auf die Laufzeit-Umgebung. In einer Baumstruktur sind die SCA-Komponenten aufgelistet und können zur genaueren Betrachtung selektiert werden. Die Übersichtsseite eines SCA-Kompositums beinhaltet zum Beispiel eine technische Statistik, eine Liste mit den letzten fünf Instanzen und eine Liste mit den zuletzt aufgetretenen Fehlern.

Auf laufende und abgeschlossene Instanzen kann einerseits über die Übersichtsseite zugegriffen werden, andererseits gibt es einen eigenen Reiter, der alle Instanzen anzeigt. Auf diesem Wege kann der Nachrichten-Fluss durch eine SCA-Komponente nicht nur verfolgt, sondern auch im Detail eingesehen werden (siehe Abbildung 6). Routing-Informationen aus dem Mediator

oder aus BPEL-Prozess-Instanzen können so genauer betrachtet werden. Die Anzeige der BPEL-Prozess-Instanz ist ähnlich der bekannten Darstellung aus der SOA Suite 10g und ermöglicht, den Prozess-Fluss grafisch oder als so genannten Audit Trail (siehe Abbildung 7) in Textform darzustellen.

Eine weitere wichtige Funktion der Konsole ist die Möglichkeit, eine SCA-Komponente zu starten. Diese Art eines Prozessstarts ist nicht für Endbenutzer gedacht, Entwickler können jedoch auf diesem Wege mit wenig Aufwand einen kurzen Test des angebotenen Services durchführen. Im Vergleich zu den Konsolen der SOA Suite 10g erscheint der neue Enterprise Manager aufgeräumter. Insbesondere das Wechseln der Konsolen beim Nachverfolgen des Nachrichten-Flusses oder bei der Suche nach Fehlern entfällt und erleichtert die Entwicklung und das Testen von Service-Anwendungen.

Fazit

Mit der SOA Suite stellt Oracle eine Infrastruktur-Lösung bereit, die die wesentlichen Anforderungen der technischen Implementierung einer service-orientierten Architektur auf der Basis von Web-Service-Technologien unterstützt und umsetzt. Mit der demnächst erscheinenden Version 11g der Oracle SOA Suite wird die Produktsuite im Hinblick auf die integrierte, werkzeuggestützte Entwicklung und Verwaltung von Service-Komponenten verbessert und um einige interessante Technologien und Ansätze erweitert, die in Auszügen in diesem Artikel aufgezeigt wurden. Darüber hinaus wird die SOA Suite 11g weitere neue Komponenten wie Complex Event Processing (CEP) zur Erkennung bestimmter Muster in Eventströmen und ein in die J2EE-Architektur integriertes Business Activity Monitoring (BAM) zur Ermittlung und Visualisierung betriebswirtschaftlich relevanter Kennzahlen enthalten.

Kontakt:

Ingo Meier
 ingo.meier@opitz-consulting.de
 Christian Schmülling
 christian.schmülling@opitz-consulting.de



Abbildung 6: Prozess-Fluss einer BPEL-Prozess-Instanz als Audit Trail

Trace
 Click on a component instance to see its detailed audit trail
 Show Instance ID:

Instance	Type	Status	Time	Composite Instance	Application
receivePO	Service	Completed Successfully	29-Feb-08 14:50:44 CET	POProcessing of 4	POProcessing
routePO	Mediator Component	Completed Successfully	29-Feb-08 14:50:52 CET	POProcessing of 4	POProcessing
approveLargeOrder	BPEL Component	Completed Successfully	29-Feb-08 14:50:52 CET	POProcessing of 4	POProcessing
getCreditCardStatus	Reference	Completed Successfully	29-Feb-08 14:50:51 CET	POProcessing of 4	POProcessing
getCreditCardCC	Service	Completed Successfully	29-Feb-08 14:50:52 CET	validatorForCC of 5	CreditCardValidation
RouteRequest	Mediator Component	Completed Successfully	29-Feb-08 14:50:52 CET	validatorForCC of 5	CreditCardValidation
getCreditValidation	Reference	Completed Successfully	29-Feb-08 14:50:52 CET	validatorForCC of 5	CreditCardValidation

Abbildung 5: Nachrichten-Fluss durch eine SCA-Komponente

Job Notification Package

Autor: Vladimir Poliakov, AREVA GmbH

In Version 10g fügte Oracle den neuen Job Scheduler (`dbms_scheduler`) ein. Dabei durften die Administratoren die E-Mail-Benachrichtigung im OEM (nicht Grid Control) selbst implementieren. Erst deutlich später wurde ein Package zum Download bereitgestellt.

Eine E-Mail zu bekommen, ob der Job erfolgreich war oder nicht, ist heutzutage ein ganz normaler Service, auf den Oracle im neuen Job-Scheduler vorerst verzichtet hat. Später wurde das Problem anerkannt – und auf der Technet-Seite ein Job Notification Package zum Downloaden bereitgestellt. Anspruchsvolle Administratoren werden damit wahrscheinlich nicht zufrieden sein. Für alle anderen, die das Rad nicht neu erfinden möchten, ist folgende Lösung akzeptabel. Das Package kann ab 10gR2 eingesetzt werden und befindet sich als ZIP-Datei unter dem Link <http://www.oracle.com/technology/products/database/scheduler/index.html>.

Installation

Man lädt zuerst die Datei `job_notification.zip` von der Technet-Seite herunter und packt sie in einem Verzeichnis aus. Der Rest ist relativ einfach und in `job_notification_readme.pdf` gut beschrieben. Die Installationsdatei `job_notification.sql` soll vom User `SYS` mit zwei Parametern aufgerufen werden. Diese Parameter sind der SMTP-Server und seine Portnummer (in der Regel 25).

Dieses Skript erstellt ein Utility-Programm `E-Mail_notification_program` und eine Prozedur `E-Mail_notification_procedure` mit entsprechenden Public-Synonymen im Schema `SYS`. Damit wird das Versenden der E-Mails via Package SMTP überhaupt möglich. Abbildung 2 zeigt die Abbildung in der Enterprise-Manager-Web-Konsole.

Außerdem stehen noch die beiden Prozeduren `add_job_E-Mail_notification` und `remove_job_E-Mail_notification` im Schema `SYS` zur Verfügung. Die Prozeduren werden durch die Public-Synonyme angesprochen und dienen zum Steuern (Einfügen oder Löschen) der E-Mail-Benachrichtigungen der Jobs (siehe Abbildung 2, Seite 68).

```
SQL> show user
USER ist 'SYS'
SQL> @job_notification.sql smtp.myserver.com 25
Enter an outgoing e-mail SMTP server host
Enter an outgoing e-mail SMTP server port

PL/SQL-Prozedur erfolgreich abgeschlossen.

Prozedur wurde erstellt.
Keine Fehler.
Benutzerzugriff (Grant) wurde erteilt.

PL/SQL-Prozedur erfolgreich abgeschlossen.

PL/SQL-Prozedur erfolgreich abgeschlossen.

Benutzerzugriff (Grant) wurde erteilt.
alt 9: email_server_host IN VARCHAR2 DEFAULT '&email_server_host',
neu 9: email_server_host IN VARCHAR2 DEFAULT 'smtp.myserver.com',
alt 10: email_server_port IN PLS_INTEGER DEFAULT &email_server_port
neu 10: email_server_port IN PLS_INTEGER DEFAULT 25

Prozedur wurde erstellt.
Keine Fehler.
Synonym wurde erstellt.

Benutzerzugriff (Grant) wurde erteilt.

Prozedur wurde erstellt.
Keine Fehler.
Synonym wurde erstellt.

Benutzerzugriff (Grant) wurde erteilt.
SQL>
```

Abbildung 1: Installation des Packages in SQL*Plus

Arbeiten mit dem Job Notification Package

Wie gesagt, stehen nur zwei Prozeduren zum Steuern der E-Mail-Benachrichtigungen für die Jobs zur Verfügung.

```
EXEC ADD_JOB_E-MAIL_NOTIFICATION('TEST_
JOB', 'to@server.de', 'JOB_SUCCEEDED');
EXEC REMOVE_JOB_E-MAIL_
NOTIFICATION(, TEST_JOB');
```

Der erste Aufruf erstellt einen `TEST_JOB_E-MAILER`-Job, der ein E-Mail versendet, wenn der Job `TEST_JOB` erfolgreich ausgeführt wurde (siehe Abbildung 3). Der zweite Aufruf entfernt den E-Mail-Job aus dem Scheduler. Die anderen Argumente (beispielsweise die E-Mail-Adresse des Absenders) können aus den `*_SCHEDULER_`

