

ORDIX news

Das IT-Magazin der ORDIX AG



Klassisches vs. agilem Projektmanagement

18 | Ein Vergleich zwischen den beiden Projektmanagement-Philosophien

- 6 | Architekturbewertung - Die richtige Betrachtung der Softwarearchitektur
- 11 | Neue Reihe: Oracle Cloud Control 12c - Über den Wolken...
- 33 | Neue Reihe: Oracle Solaris 11 - Neue Funktionen - Boot-Umgebungen
- 36 | Hibernate für Fortgeschrittene - Die SubSelect-Fetching-Strategie



Effizientes Projektmanagement ist neben der notwendigen Fachkompetenz der entscheidende Faktor für erfolgreiche Projekte. Für die Planung und Abwicklung von Projekten gibt es unterschiedliche Ansätze. In unserem Programm bieten wir sowohl Seminare zum klassischen als auch zum agilen Projektmanagement an.

IT-Projektmanagement - Methoden und Techniken

Die nächsten Termine:

06.08.-10.08.2012 Seminarzentrum Wiesbaden
19.11.-23.11.2012 Seminarzentrum Wiesbaden

Agiles Projektmanagement mit SCRUM

Die nächsten Termine:

10.09.-11.09.2012 Seminarzentrum Wiesbaden
06.12.-07.12.2012 Seminarzentrum Wiesbaden

Weitere Informationen und Anmeldung unter:

<http://training.ordix.de/>

ORDIX AG



Während ich diese Zeilen schreibe, wird gerade die 2. Halbzeit im Spiel Deutschland gegen Dänemark angepfiffen, die Griechen haben gewählt (was genau weiß ich aber noch nicht) und wie Sie diesen Zeilen entnehmen können, bin ich nicht so sehr an Fußball interessiert. Selbst wenn Sie Fußballfan sind, nehmen Sie die nächsten Zeilen also nicht so ernst. Außerdem ist, wenn Sie dies lesen, die Sache mit der EURO 2012 gelaufen, die Sache mit dem EURO vielleicht auch.

Irritiert hat mich heute aber, dass die Russen bereits ausgeschieden sind. Noch dazu gegen Griechenland, obwohl sie (die Griechen) eigentlich rein nach Zahlen schlechter als die Russen da standen, mit einer Ausnahme: Sie haben die Russen geschlagen, als einzige sogar. Eine Regeländerung der UEFA soll also Schuld daran haben. Oder wird da vielleicht gar durch die EURO der EURO unterstützt?

Sollte man dann nicht auch noch kurzfristig überlegen, weitere Regeländerungen einzuführen, so dass beispielsweise Spanien, Italien und Griechenland das Endspiel gemeinsam bestreiten und alle drei gewinnen. Dann hilft vielleicht die EURO die EUROkrise zu beenden, weil in dem Glückstaumel plötzlich die Zinsen für Anleihen aus den drei Ländern auf 1,5% eingefroren werden. Moodys, Fitch und Standard & Poor's werden abgeschafft, weil sie in Dollar bilanzieren und Dollar werden einfach immer nur gedruckt und gedruckt und gedruckt ...²⁾

Ja die Iren sind schon nach Hause gefahren und für die Portugiesen brauchen wir auch noch etwas, etwa einen EURO pro Jahr oder so. Vielleicht hilft es für die Zukunft aber auch, dass England gar nicht mehr zur EURO kommt, weil sie ja auch keinen EURO haben. Russland darf nur noch bei der Qualifikation mitspielen, muss aber, wenn Sie sich qualifizieren, den EURO einführen oder aber es kann sich durch Gasausgleichszahlungen auch ohne den EURO in die EURO einkaufen.

Ach ist das schön, dass die EURO nicht mehr Europameisterschaft heißt. Sonst könnte ich Ihnen nur das Inhaltsverzeichnis bieten, wobei der Autor des Cloud Control Artikels (Oracle Cloud Control 12c, c wie Cloud, nicht mehr g wie Grid) hoffentlich nicht durch mein letztes Editorial inspiriert wurde. Auch DB2 hat Indizes und wir erklären Ihnen warum. Um schwarze Löcher, in die hoffentlich kein EURO fällt, geht es beim Artikel zur physikalischen Speicherverwaltung.

Bei der EUROkrise wird inzwischen vermutlich schon eher agiles Projektmanagement benötigt. Wer also zum nächsten Gipfel fährt, unser Artikel „Klassisches vs. agilem Projektmanagement“ könnte für denjenigen schon ganz wichtig sein. Auch wichtig: es heißt jetzt Oracle Solaris und nicht mehr Sun Solaris, wir schreiben über Version 11.

Natürlich führen wir die Erfolgsreihen zu Softwarearchitektur, Projektmanagement in der Praxis (vielleicht auch nicht schlecht für die neue griechische Regierung), Java 7 und Hibernate fort.

Inzwischen glaube ich fast, die EZB hat die UEFA bestochen (oh nein es heißt „gesponsored“), damit der EURO zumindest ein paar Wochen positive Schlagzeilen macht. Aber da waren ja die Krawalle nicht an der Börse sondern auf Polens Straßen.

Oh je die deutsche Nationalmannschaft wird am Freitag gegen Griechenland ausscheiden müssen oder die Bundeswehr muss alle deutschen Urlauber in Griechenland evakuieren. Die Dänen sind geschlagen und ich wünsche Ihnen, wenn Sie dies lesen, viel Spaß mit der EURO gehabt zu haben.

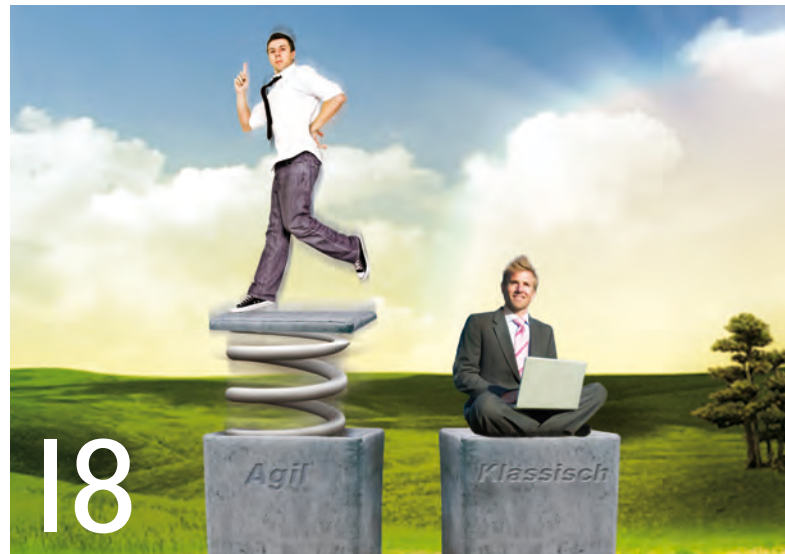
Ihr

Wolfgang Kögler

1) Ich bin mir unsicher, ob ich das verwenden darf oder die UEFA um Erlaubnis fragen muss.
2) Oettinger hat heute (oder vermutlich letzte Woche, weil es stand heute in der FAS) gesagt, dass Obama schöne Sonntagsreden hält, aber von Montag bis Freitag keine Macht hat. Da frag ich mich, was macht Obama eigentlich am Samstag? Vielleicht EURO gucken? Ja der Oettinger, der spricht so gutes Englisch, der würde es dem Obama auf dem G20 Gipfel schon erzählen, aber wir schicken halt nur die Merkel nach Mexiko.



Softwarearchitektur: Architekturbewertung



Klassisches vs. agiles Projektmanagement

Datenbanken

- 11..... **Oracle Cloud Control 12c (Teil I): Über den Wolken...**
Die Cloud ist in aller Munde. Oracle springt mit Oracle Cloud Control 12c auf dieses Thema auf. In dieser neuen Reihe geben wir einen Einblick in die Arbeitsweise des Werkzeugs.

- 24..... **IBM DB2: Indizes und Performance**
Die Vergabe von Indizes auf Datenbanktabellen dient dem schnellen Auffinden von Tabellendaten. Dies kann aber gleichzeitig auch zu einer hohen CPU-Auslastung führen. Wie man dem Problem aus dem Wege geht und was bei der Vergabe von Indizes zu beachten ist, erläutern wir in diesem Artikel.

- 30..... **Physikalische Speicherverwaltung in Oracle (Teil I): Tablespaces und schwarze Löcher**
Können Sie eine Aussage zum Füllgrad ihrer Oracle Datenbank geben? Im Zuge der neuen Artikelreihe erläutern wir, wie das physikalische Speichermanagement von Oracle funktioniert. Im ersten Teil wird das Augenmerk auf den Füllgrad einzelner Tablespaces gelegt.

- 39..... **Active Data Guard - Praxisbericht: Aufbau einer Lesefarm mit Active Data Guard**
Durch den Aufbau von Lesefarmen können Unternehmen eine hohe Leselast von ihren Primary-Datenbanken nehmen und dennoch auf aktuelle Daten zugreifen. Dieser Praxisbericht schildert den Aufbau einer solchen Farm mit Hilfe von Oracle Active Data Guard.

Projektmanagement

- 18..... **Traditionell oder evolutionär: Klassisches vs. agiles Projektmanagement**
Es gibt verschiedene Ansätze für die Abwicklung von Projekten. Aber welche wendet man auf sein Projekt an? In diesem Artikel vergleichen wir das klassische mit dem agilen Projektmanagement und stellen jeweils geeignete Einsatzgebiete vor.

- 27..... **Projektmanagement in der Praxis (Teil V): Eine Projektkultur schaffen - aber richtig!**
Um der wachsenden Bedeutung von Projekten gerecht zu werden, ist es sinnvoll, eine Projektkultur im Unternehmen zu etablieren. Dieser Artikel stellt die Ziele und die Maßnahmen für die Entwicklung einer Projektkultur vor.

Softwarearchitektur

- 6..... **Softwarearchitektur (Teil V): Architekturbewertung**
Softwarearchitekturen müssen wie jede andere Entwicklung bewertet werden. In unserem Artikel erläutern wir, mit welchen Maßstäben und Methoden eine solche Bewertung durchgeführt werden kann.



IBM DB2: Indizes und Performance



Physikalische Speicherverwaltung in Oracle

Java/JEE

**14..... Java 7 - Was bringt die neue Version? (Teil III):
NIO.2 - Filesystem & Path**
Auf welchen Pfaden wandelt die neue „New I/O Teil 2“?
Der Einzug der Klasse Path in die neue File System API vereinfacht das Navigieren erheblich.