Best Practice Job Notification

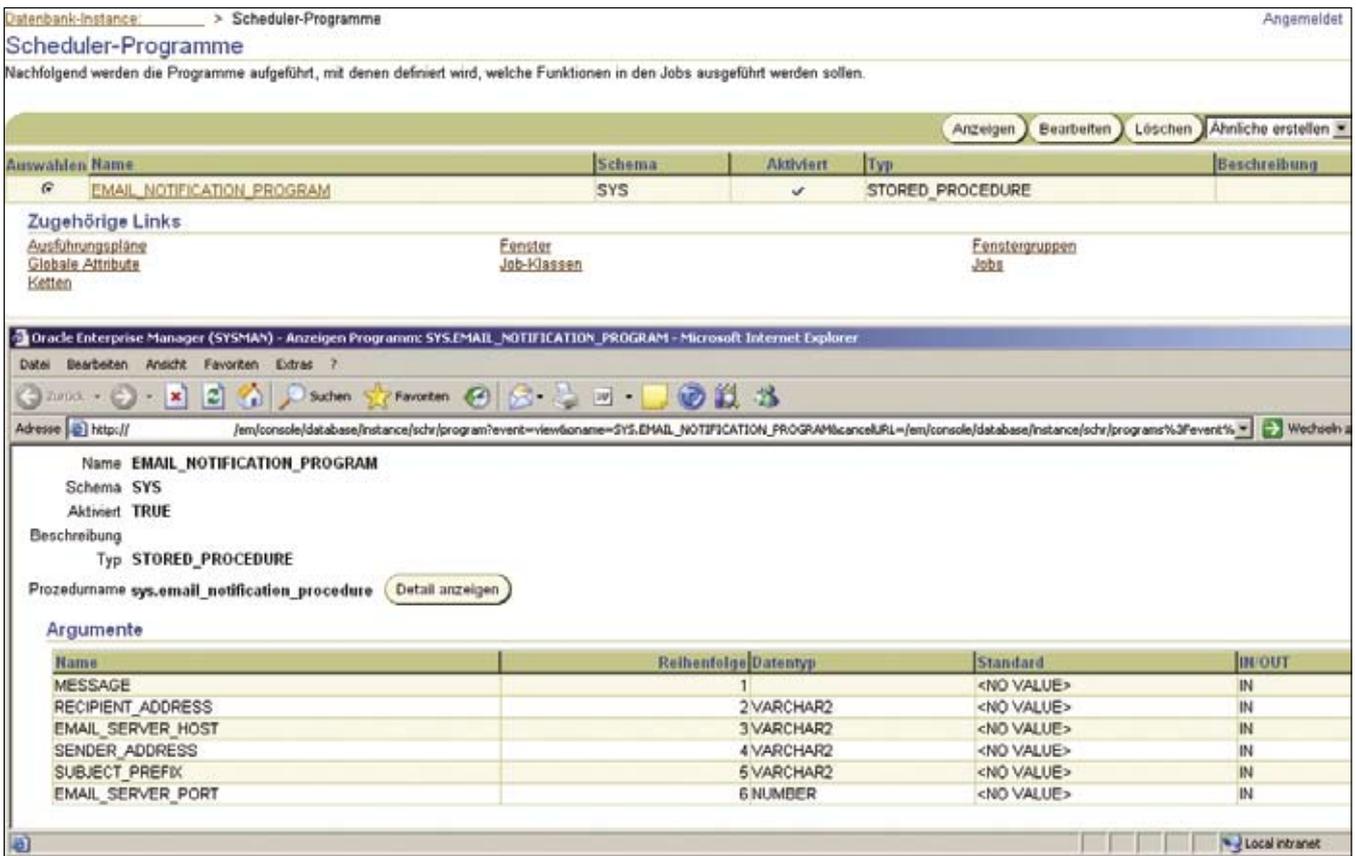


Abbildung 2: Scheduler-Programme nach der Installation des Packages

JOBS- oder *_SCHEDULER_JOB_ARGS-Views ermittelt werden.

Die Job-Namen der E-Mail-Benachrichtigungen werden nach folgendem Muster erstellt: {die ersten 10 Zeichen vom Jobnamen}_E-MAILER{optionale Nummer}. Der Job-Name darf nicht länger als 30 Zeichen sein, sonst erscheint die Fehlermeldung der add_job_email_notification-Prozedur. Diese Fehlermeldung kann man umgehen, indem die Prozedur auf persönliche Bedürfnisse angepasst wird.

Fazit

Mit diesem Package lässt sich relativ viel Zeit sparen. Es hat dennoch einen großen Nachteil: Die Output-Log-Dateien (spool-Datei etc.) können nicht über diese E-Mail-Benachrichtigung verschickt werden.

Kontakt:

Vladimir Poliakov
vladimir.poliakov@areva.com



Abbildung 3: Scheduler-Jobs nach der Definition einer E-Mail-Benachrichtigung

Nichts geht mehr ohne Virtualisierung

Autoren: Lutz Pößneck und Dr. Dietmar Müller, silicon.de

Mehr als die Hälfte aller Großunternehmen weltweit räumt dem Management von virtuellen Server-Umgebungen einen hohen Stellenwert in der IT ein. Die Zahl überrascht: An der Menge der neu auf den Markt kommenden Produkte zum Thema gemessen, sollten es noch viel mehr sein! Offenkundig gibt es noch einen Nachholbedarf in Sachen Informationen rund um die Virtualisierung.

Die wichtigsten Fähigkeiten einer Management-Lösung für die Server-Virtualisierung sind laut einer jüngst von CA in Auftrag gegebenen Studie Performance, Auslastung, Sicherheit und Automatisierung. Die Studie weist darauf hin, dass die größten Vorteile einer Server-Virtualisierung in der einfacheren Hardware-Provisionierung und Software-Entwicklung liegen. Darüber hinaus werden die größere Flexibilität der Entwicklungs- und Testumgebungen sowie die optimale Auslastung der Rechnerleistung als Pluspunkte der Virtualisierung angeführt.

Laut Studie werden vor allem Server, Speicher und Anwendungen virtualisiert. Den größten Erfolg sehen die Befragten bei der Virtualisierung von Servern. 56 Prozent der Befragten nutzen verschiedene Plattformen beziehungsweise Anbieter für das Management der Server-Virtualisierung. Nur 35 Prozent arbeiten auf einer einzigen Plattform.

Als größte Herausforderung des Server-Virtualisierungs-Managements benennen die IT-Führungskräfte den Sicherheitsaspekt. Auch das Messen des Return on Investment (ROI) stellt eine erhebliche Herausforderung dar. Weltweit haben nur 28 Prozent der Befragten eine Methode parat, um den ROI bei Virtualisierungslösungen zu messen. Jedoch sind 51 Prozent der Befragten sehr zuversichtlich, dass ihre Unternehmen den ROI bei ihren Investitionen in die Virtualisierung erhöhen werden.

Wie aus einer anderen Untersuchung hervorgeht, diesmal vom Marktforschungsinstitut IDC, wird Virtu-

Die HUNKLER GmbH & Co. KG ist ein Systemhaus, das sich ausschließlich auf Oracle-Datenbanktechnologien spezialisiert hat. Das 1987 gegründete Unternehmen war der erste offizielle Oracle-Partner in Deutschland.

Das Angebot gliedert sich in den Lizenzvertrieb sowie umfassende Dienstleistungen, von der Planung über die Realisierung bis zur Remote-Administration von Datenbankinfrastrukturen. Zudem unterstützt Hunkler die Kunden bei der Anwendungsentwicklung auf Oracle-Basis. Das Unternehmen ist Oracle Certified Advantage Partner und hat damit den höchsten Partnerstatus inne, den Oracle vergibt.

ORACLE CERTIFIED ADVANTAGE PARTNER

Der Hauptsitz von Hunkler befindet sich in Karlsruhe. Die Bodenseeregion, Österreich und die Schweiz werden von der Niederlassung in Konstanz aus bedient.

Best Solutions Based on Oracle
HUNKLER
GmbH & Co. KG

HUNKLER GmbH & Co. KG

Bannwaldallee 32
76185 Karlsruhe
Deutschland

Tel. (0721) 49 01 6-0
Fax (0721) 49 01 6-29

E-Mail: info@hunkler.de
Web: www.hunkler.de

Mit viel Know-how und Erfahrung rund um Oracle-Technologie haben wir uns am Markt einen Namen gemacht. Auf dieser Basis wollen wir weiter wachsen und suchen Verstärkung im Bereich

DBA für Remote Administration/Hotline m/w

Standort Karlsruhe

DBA Consultant m/w

Standort Karlsruhe oder Konstanz

Ihre Aufgaben

DBA für Remote Administration/Hotline (Innendienst):

- Regelmäßige Remote-Optimierung (Konfiguration/ Performance) der Kundendatenbanken und Anwendungen
- Proaktive Analyse möglicher System-/Netzwerk-schwachstellen, Entwickeln von Lösungswegen
- Bearbeitung von Kundenanfragen (Hotline)

DBA Consultant

- Planung, Konzeption und Umsetzung von Infrastrukturprojekten mit Schwerpunkt Migration, Hochverfügbarkeit
- Durchführung von Workshops und Schulungen
- Mitwirkung bei der Kundenakquise

Was wir von Ihnen erwarten

Für beide Positionen setzen wir ein abgeschlossenes Studium im Bereich Informatik oder eine Ausbildung mit vergleichbarem Schwerpunkt voraus. Fundierte Kenntnisse in Oracle Database 8i, 9i und 10g sind unbedingt notwendig, wünschenswert sind Erfahrungen mit Oracle DataGuard und RAC. Bewerber sollten sich gut mit Windows-/Linux-Betriebssystemen sowie Netzwerk- und Storatechnik auskennen. Die Qualifikation als Oracle Certified Professional ist jeweils wünschenswert. Von Bewerbern um die Position als Consultant wird Reisebereitschaft erwartet.

Was wir Ihnen bieten

Wir wollen Mitarbeiter, die Verantwortung übernehmen, Zusammenhänge erkennen, Ideen und Eigeninitiative in spannende Projekte einbringen. Das honorieren wir – mit abwechslungsreichen Aufgaben, einem leistungsorientierten Gehalt, attraktiven Sozialleistungen und gezielter Weiterbildung für die fachliche und persönliche Weiterentwicklung.

Neugierig geworden?

Dann freuen wir uns auf Ihre ausführliche Bewerbung per E-Mail oder Post mit Angabe der Gehaltsvorstellungen und nächstmöglichem Eintrittstermin.

Trends & Tendenzen

alisierung heute vor allem für Partitionierungsaufgaben verwendet. Clustering und der Einsatz von Virtual Machines stehen ebenfalls auf der Agenda der Anwender. Grid Computing und High Performance Clustering werden eher im High Performance Computing verwendet.

„Alles in allem haben die Kunden aber noch eine gewisse Scheu, die fortgeschritteneren Funktionen zu verwenden, die Virtualisierung bietet“, sagte Chris Ingle, Consulting and Research Director Systems bei IDC. Zunächst müsste die Branche aber verstehen, dass Virtualisierung derzeit noch für jeden etwas anderes bedeute. „Während die Großunternehmen, die mit neuen Techniken Erfahrung haben, ihre Finger auch nach den modernen Funktionen ausstrecken und Neues testen, ist die Reihenfolge bei den durchschnittlichen Anwendern eine andere. Wir sehen oft, dass Virtualisierung meist klein anfängt.“

Zwar sei die Akzeptanzrate hoch, doch die Umfrage habe gezeigt, dass oft erst sieben Prozent der installierten Server virtualisiert seien. Dabei sei die Penetration von Virtualisierung am höchsten in Linux-Umgebungen. 20 Prozent der Linux-Plattformen sei derzeit schon virtualisiert.

Auch Oracle ist spätestens seit der letzten Hausmesse Open World im November 2007 in den virtuellen Ring gestiegen. Dort wurde eine eigene Server-Virtualisierungslösung vorgestellt: ‚Oracle VM‘. Das ist eine Browser-basierte, in den Server integrierte Management-Plattform. Die neue Software unterstützt nicht nur Oracle-Anwendungen, sondern auch diverse Enterprise-Linux-Varianten von Oracle und Red Hat sowie Microsofts Windows-Server 2003 und Windows XP.

Nach Angaben von Oracle-President Charles Philips soll die Software bis zu dreimal effizienter als vergleichbare Produkte sein, beispielsweise von Red Hat oder Microsoft. Man werde – ähnlich wie bei Oracle Unbreakable Linux – nicht das Produkt selbst verkaufen, sondern nur die Supportleistungen dafür. „Virtualisierung ist ein heißes Thema, und unsere Antwort darauf heißt Oracle VM“, sagte Philips.

Wie Oracle weiter bekannt gab, basiert das Produkt auf der Open Source Software ‚Xen Hypervisor‘. Seit dem 14. November steht es auf der Oracle-Webseite zum kostenlosen Download bereit. Für den Support berechnet das Unternehmen mindestens 499 bis 799 US-Dollar pro Jahr.

VMware konterte umgehend und präsentierte mit der Betaversion des VMware Server 2 die nächste Generation seiner Virtualisierungslösung. Sie führt eine neue intuitive Management-Schnittstelle auf Web-Basis ein, die Linux- und Windows-Nutzern eine konsistente Oberfläche für die Verwaltung bietet. Zudem unterstützt die neue VMware Server-Version mehr als 30 Varianten von Gastbetriebssystemen, darunter ver-

schiedene Linux-Distributionen, Windows Server 2003, Windows Server 2008 (Beta) und Windows Vista.

Das mutet fast makaber an, denn langfristig stellt Virtualisierung nicht nur eine Chance für die Anwender dar – sondern auch ein echtes Risiko für Betriebssysteme, egal ob sie quelloffen sind oder nicht. Diese Meinung vertrat kürzlich Mendel Rosenblum, einer der Gründer des Virtualisierungsspezialisten VMware.

Virtual Appliances zusammen mit virtualisierten Lösungen auf Server-Ebene seien eine Bedrohung für die Plattformen, weil sie der unter einer Lösung liegenden Schicht einen Teil ihrer Bedeutung nehmen. Rosenblums Ansicht, die er auf der LinuxWorld 2007 in San Francisco äußerte, war unter den Konferenz-Teilnehmern allerdings nicht unumstritten.

Rosenblum begründete seine Aussage damit, dass eine virtuelle Appliance, die eine Software mit dem Code des Betriebssystems verbindet, damit sie auf einem Server laufen kann, auch anderweitig eingesetzt werden könne. Sie sei in der Lage – und zwar unabhängig vom dort genutzten Betriebssystem – auf jedem virtualisierten Server ebenfalls verwendet werden. Das heißt, ohne vorher noch einmal mit dem dort eingesetzten Operating System verknüpft zu werden. Rosenblum sagte wörtlich, die Betriebssysteme seien ein Konzept für den Papierkorb – „sie werden einfach verschwinden“. Dagegen könnten virtualisierte Pakete die Aufgaben der Plattformen mit übernehmen.

Wie auch immer: Am Thema Virtualisierung kommen kein Unternehmen und kein Rechenzentrum vorbei. Die Auswahl an Konzepten und Lösungen ist groß – kaum ein Anbieter fährt keine große Palette an passender Software auf. Hier sind Augenmaß und ein klares Konzept gefragt.

Kontakt:

Dr. Dietmar Müller
dietmar.mueller@cnet.com



silicon.de ist das Info-Netzwerk für Technologie- und Business-Entscheider. Mit News, Hintergründen und Analysen berichtet silicon.de nicht nur über die aktuellen Wirtschafts- und Technikrends, sondern ordnet diese auch in den für IT-Entscheider richtigen Kontext ein. So wissen silicon.de-Leser schon heute, was ihrem Business morgen den entscheidenden Vorteil verschafft. silicon.de ist offizieller Medienpartner der DOAG. Weitere Infos unter www.silicon.de.

COMPUTERWOCHE

MEHR LEISTUNG. MEHR FLEXIBILITÄT. KEINE GRENZEN.

ERP-INITIATIVE 2008

www.computerwoche.de/erp2008JETZT ANMELDEN UND
GEWINNEN!Unter allen Teilnehmern
verlosen wir eine Tagestour
mit einem Hummer H1.www.computerwoche.de/erp2008FRANKFURT AM MAIN
HILTON HOTEL
27. MAI 2008

Think global, act local: ERP für weltweite Prozesse

Unternehmen, die jetzt über eine neue Business-Software nachdenken, stellt die Globalisierung vor neue Herausforderungen. Mehrsprachigkeit, grenzüberschreitende Hersteller-Lieferanten-Kunden-Beziehungen, internationale Rechnungslegungsvorschriften und viele andere Aspekte erfordern hochflexible ERP-Lösungen.

Die Konferenz der COMPUTERWOCHE ERP-Initiative 2008 zeigt, worauf es ankommt bei der Planung von ERP-Systemen, und bietet **praktische Entscheidungshilfen**

- von aktuellen Markttrends und ihrer Bedeutung für den Anwender
- über den Auswahlprozess und finanzielle Aspekte
- bis hin zur Sicherung der Zukunftsfähigkeit von ERP-Lösungen.

Freuen Sie sich auf eine interaktive Veranstaltung mit renommierten Experten und Anwendern!

Jetzt anmelden: www.computerwoche.de/erp2008



Partner:



Sponsor:



Aussteller:



In Kooperation mit:



Detaillierte Informationen, Agenda und Anmeldung unter:

www.computerwoche.de/erp2008

Oracle Security Corner

Autor: Alexander Kornbrust, Red Database Security GmbH

Da sowohl die DOAG News als auch die Oracle-Security-Patches vierteljährlich erscheinen, ist es natürlich kein Wunder, dass der Blick auf den jeweils letzten CPU zu einer festen Größe dieser Rubrik geworden ist.

Wieder wurden Exploits veröffentlicht, um Oracle-Sicherheitslücken auszunutzen. Dies scheint sich mittlerweile zur Gewohnheit zu entwickeln. Wir haben bereits in der vorherigen Ausgabe darüber berichtet. Aktuell wurde ein Exploit [1] für den Borland Visibroker Smart Agent veröffentlicht, der auf eine remote ausnutzbare Sicherheitslücke zielt. Betroffen sind die älteren Versionen des Oracle Application Servers (Reports, Discoverer). Ein Patch ist nicht vorhanden. Vier weitere Exploits [2] für den Oracle CPU im Januar 2008 wurden wenige Tage nach Erscheinen des Januar CPUs 2008 im Internet veröffentlicht.