**36..... Hibernate für Fortgeschrittene (Teil II):
SubSelect oder batchSize? Das ist hier die Frage**
Neben der Batch-Fetching-Strategie stellt Hibernate auch die SubSelect-Fetching-Strategie zur Verfügung. Die Unterschiede der beiden Strategien werden in diesem Artikel aufgezeigt.

Betriebssysteme

**33..... Oracle Solaris 11 - Neue Funktionen (Teil I):
Boot-Umgebungen**
Umfassende Änderungen in der neuen Solaris-Version machen den Switch fast zu einem Neueinstieg in das Betriebssystem. In dem ersten Artikel der Reihe nehmen wir die Boot-Umgebungen unter die Lupe, welche die Live-Upgrade-Funktion älterer Versionen ablösen.

Aktuell | Standards

21..... Larry Ratlos

22..... Seminarübersicht: Juni 2012 bis Dezember 2012

Impressum

Herausgeber: ORDIX AG Aktiengesellschaft für Softwareentwicklung, Beratung, Schulung und Systemintegration, Paderborn
Redaktion: Jens Pothmann, Evelyn Ernst
V.i.S.d.P.: Benedikt Georgi, Wolfgang Kögler
Anschrift der Redaktion: ORDIX AG | Westernmauer 12 - 16 | 33098 Paderborn
 Tel.: 05251 1063-0 | Fax: 0180 1673490
Gestaltung/Layout: Jens Pothmann
Auflage: 9.200 Exemplare
Druck: Druckerei Bösmann, Detmold
Bildnachweis: © fotolia.de | durchblick | drubig-photo #14186901
 © sxc.hu | Modern architecture | Chemtec
 © freedesign4.me | Free Vector Blueprint
 © flickr.com | Lego!!! | Glenn Calvin
 © sxc.hu | Levitation 1 | bizior
 © sxc.hu | spring | dlunny
 © istockphoto.com | Businessmann Sits in... | PeskyMonkey
 © istockphoto.com | bolide on high-speed road | ssuaphoto
 © vectoropenstock | Abstract Color Background | webdesignhot
 © freepik | Gear chain | YouToArt.com
 © istockphoto.com | Country highway through the ... | sankai
 © sxc.hu | directions arrows | hisks
 © sxc.hu | Traffic Sign | nurcen
Autoren: Lars Brenker, Lars Eisenblatt, Christian Fertsch, Andreas Flügge, Benedikt Georgi, Martin Hoermann, Carsten Hummel, Wolfgang Kögler, Andreas Maßmann, Thomas Schwarz, Michael Skowasch
Copyright: Die ORDIX news erscheint vierteljährlich. Alle Eigentums- und Nachdruckrechte, auch die der Übersetzung, der Vervielfältigung der Artikel oder von Teilen daraus, sind nur mit schriftlicher Zustimmung der ORDIX AG gestattet. ORDIX® ist ein eingetragenes Markenzeichen der ORDIX AG.
Warenzeichen: Einige der aufgeführten Bezeichnungen sind eingetragene Warenzeichen ihrer jeweiligen Inhaber. ORDIX® ist eine registrierte Marke der ORDIX AG.
Haftung: Eine Haftung für die Richtigkeit der Veröffentlichungen kann trotz sorgfältiger Prüfung durch die Redaktion vom Herausgeber nicht übernommen werden.
 Sie können die Zusendung der ORDIX news jederzeit ohne Angabe von Gründen schriftlich (z.B. Brief, Fax, E-Mail) abbestellen.



Softwarearchitektur (Teil V)

Architekturbewertung

„Was man nicht messen kann, kann man nicht kontrollieren!“ lautet eine alte Weisheit, die mittlerweile auch in der Softwareentwicklung Berücksichtigung findet. Auch Softwarearchitekturen müssen entwickelt, kontrolliert und bewertet werden. Aber wie soll das funktionieren? Welche Maßstäbe existieren dafür und wie sollte der Softwarearchitekt vorgehen? Der folgende Artikel gibt einen Überblick, wie Softwarearchitekturen bewertet werden können, welche Methoden dem Architekten dafür zur Verfügung stehen und wie er die Ergebnisse zur Erstellung hochwertiger Softwaresysteme nutzen kann.

Motivation

Gewöhnlich entsteht die Architektur eines Systems schon in den frühen Entwicklungsphasen. Der Architekt muss sich also von Beginn an die Frage stellen, ob die gewählten architektonischen Konzepte zum Bau eines „guten“ Softwaresystems geeignet sind.

Diese Fragestellung begleitet ihn während des gesamten Lebenszyklus eines Systems. Da sich die Anforderungen immer wieder verändern können, ist die Bewertung der Architektur in jeder Iteration dringend angeraten.

Wird die Architektur nicht frühzeitig regelmäßig bewertet und angepasst, besteht die Gefahr, dass bestimmte Ziele und Anforderungen aus den Augen verloren werden. Zu spät bemerkt, ergeben sich daraus meist erhöhte Aufwände und Kosten, denn Korrekturen an der Architektur sind meist sehr teure Maßnahmen.

Gegenstand der Bewertung

Eine Softwarearchitektur direkt zu bewerten ist ein schwieriges Unterfangen. Einfacher ist es Aspekte zu betrachten, die entweder die Architektur beeinflussen, ein Teil von ihr sind oder sich als Auswirkung der Archi-

tektur präsentieren. Im Wesentlichen handelt es sich dabei um die Anforderungen an das zu entwickelnde System, den Quellcode, die Dokumentation und um sonstige Artefakte, wie Testberichte und Analysen zum Verhalten (z.B. Performance- oder Durchsatzanalysen).

Ergebnisse

Der Aspekt der Qualitätssicherung steht für den Architekten im Vordergrund der Architekturbewertung. Er erwartet eine Antwort auf die Frage, ob die Architektur eines Systems für die geforderten Merkmale geeignet und gut ist. Diese allgemein gefasste Frage lässt sich aber meistens nur durch die Klärung einer Reihe von spezielleren Teilaspekten beantworten.

Dazu gehört beispielsweise die Validierung der funktionalen und nicht-funktionalen Anforderungen auf Vollständigkeit und Eindeutigkeit. Werden hier Lücken aufgedeckt, kann sich eine zunächst als geeignet bewertete Architektur plötzlich als völlig ungeeignet herausstellen. Ähnliches gilt für die Identifizierung von Risiken und die Überprüfung der bisher bereits vorgenommenen Architekturentscheidungen.

Weiterhin ist auch die genaue Intention der Architekturbewertung zu beachten. Neben der oben formulierten Frage nach der Eignung der Architektur, könnte eine Variante die Bewertung eine konkrete Implementierung auf die Einhaltung von Architekturvorgaben sein. Das Spektrum der Ergebnisse ist breit und kann von einfachen Ja/Nein-Aussagen, über feinere Abstufungen bis hin zu konkreten Maßnahmenkatalogen variieren.

Qualität und Quantität

Grundsätzlich wird zwischen der quantitativen und qualitativen Bewertung einer Architektur unterschieden. Während quantitative Methoden in der Regel Kennzahlen für bestimmte Messgrößen liefern, entziehen sich die qualitativen Methoden einer solchen Bewertung und liefern andere Ergebnisse, mit denen sie die Güte und Beschaffenheit einer Architektur beschreiben.

Quantitative Bewertung und Metriken

Unter einer Metrik versteht man eine Kennzahl, die für einen bestimmten Sachverhalt repräsentativ ist. Dahinter verbirgt sich eine Vorschrift zur quantitativen, reproduzierbaren Messung einer Größe oder eines Zustandes. In der Softwareentwicklung werden diese

Metriken meist im Rahmen einer statischen Quellcode-Analyse angewandt.

Zu den wichtigsten Größen zählen:

- Abhängigkeitsmaße (Kopplung)
- Komplexitätsmaße (mögliche Ablaufpfade, zyklomatische Komplexität)
- Strukturmaße (Vererbungstiefe)
- Größenangaben (Anzahl Code-Zeilen pro Klasse/Methode, Anzahl statischer Methoden, Anzahl Kommentare je Code-Zeile).

Ein großer Vorteil von Metriken liegt in der guten Unterstützung durch die Werkzeuge und damit in der

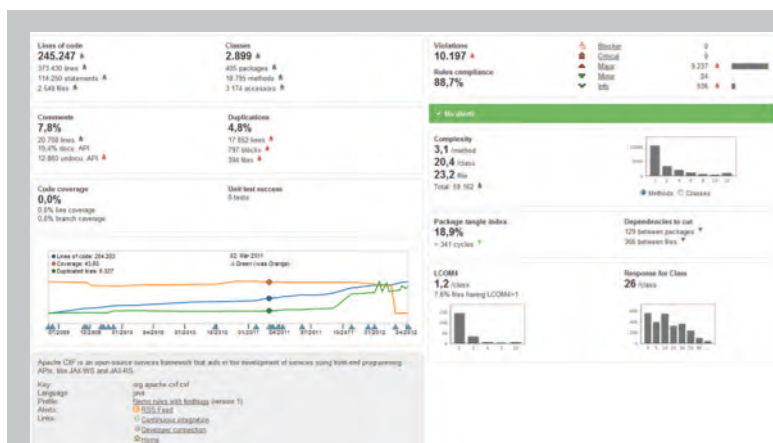


Abb. 1: Metriken (hier: Sonar für das Apache CXF Projekt).

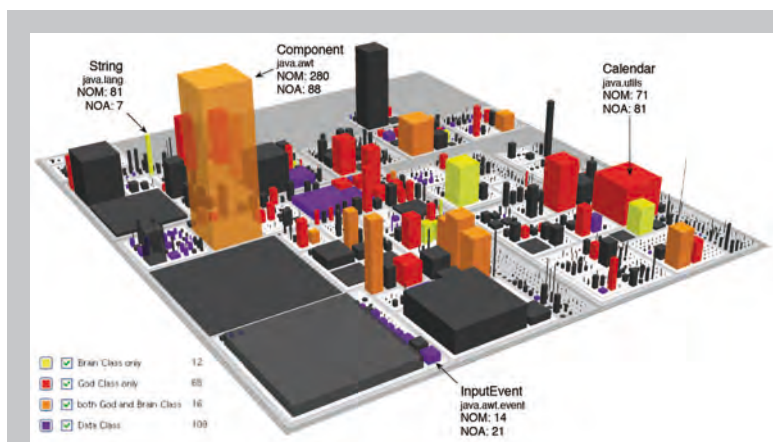


Abb. 2: Visualisierung von Code-Strukturen in Codecity. [5]

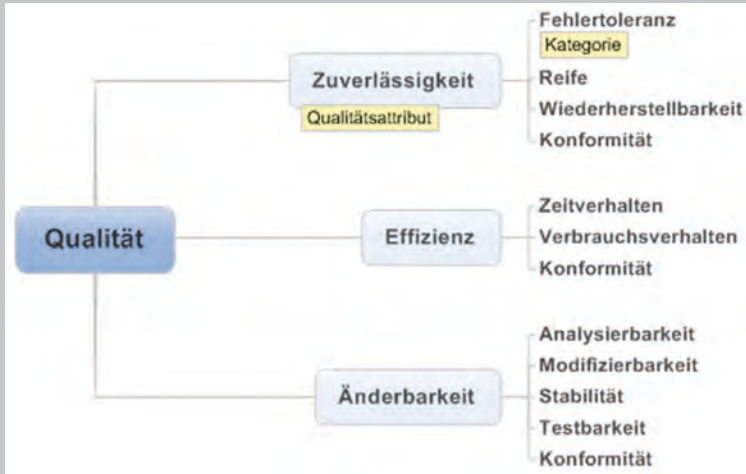


Abb. 3: Die ersten beiden Ebenen eines Utility Tree (nach ISO 9126).

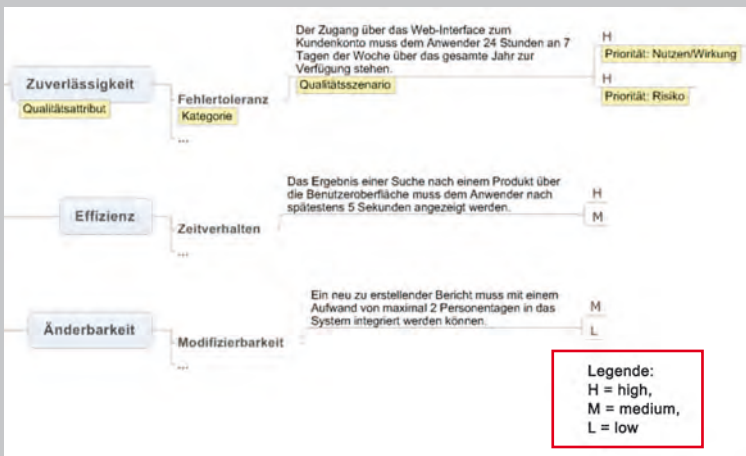


Abb. 4: Priorisierte Szenarien.

Automatisierbarkeit. Sie lassen sich sehr leicht in ein Continuous-Integration-Szenario einbinden. So stehen dem Architekten und den Entwicklern nach jedem Build die aktualisierten Ergebnisse zur Verfügung. Durch die zeitliche Bildung einer Historie lassen sich so Degenerationsprozesse des Quellcodes im Laufe der Entwicklung messen und Auswirkungen auf die Architektur einschätzen (siehe Abbildung 1).

Ein weiterer Vorteil ist die Möglichkeit der Visualisierung. Neben der Abbildung in Charts, Abhängigkeitsgraphen und anderen Formen, existieren auch eine Reihe innovativerer Darstellungen wie z.B. Codecity (siehe Abbildung 2), die dem Betrachter auf den ersten

Blick Hinweise auf potentielle strukturelle Schwächen der Architektur liefern.

Klassen und Pakete werden hier als Plateaus und Gebäude dargestellt. Die Farbe, Form, Höhe und Grundfläche der Gebäude sind mit bestimmten Metriken verbunden, so dass sich z.B. aus der Kombination aus Höhe und Farbe bestimmte Eigenschaften der Klassen und Hinweise auf Architektureigenschaften ableiten lassen.

Gesunder Menschenverstand

Metriken stellen aber kein Allheilmittel dar. Der Hinweis „Wer viel misst, misst viel Mist!“ hat auch hier seine Gültigkeit. Metriken benötigen zum einen immer eine Norm, die jemand definiert. Eine Messgröße ist erst dann interpretierbar, wenn sie in einen Kontext eingebettet wird. So kann in einem Projekt ein Testüberdeckungsmaß für den Quellcode von 50 % oder von 90 % gefordert werden, beides kann je nach Situation und Projekt vollkommen korrekt sein.

„Traue keiner Statistik, die du nicht selbst gefälscht hast!“ ist ein gerne zitierter Satz, der aber auch im Zusammenhang mit Metriken einen bedenkenswerten Funken Wahrheit enthält. So hat beispielsweise ein Testüberdeckungsmaß von 80 % natürlich eine ganz andere Aussagekraft, wenn überwiegend Fachlogik getestet wurde, als beispielsweise Getter- und Setter-Methoden. Wird die Entwicklung zu sehr durch Metriken getrieben, kann das kontraproduktiv sein, da Entwickler dabei oft eine erstaunliche Kreativität an den Tag legen.

Weiterhin neigen Metriken dazu, eine scheinbare Sicherheit vorzugaukeln. Der Architekt und die Entwickler müssen sich immer im Klaren sein, dass sie zwar Hinweise auf den Zustand und die Veränderung der Software geben können, aber keinerlei Aussage über die Funktionsfähigkeit und Qualität zur Laufzeit zulassen.

Wohl überlegt angewendet, sind Metriken aber ein nützliches Hilfsmittel, um Hinweise auf Architektureigenschaften zu ermitteln. Eine kritische Betrachtung der verwendeten Metriken sollte aber selbstverständlich sein. Als alleiniger Indikator zur Bewertung einer Architektur sind sie völlig unzureichend. Der gesunde Menschenverstand ist schließlich durch nichts zu ersetzen.

Qualitative Methoden

Im Gegensatz zu den quantitativen Verfahren fehlt bei den qualitativen Methoden die offensichtliche Messgröße. Für die Bewertung wird aber eine Art Referenzwert benötigt, gegen den ein Maß angelegt werden

kann. Dazu bedienen sich die qualitativen Methoden den nicht-funktionalen Anforderungen. Sie überprüfen, in welchem Maß die Architektur diesen Anforderungen genügt.

Szenariobasiert

Eine Möglichkeit solche „Referenzwerte“ zu definieren, ist die Beschreibung sogenannter Szenarien. Wie bereits in einem früheren Artikel zur Architekturentwicklung beschrieben [4], sollte es ein gemeinsames Verständnis aller Stakeholder bezüglich der (fachlichen) Funktionalität des Systems und deren Priorisierung geben.

Der Architekt versucht aus den fünf bis zehn wichtigsten fachlichen Anwendungsfällen anschließend die Architekturtreiber zu ermitteln. Dazu erstellt er einen speziellen Qualitätsbaum (Utility Tree), in dem er die entsprechenden nicht-funktionalen Anforderungen einträgt und in Unterkategorien aufsplittet (siehe Abbildung 3).

Anschließend werden die Architekturtreiber der konkreten Szenarien, die für die Qualitäten der nicht-funktionalen Anforderungen verantwortlich sind, formuliert. Diese müssen präzise und möglichst handlungsorientiert, für alle Projektbeteiligten verständlich und einvernehmlich, beschrieben werden.

Zusätzlich findet noch eine gemeinsame Priorisierung der Szenarien nach Nutzen bzw. Wirkung (geschäftlicher Wert) und nach Risiko (technische Schwierigkeit) statt. Als Skala reicht hier eine grobe, dreistufige Einteilung (High, Medium, Low). Eine solche Priorisierung ist beispielhaft in Abbildung 4 dargestellt. Mit diesen priorisierten Szenarien steht nun der Bewertungsmaßstab für die Architektur fest.

Bewertung

Die eigentliche Bewertung besteht in der Betrachtung der einzelnen Szenarien. Begonnen wird mit den am höchsten priorisierten (H, H). Jedes Szenario wird daraufhin untersucht, ob die aktuelle Architektur den geforderten Qualitäten des Szenarios entspricht. Durch die Analyse folgender Aspekte lassen sich Aussagen zur Güte der Architektur treffen und gegebenenfalls Maßnahmen zur Anpassung der Architektur ableiten:

- Welche Architekturkonzepte liegen der Umsetzung des Szenarios zugrunde?
- Welche Begründung gibt es für die Wahl der zugrundeliegenden Architekturkonzepte?
- Welchen Einfluss hat die Wahl der Architekturkonzepte auf andere Qualitätsmerkmale?

Glossar

ATAM / SAAM

ATAM (architecture tradeoff analysis method) und SAAM (software architecture analysis method) sind zwei Architekturbewertungsmethoden, die vom Software Engineering Institute (SEI) der Carnegie Mellon University entwickelt wurden. Es handelt sich um szenariobasierte Methoden, die sich besonders mit der Analyse nicht-funktionaler Anforderungen befassen.

Continuous-Integration-Szenario

Beschreibt einen automatisierten Prozess des ständigen Übersetzens und Testens einer sich in der Entwicklung befindlichen Anwendung. Die beteiligten Entwickler übertragen frühzeitig und regelmäßig ihre Änderungen in eine Versionsverwaltung, aus der automatisch das Gesamtsystem neu gebaut und getestet wird. So ergibt sich ständig ein Feedback zum Zustand des Systems und potentiellen Integrationsproblemen.

Stakeholder

Englischer Begriff für Interessensvertreter. Im Bereich Projektmanagement sind dies alle Personen oder Parteien, die ein berechtigtes Interesse bzw. Anspruch am Verlauf und / oder dem Ergebnis eines Projektes haben, z.B. Auftraggeber, Projektmitarbeiter, zukünftige Nutzer, etc.

zyklomatische Komplexität

Eine Softwaremetrik, mit der die Komplexität eines Softwaremoduls gemessen werden kann.

- Welche Kompromisse oder Risiken resultieren aus der Wahl der Architekturkonzepte?

Die Beurteilung der Ergebnisse unterliegt leider keinen allgemeingültigen Regeln, hier ist die subjektive Einschätzung des Architekten und damit auch eine gehörige Portion Erfahrung gefragt.

Die Architekturbewertung wird in der Regel in enger Zusammenarbeit mit den wichtigsten Stakeholdern durchgeführt. Oft sind es externe Architekten, die zusammen mit den Architekten und Entwicklern deren Systeme begutachten und das o.g. Verfahren durchführen.

Dadurch werden allen Beteiligten noch einmal die Ziele und Prioritäten verdeutlicht und die wichtigsten Architekturkonzepte und -entscheidungen herausgearbeitet. Dies ist ein nicht zu unterschätzender positiver Nebeneffekt jeder Architekturbewertung.

SAAM und ATAM

Der szenariobasierte Ansatz stammt vom Software Engineering Institute (SEI) der Carnegie Mellon University und liegt den beiden Methoden SAAM (Software Architecture Analysis Method) und ATAM (Architecture Tradeoff Analysis Method) zugrunde.