Seit Ende Februar ist der Patchset 10.2.0.4 (für Linux x86, 32 bit) erhältlich. Gemäß der Oracle Patch Policy [3] werden Sicherheitslücken zuerst im Quellcode der nächsten Oracle Version (aktuell 11gR2) korrigiert, erst danach erfolgt die Korrektur in den Patchsets der non-terminal Releases (etwa in 10.2.0.4 aber nicht in 9.2.0.8) und zu guter Letzt in den Critical Patch Updates (CPUs). Somit enthält das Patchset 10.2.0.4 korrigierte Sicherheitslücken, die erst mit einem der nächsten CPUs (Juli 2008, Oktober 2008 oder noch später) behoben werden. Demzufolge empfiehlt sich immer das Einspielen des aktuellen Patchsets. Im Übrigen ist in der zitierten Patch Policy die Vorgehensweise von Oracle sehr gut beschrieben.

Große Datenbanken & Security

Da das Schwerpunktthema dieser Ausgabe der DOAG News „Große Datenbanken“ sind, liegt die Überlegung nahe, ob große Datenbanken in puncto Sicherheit anders zu betrachten sind als kleinere. Obwohl bei großen Datenbanken die gleichen Sicherheits-Richtlinien wie bei kleinen oder mittleren Datenbanken zu berücksichtigen sind, gibt es schon einige Besonderheiten. So ist zum Beispiel der Aufwand für Testen und Einspielen von Patches größer und das Zeitfenster meist kleiner. Die Anforderungen an die Verfügbarkeit sind durch die großen Benutzerzahlen und die starke Integration großer Datenbanksysteme in die Betriebsprozesse oft höher als bei den kleineren Datenbanken.

Bei Datenbanken mit vielen Oracle-Benutzern (1.000 bis 20.000) liegt oftmals das größte Sicherheitsproblem darin, die Qualität der Kennwörter sicherzu-

stellen. Erfahrungsgemäß tendieren bis zu 25 Prozent der normalen Anwender dazu, einfache Passwörter (Kennwort=Benutzername, Begriffe wie „Lyoner“, „Ebelwoi“, „Paulaner“, „Fcbayern“ etc.) zu verwenden oder ein einmal zugewiesenes Initial-Kennwort nicht mehr zu ändern. Die genannten Passwörter sind übrigens ein Auszug aus der Trefferliste, die bei verschiedenen Audits aufgefallen sind.

Die Password-Verification-Funktion von Oracle hilft hier aus Kompatibilitätsgründen möglicherweise nicht weiter, da die eingesetzte Applikations-Software (3rd-party-Produkte, Eigenentwicklungen) mit den Return-Codes der Password-Verification-Funktion oftmals nichts anzufangen weiß. Deshalb empfiehlt es sich, die Qualität der Kennwörter regelmäßig zu überprüfen – zum Beispiel mit checkpwd [4]. Auch wenn checkpwd bis zu 600.000 Passwörter pro Sekunde und Benutzer überprüfen kann, stößt man bei einer großen Anzahl von Benutzern auf ein zeitliches Problem. Der Grund liegt darin, dass die gehashten Kennwörter in der Datenbank nicht entschlüsselt werden können. Daher muss „checkpwd“ durch Ausprobieren eine lange Wortliste (Dictionary) abarbeiten und nach Treffern suchen, und dies für jeden Benutzer. Bei der Verwendung von großen Dictionary-Dateien (etwa 1,5 Mio. Kennwörter) in Kombination mit einer großen Benutzeranzahl kann das durchaus bis zu einen Tag in Anspruch nehmen. In solchen Fällen sollte „checkpwd“ mit einer kleineren Dictionary-Datei (<50.000 Einträge) verwendet werden. Leider sinkt dann aber auch die Trefferquote.

Große Data-Warehouse-Systeme sind für Angreifer möglicherweise auch deshalb von starkem Interesse, da dort sehr häufig die interessantesten Daten eines Unternehmens aus mehreren Datenquellen bereits aufbereitet und konsolidiert zur Verfügung stehen. Außerdem ist der Personenkreis (Administratoren und Entwickler) größer, der mit stark privilegierten Accounts Zugriff hat. Daher sollte man genau überprüfen, wer sich wann an die Datenbank mit welchen Tools anmeldet.

Zusätzlich zu den Audit-Möglichkeiten von Oracle kann sehr einfach mithilfe einer externen Tabelle und dem Listener.log [5] eine Überprüfung vorgenommen werden. Eine mögliche Ausgabe in SQL sieht dann folgendermaßen aus:

Service	cnt

C:\InstalledPrograms\Quest Software\TOAD\TOAD.exe	1
C:\Program Files\Actuate7\Server\operation\fctsrvr7.exe	20232
C:\Program Files\Informatica PowerCenter\Client\pmdesign.exe	1
C:\Program Files\Microsoft Office\OFFICE11\EXCEL.EXE	19
C:\oracle\ora102\exp.exe	1
C:\Program Files\Oracle\jre\1.1.8\bin\jrew.exe	6
C:\Program Files\Quest Software\TOAD\TOAD.exe	545
exp@orcl	3
oracle	23
oracle@dwhprod	5
sqlplus	30

Während in diesem Beispiel klar zu erkennen ist, dass die Hauptanwendung auf der Datenbank der Actuate Server (20.232 Connects) ist, sind vor allem die Programme mit wenigen Connects interessant. Mit Excel wird neunzehn Mal auf die Datenbank zugegriffen. Dies könnte eine Anwendung sein, die in der Fachabteilung entwickelt wurde – aber vielleicht auch nicht. Auch der einmalige Zugriff mit dem Export-Utility von Windows auf die Produktions-Datenbank, sollte genauer verfolgt werden, wohingegen der einmalige Connect via Informatica Powercenter eher auf das Testen eines Entwicklers gegen die Produktion hindeutet (muss aber nicht sein). Weiterhin stellt sich die Frage, ob sich Benutzer mit TOAD auf die Produktions-Datenbank anmelden müssen. Letztendlich dreht es sich immer um die Frage: „Ist ein auffälliges Verhalten zu erkennen?“

Das Erstellen dieser Auswertung ist im folgenden Beispiel beschrieben. Es kann auch auf andere Log-Dateien (Apache access_log etc.) angewendet werden. Sie finden das Script auch hier [6] zum Download.

```

spool listener_analysis.txt
prompt set SQL*Plus Parameter
set echo on
set pagesize 0
set linesize 150
prompt create directory
-- Hier muss das Verzeichnis angegeben
werden,
-- in dem sich das Listener.log befindet
create directory LISTENER_LOG_DIR as
,C:\oracle\ora102\NETWORK\log';

prompt Externe Tabelle, die auf das Li-
stenerlog zeigt, anlegen
create table listener_log (
  log_date date,
  connect_string varchar2(300),

```

```

  protocol_info varchar2(300),
  action varchar2(15),
  service_name varchar2(15),
  return_code number(10)
)
organization external (
  type oracle_loader
  default directory LISTENER_LOG_DIR
  access parameters (
    records delimited by newline
    nobadfile
    nologfile
    nodiscardfile
    fields terminated by „*“ ltrim
    missing field values are null (
      log_date char(30) date_format
      date mask „DD-MON-YYYY
HH24:MI:SS“,
      connect_string,
      protocol_info,
      action,
      service_name,
      return_code
    ) )
  location (,listener.log')
)
reject limit unlimited
/

```

```

prompt Parse_function erzeugen
create or replace function parse_liste-
ner_log_line
(
  p_in varchar2,
  p_param in varchar2
)

```

```
return varchar2
as
    l_begin    number(3);
    l_end      number(3);
    l_val      varchar2(2000);
begin
    if p_param not in (
        ,SID', ,SERVICE_
NAME', 'PROGRAM', 'SERVICE',
        ,HOST', 'USER',
,PROTOCOL', 'TYPE',
        ,METHOD', 'RETRIES',
,DELAY', 'PORT', 'COMMAND'
    ) then
        raise_application_error
(-20001, 'Invalid Parameter Value ,||p_
param);
    end if;
    l_begin := instr (upper(p_in),
,(,||p_param||'='));
    l_begin := instr (upper(p_in),
,=', l_begin);
    l_end   := instr (upper(p_in),
,)', l_begin);
    l_val   := substr (p_in, l_
begin+1, l_end - l_begin - 1);
    return l_val;
end;
/
```

```
prompt Run queries against listener.log
col program format a70
col cnt format 999,999

prompt Verschiedene Programme je SID/
Servicename
select lower(service_name) Service,
parse_listener_log_line(connect_
string,'PROGRAM') program, count(1) cnt
from listener_log
group by service_name, parse_listener_
log_line(connect_string,'PROGRAM')
order by 1, 3 desc, 2;
-----
```

```
prompt Verschiedene Benutzernamen
select lower(service_name)
Service, substr(parse_listener_log_
line(connect_string,'USER'),1,70) USER-
```

```
NAME, count(1) cnt
from listener_log
group by service_name, parse_listener_
log_line(connect_string,'USER')
order by 1, 3 desc, 2;
-----
```

prompt Welcher Benutzer hat welches Pro-
gramm wie oft verwendet?

```
select lower(service_name) Service,
substr(parse_listener_log_line(connect_
string,'USER'),1,40) USERNAME,
substr(parse_listener_log_line(connect_
string,'PROGRAM'),1,60) PROGRAM,
count(1) cnt
from listener_log
group by service_name, parse_liste-
ner_log_line(connect_string,'USER'),
parse_listener_log_line(connect_
string,'PROGRAM')
order by 1, 2, 3;
-----
```