Beide Methoden sind strenger formalisiert und stehen im Ruf etwas zu akademisch zu sein. Der generierte

Links

- ▶ [1] ORDIX news Artikel 01/2011
„Softwarearchitektur (Teil I) - Die Grundsteinlegung“:
http://www.ordix.de/ORDIXNews/1_2011/softwarearchitektur.html
- ▶ [2] ORDIX news Artikel 02/2011
„Softwarearchitektur (Teil II) - Der Softwarearchitekt“:
http://www.ordix.de/ORDIXNews/2_2011/softwarearchitektur.html
- ▶ [3] ORDIX news Artikel 04/2011
„Softwarearchitektur (Teil III) - Die Architekturentwicklung“:
http://www.ordix.de/ORDIXNews/3_2011/architekturentwicklung.html
- ▶ [4] ORDIX news Artikel 01/2012
„Softwarearchitektur (Teil IV) - Architektursichten“:
http://www.ordix.de/ORDIXNews/1_2012/softwarearchitektur_t4_architektursichten.html
- ▶ [5] Downloadseite Richard Wetzel "Codecity":
<http://www.inf.usi.ch/phd/wetzel/codecity.html>

Bild: © fotolia.de | durchblick | drubig-photo #14186901
© sxc.hu | Modern architecture | Chemtec
© freedesign4.me | Free Vector Blueprint

Aufwand für die Durchführung ist höher, als bei anderen Methoden. Der zentrale Aspekt bei ATAM (als Weiterentwicklung aus SAAM) ist das Herausstellen von Tradeoff und Sensitivity Points. Damit sollen möglichst früh Risiken identifiziert werden, die aus in Konkurrenz stehenden Qualitätsattributen resultieren.

Erfahrungsbasierte Ansätze

Ebenfalls zu den qualitativen Methoden zählen die sogenannten erfahrungsbasierten Ansätze, die ganz

auf den Erfahrungsschatz des Architekten setzen. Meistens führt er eine SWOT-Analyse (Strength, Weakness, Opportunity, Threat) durch, in der die Stärken, Schwächen, Möglichkeiten und Risiken der Architektur aus seiner subjektiven Sicht heraus dokumentiert und gegebenenfalls entsprechende Maßnahmen abgeleitet werden. Diese Ansätze sind in der Regel weniger formell und lassen sich auch mit geringerem Aufwand durchführen. Sie bedürfen aber sehr erfahrener Architekten.

Fazit

Für die Bewertung von Softwarearchitekturen gibt es eine Reihe von Methoden. Keine von ihnen bietet einen universell anwendbaren Algorithmus, der ein aussagekräftiges Ergebnis liefern kann. Die Kompetenz und Erfahrung eines Softwarearchitekten ist und bleibt weiterhin notwendig.

Vielleicht sogar wichtiger als die Frage nach der Methode, ist die Erkenntnis, dass eine Architekturbewertung zur Qualitätssicherung unabdingbar ist und nur dann ihren Zweck erfüllt, wenn sie regelmäßig über den gesamten Lebenszyklus eines Softwaresystems durchgeführt wird.



Andreas Flügge
(info@ordix.de)

Seminarempfehlung: Softwarearchitekturen

- ▶ **Informationen/Online-Anmeldung:** <http://training.ordix.de/siteengine/action/load/kategorie/Java-JEE/nr/1116/index.html>

Die Qualität von Softwarearchitekturen ist ein entscheidender Erfolgsfaktor in IT-Projekten und Softwarearchitekten nehmen die zentrale Rolle bei deren Entwicklung ein. Das Seminar behandelt aktuelle Methoden, Konzepte und Technologien, die dem praktizierenden Architekten und Entwickler helfen systematisch Softwarearchitekturen zu entwerfen. Das Seminar gibt einen Überblick über die Aufgaben und Fähigkeiten des Softwarearchitekten. Es zeigt dem Teilnehmer, mit welchen Mitteln Softwarearchitekturen umgesetzt werden und demonstriert dies anhand praxisnaher Beispiele, mit konkreten Technologien. Weiter wird beleuchtet, wie Softwarearchitekturen dokumentiert und kommuniziert werden und wie ihre Qualität bewertet und sichergestellt werden kann.

Seminarinhalte

- Grundlagen Softwarearchitektur
- Architekturentwicklung
- Softwarearchitekt
- Dokumentation und Kommunikation
- Architekturprinzipien
- Grundlegende architektonische Konzepte
- Architekturmittel | Technologien
- Architekturbewertung
- Ausblick

Termine

- 25.06. - 29.06.2012 in Wiesbaden
- 24.09. - 28.09.2012 in Wiesbaden

Seminar-ID: E-SWA-01

Dauer: 5 Tage

Preis pro Teilnehmer: 1.890,00 € (zzgl. MwSt.)

Frühbucherpreis: 1.701,00 € (zzgl. MwSt.)

Neue Reihe: Enterprise Manager Oracle Cloud Control 12c (Teil I)

Über den Wolken ...

Am 3. Oktober 2011 brachte Oracle sein neues Monitoring- Werkzeug „Oracle Enterprise Manager Cloud Control 12c“ heraus. Diese Version, die als eine umfassende Weiterentwicklung von Grid Control 11g gilt, wird in dieser neuen Reihe der ORDIX news unter die Lupe genommen.

Überblick der neuen Funktionen

Mit dem Oracle Enterprise Manager Cloud Control 12c werden einige neue bzw. modifizierte Funktionen angeboten:

- eine komplett neue Oberfläche
- eine vereinfachte Vorgehensweise bei Installation und Upgrade
- ein neues Event Management und Incident Management
- ein erweitertes Database Performance Management
- ein Lifecycle Management
- die User-Defined-Metriken werden zu Metric Extensions
- ein überarbeitetes Rechtekonzept
- Cloud Computing
- ein Multi Channel Access
- eine Erweiterbarkeit des Systems über Plugins
- ein Self Update
- ein Testdaten-Management
- ein Exadata Management

In diesem Artikel geben wir einen Überblick über die wichtigsten Neuerungen.

Die neue Oberfläche

Die Optik des neuen Cloud Control 12c unterscheidet sich fundamental von dem alten Grid Control 11g. Alles wirkt auf den ersten Blick „bunter“ und ist optisch stark überarbeitet worden. Der Benutzer benötigt weniger Klicks, um zu seinem gewünschten Ziel zu gelangen.

Zusätzlich bietet die neue Oberfläche die Möglichkeit, sich eine eigene Sicht zu erstellen und seinen Fokus auf die gewünschten Objekte zu legen. So kann der Anwender z.B. seine Sicht eher auf die Administration oder das Monitoring ausrichten. Die neue Oberfläche ist in der Abbildung 1 zu sehen.

Installation und Upgrade

Die Installation und das Upgrade von Cloud Control 12c wurden im Vergleich zu seinen Vorgängern stark vereinfacht. Sämtliche Komponenten, wie der Middleware Server (WebLogic) und das JDK, werden von dem Cloud Control Installer installiert. Nur die als Repository vorgesehene Datenbank muss vorab installiert und konfiguriert sein. Zudem besteht die Möglichkeit, von Grid Control in Version 11.1 oder mit Version 10.2.0.5 auf das neue Cloud Control 12c zu migrieren. Ein Upgrade dieser Art beinhaltet automatisch ein Upgrade sämtlicher Agenten, da die Agenten der Version 11 nicht mehr mit der neuen Version 12c kommunizieren können. Daher müssen sämtliche Agenten ebenfalls ein Upgrade durchlaufen, was zu einer gewissen Downtime führt. Mehr hierzu in dem nächsten Artikel dieser Reihe.

Administration der Cloud

Die Administration der Cloud wird durch neue Oberflächen und Funktionen erleichtert. Die Übersicht über eine Datenbank ist um einige Ansichten erweitert worden. Zusätzlich zu der Standardansicht, die dem Administrator geboten wird, kann er die Ansicht personalisieren und von ihm gewünschte Werte hinzufügen.

Sicherheit in der Cloud

Cloud Control bietet für eine sichere Verbindung nach außen sowohl die Möglichkeit einer Authentifizierung über die Enterprise-Manager-Konsole als auch eine weitere Möglichkeit über das Werkzeug `emc11`. Dafür werden sechs verschiedene Authentifizierungsarten bereitgestellt:

- Repository-Authentifizierung
- Oracle Access Manager (OAM)
- SSO-Authentifizierung
- Enterprise-User-Security-Based-Authentifizierung
- Oracle Internet Directory (OID)-Authentifizierung
- Microsoft-Active-Directory-Authentifizierung

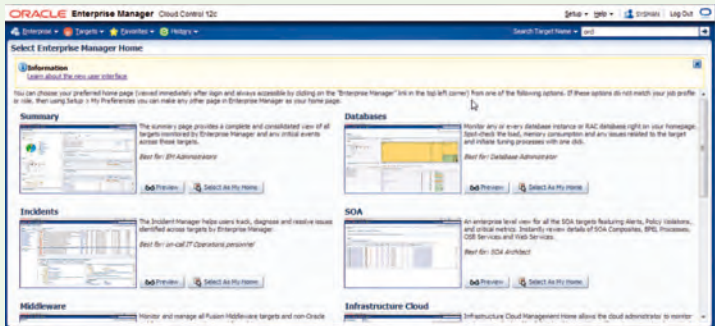


Abb. 1: Neue Oberfläche des OEM Cloud Control.

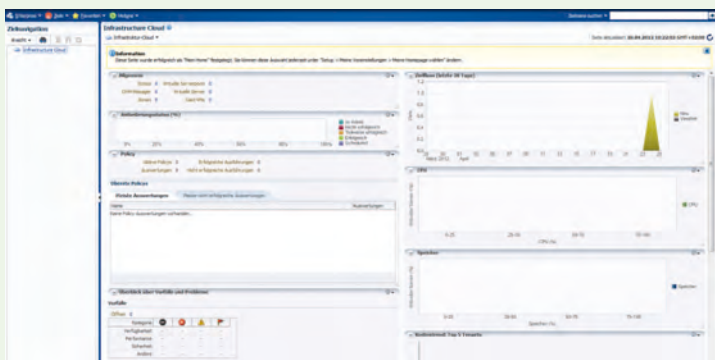


Abb. 2: Cloud Computing.

Named Credentials

Für sämtliche Schnittstellen können nun zentrale **Named Credentials** abgespeichert und global nutzbar gemacht werden. Dadurch werden Zugriffe auf viele Ziele ermöglicht, ohne dem Benutzer Passwörter bekannt zu geben. Diese werden vom Administrator in Cloud Control hinterlegt. Darüber hinaus werden für die Verwaltung der vorhandenen Umgebung über 100 verschiedene Rollen mitgeliefert. Dabei handelt es sich um Rollen, vom Superadministrator bis hin zum Administrator für spezifische Aufgaben.

Self Update

Für den Datenbankadministrator hat sich in Bezug auf die Update-Politik einiges geändert. In der neuen Cloud Control Version 12c werden keine Updates im herkömmlichen Sinne mehr installiert. Cloud Control

verfügt über eine Option, die „Self Update“ genannt wird. Es werden nur noch Plugins und dadurch neue Versionen, zum Beispiel der Datenbanksoftware / des Application Server usw., eingespielt. Damit können Updates effektiver aufgespielt werden und die Administratoren haben einen guten Überblick über die neuen Versionen. Allerdings wird nicht immer, wie im alten Grid Control, eine neue Version installiert, sondern lediglich das von Oracle bereitgestellte Plugin. Die Plugins können sowohl online als auch offline installiert werden.

Incident Management

Das Incident Management hat das Monitoring von Ereignissen stark vereinfacht. Sämtliche auftretenden Ereignisse auf den eingebunden Zielsystemen werden nur noch als Incidents abgebildet und verwaltet. Der Überblick über aufgetretene Fehler wird durch eine zentrale Konsole vereinfacht. Hierdurch ist ein Zugriff auf den „My Oracle Support“ gegeben, der eine direkte Fehlersuche ermöglicht. Es können auch benutzerdefinierte Sichten auf Ereignisse erstellt werden. Zusätzlich ist die Anbindung an ein externes Trouble-Ticket-System möglich. Oracle bietet außerdem ab sofort Schnittstellen für ein Monitoring der auf Apple-Geräten erzeugten Ereignisse an.

Database Performance Management

Im Rahmen des Database Performance Managements wurde das Real-Time-ADDM bereitgestellt. Er hilft dabei Datenbanken zu analysieren, die nur eine geringe oder keine Performance aufweisen. Hierbei können Zeiträume miteinander verglichen und ausgewertet werden. Die Unterschiede zwischen den Zeiträumen werden angezeigt und explizit hervorgehoben.

Lifecycle Management

Auch das Provisioning (Bereitstellen) von Software ist nun realisierbar. Von der Softwareverteilung (Deployment) bis hin zum Clonen von Oracle-Home-Verzeichnissen für Out-of-Place Upgrades bietet die neue Version viele Möglichkeiten. Es können nun Konfigurationseinstellungen wie zum Beispiel eine `listener.ora` überwacht und bei Änderungen eine Nachricht an die zuständigen Datenbankadministratoren geschickt werden.

Metric Extension

Die User-Defined-Metriken (UDM) in ihrer ursprünglich bekannten Form sind nicht mehr vorhanden. Sie werden von den Metric Extensions abgelöst. Es

existiert ein einfacher Workflow zum Testen, Veröffentlichlichen und Anwenden der Metric Extensions. Der Administrator hat im Vorfeld eine bessere Kontrolle über die Extensions, da durch ihr Testen eine stärkere Prüfung des Workflow gewährleistet wird. Sämtliche alten User-Defined-Metriken werden zwar weiterhin unterstützt, jedoch empfiehlt Oracle eine Migration der UDM, welche mit dem Werkzeug `emcli` ermöglicht wird.

Cloud Computing

Ein weiterer neuer und wichtiger Bestandteil des Cloud Control 12c ist das „Cloud Computing“. Es ermöglicht dem Administrator drei verschiedene Dienste als Service anzubieten:

- IaaS - Infrastructure as a Service
- DaaS - Database as a Service
- MWaaS - Middleware as a Service

Mit Hilfe dieser drei Services werden Oracle Virtual Machines, Datenbanken und Middleware per Knopfdruck bereitgestellt (siehe Abbildung 2). In Verbindung mit dem Erstellen einer Cloud können die dazugehörigen Ressourcen abgerechnet werden. Dazu steht ein „Chargeback“ zur Verfügung, bei welchem die Kosten für die einzelnen Komponenten definiert werden können. Zudem bietet Cloud Control einen Consolidation Planner, der dem Datenbankadministrator hilft eine Konsolidierung von Servern besser zu planen und die vorhandenen Ressourcen besser abstimmen zu können.

Exadata Management

Auch Exadata-Ziele können mit der neuen Version direkt überwacht werden, ohne hierfür extra Plugins nachinstallieren zu müssen. Jedoch muss das Datenbank-Diagnostic-Pack dafür lizenziert sein.

Testdaten-Management

Cloud Control ermöglicht es dem Datenbankadministrator, seinen Anwendern besser anonymisierte Testdaten zur Verfügung zu stellen. Im Rahmen des integrierten „Quality Management“ können anhand eines vorhandenen Datenmodells Daten über Data-Masking-Definitionen verwendet werden.

Neue Packs

Im Zuge der Veröffentlichung wurden einige zusätzlich zu lizensierende neue Packs bereitgestellt:

- Lifecycle Management Pack
- Cloud Management Pack
- Test Management Pack

Glossar

ASH

Automated Session History - Dieses Werkzeug dient dazu, Performance-Analysen auf Session-Ebene durchzuführen.

Cloud

Das beim Cloud Computing benutzte Rechnernetzwerk.

JDK

Java Development Kit - Umfasst die Laufzeitumgebung (JRE) und bietet zusätzlich Werkzeuge zum Entwickeln.

Provisioning

Ausstattung des Anwenders eines IT-Systems mit grundsätzlichen Voraussetzungen.

Real-Time-ADDM

Dieses Werkzeug hilft dabei, die Datenbanken im hängenden Zustand oder bei geringerer Performance zu analysieren.

Workflow

Vordefinierte Abfolge von Aktivitäten in einer Organisation.

Links

- ▶ [1] Oracle DBA Community:
http://www.oracle.com/webfolder/technetwork/de/community/dbadmin/tips/cc12_overview/index.html#ficy
- ▶ [2] Oracle Enterprise Manager Cloud Control Introduction :
http://docs.oracle.com/cd/E24628_01/doc.121/e25353/whats_new.htm#CEGIFFGA

Fazit

Neben den vielen grafischen Überarbeitungen gibt es auch einige funktionale Neuerungen, die wir Ihnen in den kommenden ORDIX news im Detail vorstellen werden. Der nächste Artikel dieser Reihe behandelt die Installation und das Upgrade auf Oracle 12c Cloud Control.



*Carsten Hummel
(info@ordix.de).*

Java 7 – Was bringt die neue Version? (Teil III)

NIO.2 – Filesystem & Path

Während sich der erste Artikel dieser Reihe [1] auf die neuen Funktionen im Project Coin konzentrierte und der zweite Teil [2] die Neuigkeiten im Concurrency-Bereich erläutert hat, richtet sich das Augenmerk in diesem Artikel auf die lang ersehnten Neuerungen in der „New I/O Teil 2“ (NIO.2). Dabei wird insbesondere die neue File System API genauer betrachtet, die u.a. das Navigieren im Dateisystem vereinfacht.

I/O, NIO und NIO.2

Sämtliche Ein- und Ausgabeoperationen von Java wurden seit Beginn unter dem Namen I/O geführt. Zu finden sind die Funktionalitäten im Package `java.io`. Mit Java 1.4 kamen dann neue Funktionalitäten hinzu, die seitdem unter dem Namen „New I/O“ im Package `java.nio` abgelegt sind. NIO.2 ist neben dem Project Coin die zweitgrößte Änderung in Java 7. Anders als bei der Einführung von NIO, gibt es für NIO.2 kein neues Package. Alle Neuerungen liegen ebenfalls im Package `java.nio`.

Die Änderungen im Überblick

Folgende essentiellen Änderungen haben sich in NIO.2 ergeben und werden in diesem Artikel genauer betrachtet:

- `FileSystem` repräsentiert ein Dateisystem
- `Path` repräsentiert einen relativen bzw. absoluten Pfad und löst die alte Klasse `java.io.File` ab
- `Files` dient als Utility-Klasse für Dateimanipulationen
- `FileAttributes` ermöglicht das Auslesen von Dateiattributen zum jeweiligen Dateisystem
- `DirectoryStream` erleichtert das Auslesen von Dateien und Verzeichnissen in einem Pfad
- `Files.walkFileTree(...)` dient zum rekursiven „Wandern“ durch den Verzeichnisbaum
- `WatchService` dient zum Überwachen von Verzeichnissen

Viele Leser werden an dieser Stelle große Augen bekommen und bereits denken: „Endlich, auf so etwas habe ich lange gewartet“. Schauen wir uns im Folgenden an, ob dies tatsächlich der Fall ist. Begeben wir uns auf eine Reise durch NIO.2.

Auf ins Dateisystem!

Eine neue abstrakte Klasse ist der Startpunkt für die Reise durch NIO.2. `java.nio.file.FileSystem` ist die Repräsentation eines Dateisystems. Über die Utility-Klasse `FileSystems` erhält der Entwickler Zugriff auf das Default Filesystem:

```
FileSystem fs = FileSystems.getDefault();
```

Mit diesem Zugriff auf das Dateisystem im Gepäck steht der Reise nichts mehr im Wege. Der nächste Halt ist die Klasse `Path`.

java.io.File adé – Die Klasse Path erhält Einzug

Die neue Klasse `Path` im Package `java.nio.file` soll die alte `File`-Klasse ablösen. Während ein `File` eine Datei oder ein Verzeichnis darstellt, repräsentiert die neue Klasse `Path` einen Pfad im Dateisystem. Dieser Pfad kann je nach Wunsch durch einen String, eine URI oder ein `File` erzeugt werden:

```
Path pfad = fs.getPath("c:", "ordixnews", "java7");
```

Da der Pfad hierarchisch aufgebaut ist, besteht die Möglichkeit u.a. den Root oder den Elternpfad zu ermitteln. Die Abbildung 1 erzeugt folgende Ausgabe:

```
Absolut: c:\ordixnews\java7
Parent: c:\ordixnews
Root: c:\
```

Vom Pfad zur Datei

Während `Path` lediglich einen Pfad zu einem Verzeichnis oder einer Datei repräsentiert und dort Dateimanipulationen vergeblich gesucht werden, hilft die Utility-Klasse `Files` an dieser Stelle weiter. Das Erzeugen, Schreiben, Lesen und Kopieren von Dateien ist somit ein Kinderspiel. Ebenfalls neu in Java 7 sind die `Open-Options`, welche eine Konfiguration des Zugriffs ermöglichen. Die `Open-Options` sind optional.

Erste Schreibversuche

Die Abbildung 2 zeigt, wie einfach es ist eine Datei zu erzeugen. Das Beschreiben der Datei geht mittels `Files.write(...)` ebenso schnell und einfach von der Hand. Existiert die Datei nicht, wird sie automatisch angelegt. Dies liegt daran, dass bei einem Aufruf ohne `StandardOpenOption` automatisch `CREATE` und `TRUNCATE_EXISTING` angehängt werden. `CREATE` erzeugt eine neue Datei, sofern sie noch nicht existiert, `TRUNCATE_EXISTING` überschreibt den Inhalt der Datei, sofern sie bereits vorhanden ist.