Referenzen

- [1] Visibroker Exploit, <http://aluigi.altervista.org/adv/visibroken-adv.txt>
- [2] Exploits für Oracle CPU January 2008, <http://blog.red-database-security.com/2008/01/31/first-exploits-for-cpujan2008-published/>
- [3] Security Vulnerability Fixing Policy and Process, <http://www.oracle.com/technology/deploy/security/securityfixlifecycle.html>
- [4] Oracle Password Checker Checkpwd (kostenlos), <http://www.red-database-security.de/software/checkpwd.html>
- [5] Externe Tabellen, <http://www.dbazine.com/oracle/or-articles/nanda14>
- [6] Beispielcode externe Tabelle & Listener.log, <http://www.red-database-security.com/scripts/analistener.sql>

Kontakt:

Alexander Kornbrust
ak@red-database-security.com

Weitere Informationen zu den Themen
der DOAG News finden Sie unter
<http://www.doag.org/>

Tipps und Tricks aus Gerds Fundgrube

Heute: JavaScript-API in Forms 11g

Autor: Gerd Volberg, OPITZ Consulting GmbH

Forms 11g erlaubt erstmals eine direkte Kommunikation zwischen dem generischen Java-Applet des Browsers und der umgebenden Welt. Das neue JavaScript-API macht dies möglich.

In der neuen Version gibt es zur Kommunikation mit dem JavaScript-API einen neuen Forms-Trigger, neue Systemvariablen und neue Built-Ins. Der Trigger `WHEN-CUSTOM-JAVASCRIPT-EVENT` reagiert auf alle Events, die per JavaScript von außen an Oracle Forms herangetragen werden. Innerhalb dieses Triggers kann man die Payload aus zwei neuen Systemvariablen auslesen: `system.javascript_event_name` und `:system.javascript_event_value`. Informationen, die von einer HTML-Seite an Forms weitergereicht werden, kann man nun ganz einfach auslesen.

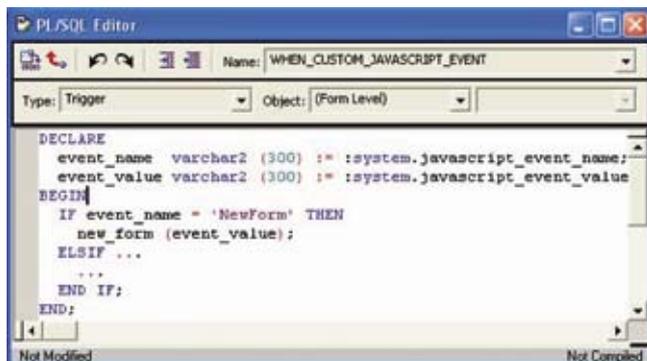


Abbildung 1: Auslesen der Payload in Forms 11g

In diesem kleinen Beispiel wird in der Payload als Event-Name der Befehl `NewForm` übergeben und im Event-Value der Name der Maske. Die Daten werden zum Beispiel wie folgt aus der Internetseite übergeben:

```

<INPUT id="outside_field_id">
<SCRIPT>
  function set_field (field_id, myValue) {
    document.getElementById(field_id).
value=myValue;
  };
  function clickEvent1()
  {
    document.forms_applet.
raiseEvent(„NewForm“, „payload“);
  }
    
```

```

</SCRIPT>
<input id="button1" type="button"
onClick="void clickEvent1();"
value="NewForm">
    
```

Benutzt wird hier die Methode `raiseEvent` der Klasse `forms_applet`. Dieses Applet muss zuvor im OAS in der Konfigurationsdatei `formsweb.cfg` dem Systemparameter `applet_name` zugewiesen worden sein: `applet_name=forms_applet`. Forms ist in der Lage, bidirektional mit dieser Internetseite zu kommunizieren. Dazu nutzt man die neuen Built-Ins `web.javascript_eval_expr` und `web.javascript_eval_function`.

```

web.javascript_eval_expr
  (,document.getElementById(„outside_
field_id“).value="" ||
  :control.ti_inside || „“);
web.javascript_eval_expr
  (,set_field(„outside_field_id“, „‘ ||
:control.ti_inside
  || „“)’);
:control.ti_get_value := web.javas-
cript_eval_function
  (,document.getElementById(„outside_
field_id“).value’);
    
```

Dieser Source-Code befüllt in der Internetseite ein Feld namens `outside_field_id` mithilfe der Built-In `web.javascript_eval_expr`. Zwei Techniken können hierbei genutzt werden: Die direkte Zuweisung oder der Aufruf einer JavaScript-Funktion wie `set_field`. Felder können ausgelesen werden durch die Nutzung der Built-In `web.javascript_eval_function`. Der Return-Wert ist der Wert des entsprechenden Feldes in der HTML-Seite, hier wieder `outside_field_id`.

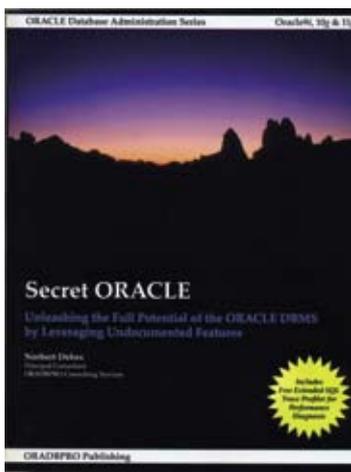
Kontakt:

Gerd Volberg
talk2gerd.blogspot.com

Secret ORACLE

Unleashing the Full Potential of the ORACLE DBMS by Leveraging Undocumented Features

rezensiert von Jaroslav Dutov



Bitte fragen Sie Ihren Arzt oder Apotheker, ob Sie dieses Buch lesen sollten – es ist nämlich ein gefährliches Buch. Ein Buch über dunkle Geheimnisse und geheime Verstecke. Aber gleichzeitig ist es ein Buch, das uns hilft, Oracle DBMS besser zu verstehen und leichter damit zu arbeiten. Das Buch besteht aus zwölf Teilen, die sich

mit den verborgenen Ecken des Oracle DBMS beschäftigen: Initialisierungsparameter, Data Dictionary Tabellen, Events, X\$-Tabellen, SQL-Statements, PL/SQL-Packages, Applikationsentwicklung, Performance, Oracle Net, RAC und Utilities.

Um es leichter lesen zu können, benötigt man tiefere Oracle-DBMS-Kenntnisse. Einzig für das Kapitel 22 „Perl DBI and DBD::Oracle“ reicht ein allgemeines Verständnis über Oracle-Datenbanken aus. Wie der Autor in der Einleitung verspricht, geht es im Buch ohne lange und meist überflüssige Vorgeschichten direkt zur Sache. Die Performance ist das Leitmotiv des Buchs, obwohl nur Teil VIII „Performance“ heißt. Die restlichen Teile diskutieren die Performance-relevanten Themen jedoch reichlich. Index-Monitoring, pga_aggregate_target-Tuning, Events 10027, 10046, 10053, trcsess, hot-Blocks-Ermittlung über X\$BH und dba_objects, oradebug Utility – darüber liest jeder DBA immer gerne etwas!

Im Teil VIII „Performance“ wird detailliert über das extended SQL-Trace-File-Format berichtet. Zudem ist eine Referenz zum extendend SQL-Trace-Profiler esqltrcprof im Angebot. Esqltrcprof ist ein vom Autor selbst erstelltes Tool, um das extended SQL-Trace-File leicht und bekömmlich zu formatieren. Im Kapitel 25 „Statspack“ handelt es sich, wie bereits der Name sagt, um Statspack. Es ist sehr interessant, über das Auffinden von Snapshots mit hohem Ressourcen-Verbrauch zu lesen.

Eine angenehme Überraschung ist die ausführliche Beschreibung der MERITS-Performance-Optimierungsmethode im Kapitel 28. Hervorragend ist, dass es zum Buch ein reichlich gefülltes Source-Code-Depot gibt. Dort findet man Skripte, die für den DBA sehr hilfreich sein werden – unter anderem auch den extendend SQL-Trace-Profiler esqltrcprof.

Fazit

Das Buch liefert eine große Menge vielfältiger Informationen. Teile davon sind zwar im Internet zu finden, aber die ausführlichen, auf praktischen Erfahrungen basierenden Kommentare des Autors sowie die zahlreichen Beispiele sprechen eindeutig für die Anschaffung des Buchs. Aber Vorsicht! In Ihre Hände gerät ein hochexplosiver Stoff. Gehen Sie sehr vorsichtig damit um ...

Kontakt:

Jaroslav Dutov
j.dutov@dbtotal.de

Titel:	Secret ORACLE Unleashing the Full Potential of the ORACLE DBMS by Leveraging Undocumented Features aus der ORACLE Database Administration Series, Oracle 9i, 10g, 11g
Autor:	Norbert Debes, Principal Consultant, ORADBPRO Consulting Services
Verlag:	ORADBPRO Publishing
Seitenzahl:	402
Sprache:	Englisch
ISBN:	1-4357-0551-3

Oracle Designer – Tool mit Zukunft?