Anhängen leicht gemacht

Der Aufruf von `Files.write(pfad, array, StandardOpenOption.APPEND)` fügt Text am Dateiende hinzu. Existiert die Datei nicht, wird eine `IOException` geworfen. Dies kann durch die `StandardOpenOption.CREATE` vermieden werden. Wie einfach es ist eine Datei zu kopieren oder zu verschieben, ist in Abbildung 3 beschrieben. Auch das Auslesen des Dateiinhaltes ist keine große Kunst. `Files.readAllLines(...)` liefert eine Liste von Strings, über die iteriert werden kann. Wenn die Datei wieder gelöscht werden soll, reicht der Aufruf von `Files.delete(pfad)`.

Dateiattribute lesen

Das Auslesen von Dateiattributen wurde ebenfalls überarbeitet. Während das Auslesen einiger Attribute noch über die `Files`-Klasse möglich ist, wurde in NIO.2 ein neues Paket bereitgestellt.

Dies heißt `java.nio.file.attribute` und enthält File Attribute Views und somit eine flexible Möglichkeit auf unterschiedliche Dateisysteme zu reagieren. Während die Klasse `BasicFileAttributes` allgemeine Dateiattribute bereitstellt, die in allen Dateisystemen vorhanden sind, bietet `DosFileAttributes` Zugriff auf Dateiattribute eines DOS-basierten Dateisystems, sowie `PosixFileAttributes` Zugriff auf Dateiattribute eines Unix-Dateisystems.

Dateiattribute schreiben

Um schreibenden Zugriff auf das Dateisystem zu erhalten, muss man sich der Views bedienen. Während die File Attributes wie z.B. `PosixFileAttributes` lediglich lesenden Zugriff bereitstellen, bietet die `PosixFileAttributeView` schreibenden Zugriff, um z.B. Dateiberechtigungen zu setzen. Abbildung 4 zeigt dies an einem Beispiel.

```
public void showPathInfo() {
    FileSystem fs = FileSystems.getDefault();
    Path pfad = fs.getPath("c:", "ordixnews", "java7");
    System.out.println("Absolut: " + pfad);
    System.out.println("Parent: " + pfad.getParent());
    System.out.println("Root: " + pfad.getRoot());
}
```

Abb. 1: Beispiel für die Klasse Path.

```
public void createAndAppendFile() throws IOException {
    FileSystem fs = FileSystems.getDefault();
    Path pfad = fs.getPath("c:", "ordixnews", "java7",
        "myFile.txt");
    byte[] array = new String("Schreibtest").getBytes();
    Files.write(pfad, array);
    byte[] zusatzText = new String("mehr Text").getBytes();
    Files.write(pfad, zusatzText, StandardOpenOption.CREATE,
        StandardOpenOption.APPEND);
}
```

Abb. 2: Beispiel für CreateandAppend.

```
public void copyAndMoveFile() throws IOException {
    FileSystem fs = FileSystems.getDefault();
    Path source = fs.getPath("c:", "ordixnews", "java7", "myFile.txt");
    Path target = fs.getPath(source.getParent().toString(),
        "fileCopy.txt");
    Files.copy(source, target,
        StandardCopyOption.COPY_ATTRIBUTES);
    target = fs.getPath(source.getParent().toString(),
        "movedFile.txt");
    Files.move(source, target,
        StandardCopyOption.REPLACE_EXISTING);
}
```

Abb. 3: Beispiel für CopyandMove.

Auflisten aller Dateien in einem Pfad

NIO.2 wurde auch hinsichtlich der Performance optimiert. Per `DirectoryStream` ist es nun auf sehr einfache und speicherfreundliche Weise möglich, alle Dateien in einem Pfad auszulesen. Wie der Name schon andeutet, handelt es sich hierbei um einen Stream. Wenn es darum geht Dateien aufzulisten, ist dies einer der großen Vorteile gegenüber den vorherigen Java-Versionen, .

Bis Java 7 wurde über `File.listFiles()` der komplette Inhalt eines Verzeichnisses ausgelesen. Der Nachteil hierbei ist, dass sofort alle Dateien erfasst und in ein Array verpackt werden. Dies bringt eine unnötige Speicherreservierung mit sich, sofern nicht alle Dateien für eine Operation benötigt werden.

```

public void fileAttributes() throws IOException {
    FileSystem fs = FileSystems.getDefault();
    Path pfad = fs.getPath("/ordixnews/myFile.txt");

    Files.isWritable(pfad);
    Files.isExecutable(pfad);
    Files.isHidden(pfad);
    Files.getOwner(pfad);

    PosixFileAttributes atts =
    Files.readAttributes(pfad, PosixFileAttributes.class);

    for (PosixFilePermission perm : atts.permissions()) {
        System.out.println(perm);
    }

    // bedient sich intern der PosixFileAttributeView
    Files.setPosixFilePermissions(pfad,
    PosixFilePermissions.fromString("rwxrwxr--"));
}

```

Abb. 4: Dateiattribute auslesen und schreiben.

```

public void listFiles() {
    try (DirectoryStream<Path> stream =
    Files.newDirectoryStream(PFAD)) {
        for (Path p : stream) {
            Files.isDirectory(p);
        }
    } catch (IOException e) {
        e.printStackTrace();
    }
}

```

Abb. 5: Auflisten von Dateien per DirectoryStream.

```

private static void removeDirectories() throws IOException
{
    if (!Files.exists(PATH)) {
        return;
    }
    Files.walkFileTree(PATH, new SimpleFileVisitor<Path>() {
        @Override
        public FileVisitResult visitFile(Path file,
        BasicFileAttributes attrs) throws IOException {
            Files.delete(file);
            return FileVisitResult.CONTINUE;
        }
        @Override
        public FileVisitResult postVisitDirectory(Path dir,
        IOException e) throws IOException {
            if (e == null) {
                Files.delete(dir);
                return FileVisitResult.CONTINUE;
            } else {
                // directory iteration failed
                throw e;
            }
        }
    });
}

```

Abb. 6: Rekursives Wandern durch den Verzeichnisbaum.

Speichereffizient

Mit Java 7 wurde der `DirectoryStream` eingeführt, welcher u.a. das `Iterable` Interface implementiert. Dadurch werden die Dateiojekte erst mit einem `next()` nach und nach in den Speicher geladen.

`DirectoryStream` implementiert aber nicht nur das Interface `Iterable` um sehr einfach über die Ergebnismenge von `Path`-Objekten zu iterieren, sondern auch das Interface `Closable`. Wie wir im ersten Artikel dieser Reihe [1] gelernt haben, bietet uns die Verwendung solcher Implementierungen die Möglichkeit, den Stream nicht mehr manuell schließen zu müssen. Dazu instanzieren wir den `DirectoryStream` einfach im Header der Anweisung `try` des `try-catch`-Blocks (siehe Abbildung 5).

Dateifilter

Doch dies ist noch nicht alles. Der `DirectoryStream` bietet ebenfalls über die Angabe von Filtern z.B. die Möglichkeit nur Dateien bis zu einer bestimmten Größe aufzulisten oder aber durch Angabe von `Regex`-Anweisungen nach Dateinamen zu filtern. Die Implementierung des Filters ist im Vergleich zu vorherigen Java-Versionen insofern gleich geblieben, dass die Methode `accept` überschrieben werden kann. Allerdings befindet er sich nun im `DirectoryStream` und nennt sich nicht mehr `FileFilter` sondern `Filter`.

Verzeichnisbäume rekursiv durchstöbern

Die neue Methode `walkFileTree` der Klasse `Files` ermöglicht es, rekursiv durch einen Verzeichnisbaum zu wandern. Der Methode werden der Startpfad und eine `FileVisitor`-Implementierung übergeben. Diese beinhaltet Verhaltensmuster, wie mit Dateien und Verzeichnissen umgegangen werden soll. Im Falle von Verzeichnissen sogar vor und nach eines jeden „Besuches“. Die Methoden lassen sich bereits am Namen identifizieren:

- `FileVisitor.visitFile`
- `FileVisitor.visitFileFailed`
- `FileVisitor.preVisitDirectory`
- `FileVisitor.postVisitDirectory`

Zurückgeliefert wird jeweils ein `FileVisitResult` mit dem bestimmt werden kann, ob die Suche fortgesetzt werden soll.

Darüber hinaus lässt sich noch die maximale Suchtiefe konfigurieren. So ist es möglich über das Setzen des Parameters `maxDepth` zu bestimmen, dass der Baum nur bis zur Ebene `x` durchsucht werden soll. Ebenfalls lassen sich symbolische Links verfolgen, wenn dies gewünscht wird.

Eine einfache Implementierung des `FileVisitor` liefert Oracle mit. Der `SimpleFileVisitor` implementiert die oben genannten Methoden rein lesend. Es ist allerdings einfach möglich die Methoden zu überschreiben, um z.B. aus dem `SimpleFileVisitor` ei-

nen eigenen `FileVisitor` zu schreiben, der Dateien und Verzeichnisse rekursiv löscht (siehe Abbildung 6).

WatchService is notifying you

Java-Programmierer, die eine Verzeichnisüberwachung implementieren mussten, wurden bisher dazu gezwungen per „polling“ Datei- oder Verzeichnisänderungen zu erkennen. Diese nicht große aber unschöne Hürde nimmt uns nun der `WatchService` ab. Er überwacht alle Objekte, die das `Watchable` Interface implementieren. Aktuell ist dies nur die Klasse `Path`. Überwacht werden können die Aktionen `ENTRY_CREATE`, `ENTRY_MODIFY` und `ENTRY_DELETE`.

Die Abbildung 7 zeigt, wie eine Überwachung aussehen könnte. Während `pfad.register(...)` eine Repräsentation auf das überwachende Objekt zurückgibt, steuert man mit `watcher.take()` eine Abfrage auf neue Ereignisse. Während `watcher.poll()` durch Rückgabe von null die Abarbeitung des Programms fortführt, auch wenn keine Ereignisse eingetroffen sind, wartet `take()` blockierend auf ein eintreffendes Ereignis. Per `event.context()` gelangt man an das Objekt der Begierde. Nach der Verarbeitung des oder der Ereignisse muss der Status des Key per `reset()` wieder auf `ready` gesetzt werden, sodass er für neue oder zwischenzeitlich eingetretene Ereignisse „re-queued“ werden kann.

Polling Fallback

Erwähnenswert ist, dass der `WatchService` per JNI die Mittel des Betriebssystems nutzt. Dies bedeutet zum einen, dass der Aufruf nicht komplett blockierend ist, aber auch, dass der `WatchService` den Benachrichtigungsmechanismus des Betriebssystems nur nutzen kann, wenn das Betriebssystem diesen bereitstellt. Ist dies nicht der Fall, übernimmt dieser das Polling der Dateisystemänderungen.

Und sonst?

Die dargestellten Aspekte geben einen kleinen Einblick in die neuen FileSystem-Funktionen. Zahlreiche Erweiterungen gibt es in der asynchronen I/O. Darüber hinaus wird die Unterstützung des Socket Direct Protocol (SDP) unter Solaris und Linux sowie TLS 1.2 und IPv6 in NIO.2 verortet. Alle Informationen zu NIO.2 finden Sie auch in der Java 7 API Specification [3].

Fazit

Zurück von der Reise durch NIO.2 nehmen wir einige interessante Eindrücke mit. Es ist jetzt nicht nur um

```
public void watch() throws InterruptedException,
IOException {
    Path pfad= FileSystems.getDefault().
    getPath("c:\\ordixnews\\java7");

    WatchService watcher = FileSystems.getDefault().
    newWatchService();

    WatchKey key = pfad.register(watcher, StandardWatchEvent
    Kinds.ENTRY_CREATE);

    key = watcher.take();
    for (WatchEvent<?> event : key.pollEvents()) {
        if (event.kind().equals(StandardWatchEventKinds.
        ENTRY_CREATE)) {
            Path path = (Path) event.context();
            System.out.println("Neuer Eintrag: " +
            path.getFileName());
        }
    }
    key.reset();
}
```

Abb. 7: Beispiel für WatchService.

Links

- ▶ [1] ORDIX news Artikel 04/2011: „Java 7 - Was bringt die neue Version? Project Coin (Teil I)“: http://www.ordix.de/ORDIXNews/4_2011/java7.html
- ▶ [2] ORDIX news Artikel 01/2012: „Java 7 - Was bringt die neue Version? Neuigkeiten aus dem parallelen Universum (Teil II)“: http://www.ordix.de/ORDIXNews/1_2012/java7_concurrency.html
- ▶ [3] Java 7 API Specification: <http://docs.oracle.com/javase/7/docs/api/>

einiges einfacher Dateien zu kopieren und zu verschieben, sondern auch auf eine sehr elegante Art und Weise möglich, auf Ereignisse in Verzeichnissen zu reagieren. Wer regelmäßig in Projekten die I/O verwendet, wartet sicherlich schon lange auf diese Änderungen und wird den Einzug von NIO.2 sehnsüchtig erwarten. Insbesondere den `WatchService` sowie das rekursive Durchsuchen des Verzeichnisbaumes wird man nicht mehr missen wollen.

Eine neue Schreibweise für numerische Literale, Strings in Switch-Anweisungen, ein verbessertes Exception Handling, das neue Fork/Join Framework aus dem Concurrency Framework sowie diverse Änderungen in der I/O haben wir in dieser ORDIX news Reihe kennengelernt. Nun sind wir am Ende unserer Java 7 Reihe angekommen und haben Ihnen einen kurzen Einblick in die Neuheiten von Java 7 gegeben.



Lars Brenker
(info@ordix.de).



Traditionell oder evolutionär

Klassisches vs. agilem Projektmanagement

Es gibt viele erfolgreiche Verfahren zur Umsetzung von Anforderungen im Rahmen eines Projektes. Häufig stößt man dabei auf mindestens zwei Lager: das klassische und das agile Projektmanagement. Im vorliegenden Artikel werden beide Ansätze miteinander verglichen und die geeigneten Einsatzgebiete vorgestellt.

Unterschiedliche Philosophien

Das klassische Projektmanagement in Unternehmen ist an ein Vorgehensmodell (VGM) gebunden, das an bestehende, standardisierte Projektmanagementmethoden, wie z.B. IPMA (GPM), PMI oder PRINCE2 angelehnt ist. Es ist planerisch, eher sequentiell und in disjunkten Phasen organisiert.

Dabei wird ein überwiegend ingenieurstechnischer Ansatz verfolgt, der mit Hilfe von Methoden und Werkzeugen einen Prozess steuert und Aufgaben delegiert. Die Aufgaben werden dabei am Anfang klar vereinbart. Änderungen werden eher vermieden und wenn doch

notwendig, über einen speziellen Änderungsprozess später integriert.

Das agile Projektmanagement steht als Oberbegriff für eine empirische und adaptive Steuerung von Projekten und Prozessen innerhalb eines vorgegebenen organisatorischen Rahmens.

In der Praxis finden sich verschiedene Ausprägungen, wie z.B. Scrum oder APM, wobei der Ansatz zur empirischen Prozesskontrolle immer gemeinsam ist:

- **Transparenz** über die Anforderungen und den Fortschritt des Projektes in Richtung des vorgegebenen Ziels

- **Überprüfung**, ob die gesetzten Ziele innerhalb vorher bestimmten Abweichungskorridoren erreicht werden konnten
- **Anpassung** des Prozesses oder des Arbeitsgegenstandes bei nicht tolerabler Abweichung

Durch die empirische Prozesskontrolle sind agile Prozesse inkrementell und iterativ. Die im Jahr 2001 vereinbarten Leitsätze im Agilen Manifest, betonen die Wichtigkeit von Individuen und Interaktion, die Fertigstellung von Teilergebnissen, die vertrauensvolle Zusammenarbeit mit dem Kunden, sowie die jederzeit mögliche Anpassung an neue Anforderungen.

Anforderungsmanagement

Der Schwerpunkt des Anforderungsmanagements in den klassischen Verfahren liegt auf der frühzeitigen Erreichung von Vollständigkeit und Eindeutigkeit. Meist sind die aus dem Anforderungsmanagement stammenden Informationen die Basis für langfristige Zeit- und Aufwandsplanungen, sowie Grundlage eines Vertrages und darin enthaltener Fristen zur Fertigstellung. Eine Veränderung der Anforderungen wird vermieden, um diese Grundlagen nicht wiederholt aufwendig anpassen zu müssen.

Die Anforderungen werden bei der agilen Vorgehensweise durch „User Stories“ (ähnlich zu Use Cases) beschrieben. Diese werden in einem Product-Backlog gesammelt, bewertet und priorisiert. Dieses Product-Backlog kann während des Projektverlaufs ständig erweitert und reduziert sowie re-priorisiert werden. Die am höchsten priorisierten Anforderungen kommen als nächstes zur Umsetzung. Der Schwerpunkt liegt hier auf der Adaptionfähigkeit an die Anforderungen des Kunden.

Organisation

Ein Projekt gliedert sich nach der klassischen Vorgehensweise in der Regel in mehrere Phasen. Minimal können die folgenden Phasen betrachtet werden:



Abb. 1: Projektphasen als Prozessfolge.

- Projektinitialisierung
- Projektplanung
- Projektumsetzung
- Projektabschluss

Die Fortschrittsverfolgung erfolgt zumeist über Meilensteine, die während der Planungsphase festgelegt werden (siehe Abbildung 1).

Um den agilen Anforderungen nachzukommen, ohne sich in zu viele Änderungen zu verlieren, nutzen agile Verfahren Timeboxes zur Fokussierung. Eine Timebox in Scrum wird als Sprint bezeichnet.

Innerhalb eines Projektes wird als Sprint eine gleiche, weitestgehend konstante Zeitdauer vorab vereinbart. Der Sprint setzt sich immer aus Planungsphase, Umsetzungsphase und Abschlussphase zusammen.

Zu Beginn des Sprints werden die am höchsten priorisierten Anforderungen aus dem Product-Backlog zur Umsetzung geplant. Die Fortschrittsverfolgung erfolgt automatisch durch das am Ende des Sprints fertige auslieferbare Product-Increment (siehe Abbildung 2).

Rollen

Im klassischen Projektmanagement gibt es überwiegend hierarchisch aufgebaute Rollen zur Steuerung der Umsetzung wie Projektleiter, Projektteam, Lenkungsausschuss, Auftraggeber, usw. Diese wer-

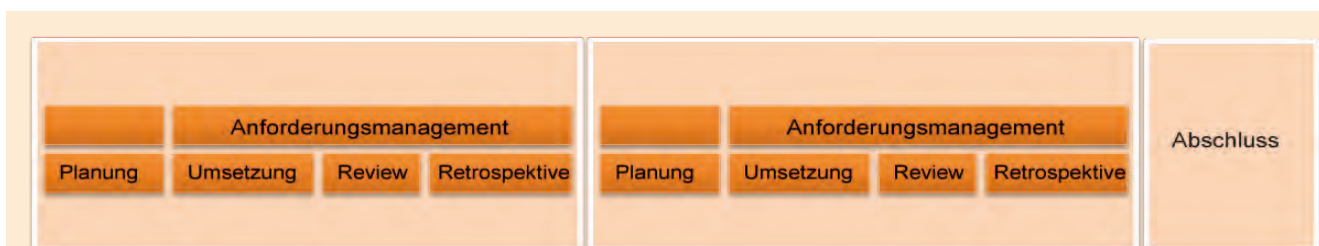


Abb. 2: Timeboxen mit jeweils Planung, Umsetzung und Abschluss.

Aufgabenbereich	Scrum-Rollen
Scope-Management	Product Owner für Produktversion, Team für Sprints
Zeitmanagement	Product Owner für Release-Plan, Team für Sprint Backlog
Kostenmanagement	Product Owner
Kommunikations-Mgmt	Product Owner für Release Reporting, Team für Sprint Reporting
Risikomanagement	Product Owner mit Input vom Team
Qualitätsmanagement	Product Owner, Team, Scrum Master
Lieferantenmanagement	Product Owner und Team
Personalmanagement	Management, Team für Produktivität und KVP

Abb. 3: Typische Rollen des Projektmanagements.



Abb.4: Rollenbild mit Scrum Master, Product Owner und Team.

Glossar

Agiles Manifest

Vorgehensmodell der agilen Softwareentwicklung. Zugrundeliegen zwölf Prinzipien, die bei der Softwareentwicklung berücksichtigt werden sollen.

APM

Agile Project Management - Generalisierter Projekt Management Prozess, der auf agilen Methoden aus der Softwareentwicklung aufsetzt.

Product Backlog

Das Product Backlog ist eine priorisierte Liste, die alles enthält, was im zu entwickelnden Produkt enthalten sein sollte.

Product Owner

Der Product Owner ist verantwortlich für die Anforderungen und deren Priorisierung in Zusammenarbeit mit dem Auftraggeber und Know-how-Träger.

Scrum

Vorgehensmodell der Softwaretechnik. Der Ansatz von Scrum ist empirisch, inkrementell und iterativ.

Scrum Master

Der Scrum Master unterstützt das Team im Scrum-Prozess bei der Umsetzung der Anforderungen.