Autor: Christian Schwitalla

Das erste Treffen der SIG Development in diesem Jahr fand am 21. Februar 2008 in Dresden statt. Der Titel lautete „Quo vadis Oracle Designer User?“

Der Designer wird seit Jahren erfolgreich in zahlreichen Oracle-Projekten eingesetzt. Anwender schätzen das breite Funktionsspektrum des Tools. Dank der Teamfähigkeit und des eingesetzten Repositories ist der Designer in vielen Projekten zu dem zentralen Werkzeug im Software-Entwicklungsprozess geworden. Leider nimmt seit Jahren die Unzufriedenheit über Oracles Strategie zu: Der Hersteller hat angekündigt, dass der Designer nicht weiterentwickelt wird – ohne eine klare Aussage über ein Nachfolge-Produkt zu treffen. Anwender kritisieren diese intransparente Produktpolitik. Inzwischen wird offen darüber diskutiert, ob der Investitionsschutz für die im Designer erfassten Daten noch gewährleistet ist. Entsprechend hoch war die Beteiligung an der Veranstaltung.

Im ersten Vortrag „Datenmodellierung und Forms-Generierung aus dem Designer“ berichtete Rolf Wesp, Trivadis AG, über den Einsatz des Tools in einem Langläufer-Projekt bei einem mittelständischen Automobil-Zulieferer. Nach einer kurzen Projektvorstellung und der Designer Konfiguration kam einer der beiden Haupteinsatzgebiete des Designers zur Sprache: der Datenmodell-Entwurf. Der Schwerpunkt des Vortrags lag hingegen auf dem zweiten Schwerpunkt: die Forms-Generierung. Hier wurde das Vorgehen sehr detailliert beschrieben. Fazit des Dozenten: „Das Projekt entspricht nicht der reinen Lehre, ist aber ein pragmatischer Ansatz mit sehr guter Kosten-/Nutzen-Relation“. Zum Abschluss stellte Rolf Wesp seine Erwartungen an ein Nachfolgeprodukt vor. Die Schlüsselworte waren: Investitionsschutz, Funktionalität und Handhabung. Unter dem Strich lässt sich sagen, dass der von Rolf Wesp vorgestellte Einsatzbereich des Oracle Designers als typisch bezeichnet werden kann, da das Werkzeug in ähnlicher Art und Weise in vielen Projekten eingesetzt wird.

Weiter ging es mit dem Vortrag „Nutzung der Designer Metadaten für die ADF-Entwicklung“ von Jürgen Menge, Oracle Deutschland. Der Redner stellte eine weitere Nutzungsmöglichkeit für die im Designer Repository erfassten, oft sehr wertvollen Unternehmensdaten vor: das Generieren von ADF-Anwendungen. Mit ADF stellt Oracle ein JEE-Anwendungsentwicklungs-Framework zur Verfügung, das die Programmierung von Java-Masken stark vereinfacht. Die ADF-Entwicklung kann

auch Meta-Daten-basierend erfolgen, analog zu der Designer gestützten Forms-Entwicklung (mithilfe des Forms-Generators, siehe erster Vortrag). Die Komponente dafür trägt den Namen Oracle JHeadstart (JHS) und ist als Plug-In für den JDeveloper verfügbar.

Der JHS-Forms-Migrator kann bestehende Forms-Masken in ADF-Masken umwandeln, allerdings ist dazu die Existenz der Modul-Definition für die Forms-Masken im Designer notwendig. Nach der kurzen Beschreibung des ADF-Technologie-Stacks (insbesondere wurde hier auf die Punkte eingegangen, die für Forms-Entwickler wichtig sind) stellte Jürgen Menge recht detailliert alle Schritte der Forms-ADF-Migration vor. Anschließend folgte eine Live-Demo. Fazit von Jürgen Menge: Die Arbeitsweise von JHeadstart ist Designer/Forms-Entwicklern vertraut. Gründe dafür sind:

- Deklarative Arbeitsweise
- Verwendung ähnlicher Bezeichnungen wie im Oracle Designer
- Konzept der Anwendungs-Generierung

Im nächsten Vortrag „Nutzung und Erweiterung des Designer-API“ ging Ralph Hoppe, Hoppe Dialog GmbH, recht detailliert auf die Designer-Internia ein: API, Repository und Meta-Modell. Das Designer API ermöglicht den Zugriff auf die im Designer gespeicherten Daten aus anderen Programmen/Werkzeugen. Beschrieben wurden die Funktionalität, die Privilegien und Zugriffsrechte sowie die Packages, um nur einige Hauptpunkte zu nennen. Anhand von Beispielen nannte Ralph Hoppe auch einige Einsatzmöglichkeiten des direkten Zugriffs auf die Designer-Daten:

- Migration von Daten
- Korrekturen und Ergänzungen im API
- Hilfreiche Routinen (Synonyme anlegen, Views aus Tabellen ableiten etc.)
- Source-Code vergleichen

Besonders der letzte Punkt weckte das Interesse der Teilnehmer. Dazu präsentierte Ralph Hoppe ein selbstentwickeltes Tool. Es führt eine Syntax-Analyse der PL/SQL-Einheiten durch und vergleicht anschließend die Syntax-Bäume. Damit lassen sich auch dann korrekte

Vergleiche ausführen, wenn beispielsweise die Groß-/Kleinschreibung dies sonst verhindert hätte.

Einen interessanten, praxisnahen Einsatzbericht präsentierten Gabriele Friedel und Dr. Peter Koch in ihrem Vortrag „100-prozentige Generierung von Forms-6i-Masken bei der Nordrheinischen Ärzteversorgung“. Zunächst wurde der Einsatzbereich des Designers vorgestellt: Datenbank-Modellierung und Forms-Masken sowie Menü-Generierung. Die Referenten sehen folgende Vorteile in dieser Lösung:

- Entwicklung und Dokumentation in einem Tool
- Qualitativ hochwertige Forms-Masken mit minimalem Aufwand
- Einheitliches Look&Feel
- Konzentration auf das Wesentliche bei der Masken-Erstellung
- Einfaches Deployment
- Client-/Server-Architektur statt einer 3-Tier-Architektur

Da die Software ausschließlich im lokalen Netzwerk eingesetzt wird, ist man beim Client-/Server-Modell geblieben. Eine Evaluierungsphase der 3-Schichten-Architektur hat keine signifikanten Vorteile aufgezeigt. Einige SIG-Teilnehmer haben von ähnlichen Situationen in ihren Projekten berichtet. Durch Verzicht auf ausgefallenes Layout ist es im Projekt tatsächlich gelungen, die aus dem Designer generierten Forms-Masken nahezu ohne Nacharbeitung zum Einsatz zu bringen. Die Dozenten berichteten, dass sie sich mit den Anwendern darüber einig sind, dass der Funktionalität der Masken die höchste Priorität einzuräumen sei. Als Fazit haben sie folgenden Ausblick vorgestellt:

- Alternativen zum Designer sind nicht in Sicht.
- Zunächst ist die Weiterbenutzung der bisherigen Lösung geplant.
- Unterstützung neuer DB-Features wird in Zukunft wahrscheinlich problematisch werden.

Im nächsten Vortrag verglich Peter Sechser, PITSS GmbH, den Oracle Designer mit dem hauseigenen Produkt PITTS.CON. Obwohl die Tools häufig in denselben Projekten zu finden sind, ist deren Einsatzbereich doch sehr unterschiedlich. Laut Peter Sechser kann man den Designer in Kurzform folgendermaßen beschreiben:

- Modellierungs-Tools
- Ausgeprägte Fähigkeiten zur Anwendungs-Generierung
- Schwerpunkte in Business-Prozess-Analyse oder Datenmodellierung
- Top-Down-Vorgehen

Dagegen setzt PITTS.CON bestehende Forms-Anwendungssysteme voraus und kommt dann zum Einsatz, wenn es um Migration und Weiterentwicklung geht:

- Plattform für hochautomatisierte Forms-Migrationen und Weiterentwicklungs-Projekte
- Umfassende Analyse der Applikationen
- Erstellen vollständiger Abhängigkeits-Bäume
- Leistet Unterstützung beim Bearbeiten des von Designer generierten Codes

Beide Produkte verfügen über ein eigenes Repository in der Datenbank. Zum Abschluss präsentierte Peter Sechser eine ausführliche Demo.

Im nächsten Vortrag „TOOLBUS-Schnittstellen für den Oracle Designer“ stellte Josef Reischmann, Reischmann Informatik GmbH, eine Lösung vor, um Daten zwischen dem Oracle Designer und anderen Modellierungswerkzeugen auszutauschen. Die TOOLBUS-Schnittstelle, eine Eigenentwicklung von Reischmann, wird weltweit eingesetzt und beherrscht den Datenaustausch zwischen zahlreichen CASE-Tools, wie Sybase PowerDesigner oder IBM Rational Rose. Josef Reischmann beleuchtete anhand von Diagrammen den Vorgang beim Datenaustausch.

Im letzten Vortrag „Datenmodell-Connectivity“ präsentierte Jens Gassner, InterFace AG, einen Praxisbericht. In dem Projekt wurden die TOOLBUS-Schnittstellen (siehe vorheriger Vortrag) eingesetzt, um Daten zwischen Borland Together, Magic Draw und Oracle Designer auszutauschen.

Jens Gassner stellt die Problematik der Datenbank-Entwürfe in UML beziehungsweise der Transformation von UML-Klassendiagrammen in ER-Diagramme detailliert vor. Auch die umgekehrte Richtung des Datenaustauschs, sprich von ER-Diagrammen hin zu UML-Klassendiagrammen wurde im Projekt implementiert. Beim Datenaustausch kommt hier XMI als Austauschformat von UML zum Einsatz. Oracle Designer agiert als Integrationsplattform, die UML-Modelle zu Datenbank-Modellen zusammenführt. Im Projekt wird die Zukunft des Designers als relativ unsicher bezeichnet. Jens Gassner berichtete von Überlegungen, einen Designer-Nachfolger zu erproben.

Zum Abschluss fand eine Podiumsdiskussion über die Zukunft des Designers statt. Aus einer Umfrage der DOAG geht hervor, dass für rund 23 Prozent von 334 Unternehmen der Oracle Designer ein strategisch wichtiges Tool in der Software-Entwicklung darstellt. Weitere 30 Prozent halten das Entwicklungswerkzeug für wichtig. Die DOAG nahm das Ergebnis zum Anlass, Oracle auf die Bedeutung des etwas in Vergessenheit geratenen Produkts hinzuweisen und eine bessere Unterstützung der Anwender anzumahnen.

Im Rahmen der Diskussionsrunde betonten Fried Saacke, Vorstandsvorsitzender der DOAG und Christian Schwitalla, Leiter der SIG Development, dass „zahlreiche Unternehmen das Produkt einsetzen, obwohl es technisch nicht mehr auf dem neuesten Stand ist“. Die DOAG wertete die Umfrage als klares Signal an Oracle, „solche Werkzeuge weiterhin den Kunden zur Verfügung zu stellen“ und appellierte an den Hersteller, „auch in diesem Bereich die Investitionen der Kunden zu schützen“.

Mehrere betroffene Anwender bemängelten, „dass keine neuen Features entwickelt wurden, dass es keine guten Patches mehr gab, dass entsprechend ausgebildete Mitarbeiter im Support fehlten und dass Oracle keine alternativen Tools im Angebot habe“.

Günther Stürmer, Vice President Server Technology & SalesConsulting der ORACLE Deutschland GmbH, hatte das Problem bereits im US-Headquarter kommuniziert und stellte den Diskussionsteilnehmern „noch in diesem Jahr Erfreuliches“ in Aussicht. Konkrete Aussagen konnte er jedoch noch nicht machen.

Die Umfrage der DOAG sowie die Vorträge der Veranstaltung stehen unter <http://www.doag.org/termine/222243> zum Download bereit.

An dieser Stelle einen herzlichen Dank an alle Dozenten für die interessanten Beiträge. Die nächste SIG Development findet am 6. Juni 2008 in Berlin statt. Das Thema lautet „Data Services mit Oracle“, siehe <http://www.doag.org/termine/222244>.

Kontakt:

Christian Schwitalla
sig-development@doag.org

Impressum

DOAG News Ausgabe Q2/2008
ISSN 0936-0360

Herausgeber:

DOAG – Deutsche ORACLE
Anwendergruppe e.V.
Tempelhofer Weg 64
12347 Berlin
Tel.: 07 00 11 - 36 24 38
Fax: 07 00 11 - 36 24 39
E-Mail: office@doag.org

Verlag:

DOAG Dienstleistungen GmbH
Gerhard Andreas Schreiber
Geschäftsführer
E-Mail:
info@doag-dienstleistungen.de

Chefredakteur (VisdP):

Wolfgang Taschner
redaktion@doag.org

Chef vom Dienst (CvD):

Carmen Al-Youssef
office@doag.org

Gestaltung und Satz:

DOAG Dienstleistungen GmbH
Claudia Wagner

Druck:

Parzeller Druck- und Medien-
dienstleistungen
GmbH & Co. KG
36043 Fulda
Tel. 0661 - 280-0
Internet: www.parzeller.de

Anzeigen:

DOAG Dienstleistungen GmbH
Carmen Al-Youssef
Tempelhofer Weg 64
12347 Berlin
Tel.: 07 00 11 - 36 24 38
office@doag.org

Mediadaten und Preise

finden Sie unter:
<http://www.doag.org/doagnews/>



Oracle 11g Neuheiten

Sie sind ein erfahrener DBA oder Oracle-Entwickler?
Möchten Sie sich intensiv mit der Version 11g beschäftigen?

In diesem 5-tägigen Seminar werden die neuen Funktionen von Oracle 11g ausführlich behandelt. Schwerpunkte sind dabei SQL, PL/SQL, Administration, Speicherverwaltung, Migration, Security, Performance Tuning, Backup & Recovery und Hochverfügbarkeit. Zahlreiche Übungen und Demos helfen Ihnen, die neuen Konzepte zu beherrschen. Zudem zeigen wir Stärken und Schwächen der Version 11g auf.

Termine:

- 02.06.2008 - 06.06.2008 in Wiesbaden
- 11.08.2008 - 15.08.2008 in Wiesbaden
- 17.11.2008 - 21.11.2008 in Wiesbaden

Gebühren/Teilnehmer: 1.890,00 € (zzgl. MwSt.)

15 % Rabatt

für DOAG-Mitglieder auf alle ORDIX Seminare

Detaillierte Inhalte und weitere Seminare finden Sie im ORDIX Trainingsshop. Melden Sie sich direkt an unter <http://training.ordix.de>

Gerne planen wir auch individuelle Seminare für Sie!
Sprechen Sie uns an: training@ordix.de

Die DOAG bezieht neue Geschäftsräume

Autor: Wolfgang Taschner

Der Berlin-Neuköllner Bezirksbürgermeister Heinz Buschkowsky, Repräsentanten von Oracle, der Vorstand sowie einige Mitglieder und Gäste der DOAG eröffnen gemeinsam die neuen Geschäftsräume in Berlin. In der angeschlossenen DOAG-Konferenzlounge werden künftig eigene Veranstaltungen durchgeführt.

Am 4. April 2008 hat die DOAG im Tempelhofer Weg 64 in 12347 Berlin ihre neuen Geschäftsräume in der 6. Etage bezogen. Den Mitarbeiterinnen und Mitarbeitern der DOAG-Geschäftsstelle, die von der DOAG Dienstleistungen GmbH betrieben wird, stehen nun insgesamt sieben große und helle Büroräume sowie ein Besprechungszimmer zur Verfügung. Darüber hinaus kann die DOAG in der angeschlossenen Konferenzlounge künftig eigene Veranstaltungen durchführen.



Kleine Stärkung nach der Eröffnungsfeier

Im Rahmen der feierlichen Eröffnung hob Fried Saacke, Vorstandsvorsitzender der DOAG, die Bedeutung der neuen Geschäftsräume hervor: „Die DOAG hat damit einen weiteren Meilenstein auf ihrem Expansionskurs erreicht. Die wachsende DOAG Dienstleistungen GmbH kann ihr Angebot erweitern. Und wir werden die neue Konferenzlounge zum Erfahrungsaustausch und zur Wissensvermittlung für alle Oracle-Anwender intensiv nutzen.“

Der Berlin-Neuköllner Bezirksbürgermeister Heinz Buschkowsky verwies in diesem Zusammenhang auf die große Bedeutung des Wirtschaftsstandortes Neukölln: „Die DOAG befindet sich in guter Gesellschaft mit vielen weltweit aktiven Unternehmen.“

Als Vertreter von Oracle freute sich Günther Stürner, Vice President Server Technology & SalesConsulting



Networking im Vorraum der DOAG-Konferenzlounge

der ORACLE Deutschland GmbH, dass sich die DOAG um die Wissensvermittlung kümmert: „Es ist das Wissen, das alle Oracle-Anwender weiterbringt. Die DOAG ist auf dem besten Weg zum innovativsten Wissensvermittler im Oracle-Umfeld.“

Fotos: Wolfgang Taschner



Von links: Heinz Buschkowsky, Bezirksbürgermeister von Berlin-Neukölln; Gerhard Andreas Schreiber, Geschäftsführer der DOAG Dienstleistungen GmbH; Günther Stürner, Vice President Server Technology & SalesConsulting, ORACLE Deutschland GmbH; Fried Saacke, Vorstandsvorsitzender der DOAG

Wir begrüßen unsere neuen Mitglieder!

Persönliche Mitglieder

Christian Ampletzer	
Thomas Beetz	NetAge Solutions GmbH
Frank Beutelschiess	
Christine Brand	
Eilert Brinkmeyer	walter services Administration GmbH
Markus Flechtner	
Andreas Gatz	NetConcept
Fred Gorczak	BALVI Software GmbH
Oliver Gräfe	12snap
Klaus Hipp	Bucher Hydraulics GmbH
Ewald Hülsing	Gaz de France PEG
Michael Lampeitl	
Theofil Lepschy	BucherHydraulics GmbH
Jörg Lewe	allesklar.com AG
Peter Morcinek	
Ulrich Müller	
Andreas Neuhauser	
Perry Pakull	
Oliver Radmann	J.R. Weichelt GmbH
Michael Riester	web-ok internetservices GmbH
Norbert Rüth	
Florian Schafhauser	
Norbert Schlüter	
Alexander Schmidt	Comline AG
Michael H.G. Schmidt	Toorit Ltd.
Danilo Schmiedel	HTWK-Leipzig
Sabine Teuber	
Karl-Heinz Urban	
Hubert Weyers	
Peter Wilkens	ELANTAS Beck GmbH
Jürgen Witsch	

Firmenmitglieder

Laurent Bigenwald	Celesio AG
Frank Bochtler	DFS Deutsche Flugsicherung GmbH
Günther Eberle	Liechtensteiner Landesverwaltung
Peter Elliott	Fortis Deutschland Lebensversicherung AG
Petra Freutel	Mettenmeier GmbH
Heiko Fricke	evosoft business relations GmbH
Andreas Frommhold	Arkema GmbH
Xiaoyun Gao	Jeppesen GmbH
Torsten Gerken	Fortis Deutschland Lebensversicherung AG
Uwe Gomoll	H&R InfoTech GmbH
Jens Gorman	Kühne + Nagel (AG & Co.) KG
Ursula Gossen	NGK Spark Plug Europe GmbH
Wolfgang Herz	DFS Deutsche Flugsicherung GmbH
Klaus Hoeper	Enthone GmbH
Klaus Höling	Sennheiser electronic GmbH & Co. KG
Marco Holst	delphi BusinessInformation Consultants GmbH
Michaela Höpfner	
Tobias Indlekofer	DEKRA Arbeit GmbH
Rainer Jurgies	Bayer Business Services GmbH
Monika Kern	NEXUS / INOVIT GmbH
Andy Kielhorn	PLATH GmbH

Holger Klatt	Kassenärztliche Vereinigung Thüringen
Martin Klier	A.T.U Auto-Teile-Unger Handels GmbH u. Co KG
Werner Köpke	SGS INSTITUT FRSENIUS GmbH
Peter Kossmann	Bayer Business Services GmbH
Andreas Kronsteiner	Utah Informationsservice GmbH
Andreas Lein	Kieback & Peter GmbH & Co. KG
Christian Maßnick	Fourth Project Consulting GmbH
Reinhard Mense	ARETO Consulting GmbH
Claudio Mertz	Fortis Deutschland Lebensversicherung AG
Georg Metz	EBIS valuationManagement GmbH
Wolfgang Miebs	Hamburg Süd
Uwe Pauli	ESAB GmbH
André Probst	alphaQuest GmbH
Dr. Heike Püttmann	Landeshauptstadt München - Direktorium
Johannes Reinert	kohlpharma GmbH
Thomas Reulbach	JVS Straubing
Johannes Rinke	Comline AG
Oliver Schäff	noris network AG
Frank Schaffhauser	Horstick it-solutions
Guenter Schneider	InforSense
Alexander Sorge	Rudolf Wild GmbH & Co. KG
Torsten Spieckermann	Informatics Consulting GmbH
Georg Stowasser	Biotronik GmbH & Co. KG
Oliver Tiemann	Hellmann Worldwide Logistics GmbH & Co. KG
Klaus Dieter Vogt	Merck KGaA
Wolfgang Werz	Erbe Elektromedizin GmbH
Steffen Wolter	Wolff Consulting GmbH
Dirk Zapp	NürnbergMesse GmbH

Vorschau auf die nächste Ausgabe

Das Schwerpunkt-Thema der Ausgabe Q3/08 lautet:
„Entwicklung“

Darunter fallen ein ganze Reihe unterschiedlicher Aspekte, die von der Business-Seite bis hin zur rein technischen Schiene angesiedelt sind:

- Geschäftsprozessmodellierung (insbesondere ARIS, ein Hauptbestandteil der Oracle BPA Suite)
- Referenzmodelle für Geschäftsprozesse
- Model Driven Software Development
- UML-Modellierung mit dem JDeveloper
- ER-Modellierung: Was kommt nach dem Designer? Der JDeveloper, der SqlDeveloper, 3rd-party Tools?

Darüber hinaus finden Sie wie in jeder Ausgabe:

- Fachartikel aus dem Bereich Datenbank
- Best Practice im Umgang mit den Oracle-Produkten
- Aktuelle Informationen zu Veranstaltungen der DOAG
- Neues von Oracle

Redaktionsschluss ist am 13. Juni 2008.

Die Ausgabe Q3/2008 erscheint am 15. August 2008.

Mai

Donnerstag, 15.05.2008
SIG JD-Edwards

Kontakt: Uwe Gomoll
sig-jde@doag.org

Mittwoch, 21.05.2008
Regionaltreffen NRW

Kontakt: Dierk Lenz
regio-nrw@doag.org

Montag, 26.05.2008
**Regionaltreffen Osnabrück/Bielefeld/
Münster**

Thema: Security
Kontakt: Wilhelm Breßer
regio-osnabrueck@doag.org

Dienstag, 27.05.2008
**Nordlichtertreffen Bremen/Hamburg/
Hannover**

Kontakt: Michael Paeye, Henrik Rüniger,
Ralf Kölling
regio-nord@doag.org

Juni

Dienstag, 03.06.2008
DOAG ITIL Day
Thema: Anforderungen an den Betrieb
von Oracle Systemen
Kontakt: Jörg Hildebrandt, Frank Stöcker
sig-til@doag.org

Mittwoch, 04.06.2008
Regionaltreffen Berlin/Brandenburg
Thema: Oracle Application Server &
Oracle Spatial / 3D Stadtmodell
Kontakt: Michel Keemers
regio-berlin@doag.org

Donnerstag, 05.06.2008
SIG Development
Thema: Data Services mit Oracle: SDO,
TopLink, Coherence, ADF Data Binding ...
Kontakt: Christian Schwitalla,
Andreas Badelt
sig-development@doag.org

Montag, 09.06.2008
SIG ORACLE und SAP
Kontakt: Jörg Hildebrandt
sig-sap@doag.org

Montag, 09.06.2008
Regionaltreffen München/Südbayern
Thema: Security

Kontakt: Dr. Dietmar Neugebauer,
Andreas Ströbel
regio-muenchen@doag.org

Dienstag, 10.06.2008
Regionaltreffen Rhein/Main

Kontakt: Thomas Tretter
regio-rhein-main@doag.org

Donnerstag, 12.06.2008
SIG Database
Thema: Backup & Recovery

Kontakt: Christian Trieb
sig-database@doag.org

Dienstag, 17.06.2008
SIG Data Warehouse
Thema: Data Warehouse Operations & BI
Infrastructure

Kontakt: Christian Weinberger
sig-dwh@doag.org

Mittwoch, 18.06.2008
SIG Siebel

Kontakt: Hans-Jörg Wiebe
sig-siebel@doag.org

Mittwoch, 18.06.2008
SIG Database
Thema: Backup & Recovery
(Wiederholungsveranstaltung
vom 12.06.2008)

Kontakt: Christian Trieb
sig-database@doag.org

Mittwoch, 18.06.2008
Regionaltreffen NRW

Kontakt: Stefan Kinnen
regio-nrw@doag.org

Donnerstag, 19.06.2008
SIG Development
Thema: Data Services mit Oracle: SDO,
TopLink, Coherence, ADF Data Binding ...
(Wiederholungsveranstaltung
vom 5.6.2008)

Kontakt: Christian Schwitalla,
Andreas Badelt
sig-development@doag.org

Donnerstag, 19.06.2008
Regionaltreffen Rhein-Neckar

Kontakt: Kai Christianus
regio-rhein-neckar@doag.org

Freitag, 20.06.2008
SIG Development: Active Workshop
Thema: Data Services mit Oracle: SDO,
TopLink, Coherence, ADF Data Binding ...

Kontakt: Christian Schwitalla,
Andreas Badelt
sig-development@doag.org

Juli

Dienstag, 08.07.2008
Regionaltreffen München/Südbayern
Thema: Backup & Recovery

Kontakt: Dr. Dietmar Neugebauer,
Andreas Ströbel
regio-muenchen@doag.org

Dienstag, 15.07.2008
Regionaltreffen Jena/Thüringen
Thema: Komprimierung von DB-Objekten:
Technologie, Verfahren, Nutzen

Kontakt: Jörg Hildebrandt
regio-jena@doag.org

Donnerstag, 17.07.2008
Regionaltreffen Stuttgart
Kontakt: Jens-Uwe Petersen
regio-stuttgart@doag.org

Dienstag, 22.07.2008
Regionaltreffen Freiburg/Südbaden
Kontakt: Volker Deringer
regio-freiburg@doag.org

Samstag, 26.07.2008
Regionaltreffen Trier/Saarland
Thema: Needful Things – Tipps für den
Teilzeit DBA

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

Weitere, aktuelle Informationen
und Details finden Sie in unserem
Online-Terminkalender unter
www.doag.org/termine