Sprint

Iteration fester Länge zur Umsetzung von Anforderungen.

Timebox

Ein fester Zeitrahmen für das Projekt oder einen Vorgang im Projekt.

den zur Unterstützung des Teams ergänzt um spezielle fachliche Funktionen, wie z.B. Qualitätssicherung oder Testmanagement (siehe Abbildung 3).

Der Projektleiter steuert das Projektteam verantwortlich, delegiert Aufgaben an Teammitglieder und sammelt das Feedback zur Konsolidierung ein.

Im agilen Projektmanagement am Beispiel von Scrum gibt es ein Scrum Team. Dieses organisiert sich eigenverantwortlich, ist interdisziplinär zusammengesetzt und trägt als Team die Verantwortung für die Umsetzung.

Das Scrum Team setzt sich zusammen aus:

- **Product Owner** als Verantwortlicher des Product Backlog und somit Vertreter des Auftraggebers. Er bestimmt die Reihenfolge der Umsetzung der Anforderungen.
- **Development Team** als verantwortliche Planer der Umsetzung, Umsetzer und Steuernde während des Sprints. Es ist auch verantwortlich für die Prozessverbesserung.
- **Scrum Master** ist der Coach und Moderator des Umsetzungsprozesses (Servant Leadership).

Die Abbildung 4 verdeutlicht das Zusammenspiel der einzelnen Rollen.

Stärken und Schwächen

Das klassische Projektmanagement funktioniert hervorragend, wenn die Anforderungen während der Planungsphase möglichst vollständig bekannt sind. Dann kann das Ziel ohne Umwege angesteuert werden. Über die verschiedenen existierenden Vorgehensmodelle sind die Abläufe bekannt und können mit Best-Practices unterstützt werden.

Das agile Projektmanagement kann seine Stärken ausspielen, wenn viele Prämissen während der Planungsphase noch nicht bekannt sind bzw. im Projekt erarbeitet werden müssen. Durch den empirischen Prozess kann jederzeit nachgesteuert werden. Er bedingt jedoch auch einen weitaus höheren Steuerungs- und Verwaltungsaufwand. Durch die iterative Vorgehensweise mit Sprints erzeugt Scrum mehr Transparenz sowie regelmäßige Produktlieferungen. Hindernisse werden früher sichtbar und das Produkt kann durch die inkrementelle Vorgehensweise bei Bedarf schneller geändert werden.

Im agilen Projektmanagement werden an die Teammitglieder besonders hohe Anforderungen in Bezug auf Eigenverantwortlichkeit, Disziplin und Kommunikation gestellt. Insbesondere der Scrum Master muss über ausgeprägte Soft Skills verfügen. Jeder Sprint

muss ein potentiell auslieferbares Produkt als Ergebnis liefern, wofür das Team die Verantwortung trägt.

Fazit

Sind die Voraussetzungen und Rahmenbedingungen sowie die Anforderungen zur Erreichung des Ziels eines Projektes im voraus ausreichend bekannt, so wird das Ziel mit dem klassischen Projektmanagement meist ohne notwendige Umwege erreicht. Ein Projekt zur Bereitstellung einer Infrastruktur, basierend auf einer bekannten Architektur, ist eine gutes Anwendungsbeispiel des klassischen Projektmanagements.

Für Projekte deren Anforderungsumfang zu Beginn nur hinreichend, jedoch nicht ausreichend spezifiziert ist und somit ein präziser, zielführender Planungsverlauf am Anfang nicht klar erkennbar ist, eignen sich agile Verfahren, wie z.B. Scrum. Als Anwendungsbeispiele sind Projekte zur Verbesserung von Prozessen oder Verfahren zu nennen, sowie Softwareprojekte mit initial unscharf vorliegenden Vorgaben und hohem Komplexitätsmaß.

Links

- ▶ [1] Webseite der coniatos AG: <http://www.coniatos.de>
- ▶ [2] ScrumAlliance: <http://www.scrumalliance.org>
- ▶ [3] Agiles Manifest: <http://www.agilemanifesto.org/iso/de/>
- ▶ [4] Projektmanagement-Seminare der coniatos AG/ORDIX AG: <http://training.ordix.de/siteengine/action/load/kategorie/Projektmanagement/index.html>

Bild: © sxc.hu | Levitation 1 | biziör
 © sxc.hu | spring | dlInny
 © istockphoto.com | Businessmann Sits in... | PeskyMonkey



Lars Eisenblatt
 (info@coniatos.de).

Rätsel zu den Sommerferien

Larry macht Urlaub

Larry verbringt seine freien Tage mit seinem Kollegen Max an einem Fluss. Sie sitzen am Ufer und genießen den Blick aufs Wasser. Damit die beiden nicht „einrosten“, beschließen sie Sport und Gehirnjogging zu verbinden.



Sie stellen sich die Frage, wie breit der Fluss ist?

Dazu schwimmt zunächst Max zum gegenüberliegenden Ufer. Dann schwimmen sie durch einen Fluss mit unterschiedlichen, aber konstanten Geschwindigkeiten. Sie treffen sich das erste Mal 80 Meter vom nördlichen Ufer. Die beiden schwimmen aneinander vorbei, wenden an den Ufern und schwimmen wieder beide zurück.

Sie begegnen sich das zweite Mal 40 Meter vom südlichen Ufer entfernt. Beide Male kamen sich die Schwimmer entgegen.

Nach dieser sportlichen Erfrischung setzen sich beide wieder gemeinsam ans Ufer. Sie versuchen nun die Breite des Flusses zu berechnen.

Wissen Sie, wie breit der Fluss ist?

Larry und Max freuen sich auf Ihren Lösungsvorschlag. Senden Sie bis zum **06. Juli 2012** Ihre Antwort an kniffel@ordix.de.

Das Passwort-Problem wurde nicht gelöst!

Wir haben mit Hilfe unserer Referenten eine Lösung gefunden, diese Beispiel-Lösung finden Sie als SQL-Datei im Internet unter:

http://www.ordix.de/ORDIXNews/verzeichnis_22012.html

Datenbanken				Juni - Dezember 2012	
DB-DB-02	Datenbank-Hochverfügbarkeitslösungen für Entscheider	1 Tag	550,00 €	01.06.2012 29.10.2012 10.12.2012	
DB-DB-03	Datawarehouse Grundlagen	3 Tage	1.290,00 €	02.07.2012 29.10.2012 10.12.2012	
DB-DB-01	Datenbank-Modellierung	2 Tage	890,00 €	26.07.2012 22.10.2012 13.12.2012	
DB-ORA-01	Oracle SQL	5 Tage	1.890,00 €	11.06.2012 30.07.2012 24.09.2012 03.12.2012	
DB-ORA-01A	Oracle SQL für Experten	3 Tage	1.290,00 €	23.07.2012 29.10.2012 17.12.2012	
DB-ORA-02	Oracle Datenbankprogrammierung mit PL/SQL Grundlagen	5 Tage	1.890,00 €	25.06.2012 27.08.2012 15.10.2012 19.11.2012	
DB-ORA-34	Oracle Datenbankprogrammierung mit PL/SQL Aufbau	5 Tage	1.890,00 €	18.06.2012 27.08.2012 24.09.2012 12.11.2012	
DB-ORA-42	Oracle PL/SQL Tuning	3 Tage	1.890,00 €	18.06.2012 03.09.2012 19.11.2012	
DB-ORA-44	Entwicklung von Web-Anwendungen mit Oracle PL/SQL	2 Tage	1.090,00 €	21.06.2012 30.08.2012 27.09.2012 06.12.2012	
DB-ORA-43	Oracle APEX Programmierung	3 Tage	1.890,00 €	11.06.2012 06.08.2012 17.09.2012 15.10.2012	
DB-ORA-38	Objektorientierung in Oracle	5 Tage	1.990,00 €	13.08.2012 12.11.2012	
DB-ORA-03	Oracle Datenbankadministration Grundlagen	5 Tage	1.990,00 €	18.06.2012 20.08.2012 10.09.2012 22.10.2012	
DB-ORA-04	Oracle Datenbankadministration Aufbau	5 Tage	1.990,00 €	02.07.2012 06.08.2012 03.09.2012 26.11.2012	
DB-ORA-06	Manuelles Oracle Backup und Recovery & weitere Konzepte	5 Tage	1.990,00 €	auf Anfrage	
DB-ORA-32	Oracle Backup und Recovery mit RMAN	5 Tage	1.990,00 €	18.06.2012 16.07.2012 24.09.2012 12.11.2012	
DB-ORA-07	Oracle Tuning und Monitoring	5 Tage	1.990,00 €	23.07.2012 10.09.2012 05.11.2012	
DB-ORA-41	Oracle AWR und ASH Analyse und Interpretation	3 Tage	1.290,00 €	20.08.2012 29.10.2012 12.12.2012	
DB-ORA-11	Oracle Troubleshooting Workshop	5 Tage	1.990,00 €	25.06.2012 10.09.2012 03.12.2012	
DB-ORA-08	Oracle 11gR2 RAC und Grid Infrastructure	5 Tage	1.990,00 €	02.07.2012 06.08.2012 22.10.2012 26.11.2012	
DB-ORA-15	Oracle 11g Neuheiten	5 Tage	1.990,00 €	25.06.2012 20.08.2012 03.09.2012 19.11.2012	
DB-ORA-33A	Oracle Security	4 Tage	1.590,00 €	16.07.2012 17.09.2012 03.12.2012	
DB-ORA-31	Oracle Data Guard	4 Tage	1.590,00 €	02.07.2012 24.09.2012 22.10.2012 26.11.2012	
DB-ORA-35	Oracle Grid Control	3 Tage	1.290,00 €	13.08.2012 08.10.2012 10.12.2012	
DB-ORA-36	Oracle Replikation	3 Tage	1.290,00 €	auf Anfrage	
DB-ORA-37	Oracle Streams Advanced Queuing	3 Tage	1.290,00 €	30.07.2012 19.11.2012	
DB-ORA-39	Oracle Migration und Patching	3 Tage	1.290,00 €	06.08.2012 05.11.2012	
DB-ORA-40	Oracle Capacity Planning	3 Tage	1.290,00 €	auf Anfrage	
DB-INF-01	IBM Informix SQL	5 Tage	1.790,00 €	30.07.2012 08.10.2012	
DB-INF-02	IBM Informix Administration	5 Tage	1.990,00 €	20.08.2012 22.10.2012	
DB-INF-04	IBM Informix Dynamic Server Backup und Recovery	3 Tage	1.290,00 €	03.09.2012 19.11.2012	
DB-INF-03	IBM Informix Tuning und Monitoring	5 Tage	1.990,00 €	18.06.2012 24.09.2012	
DB-INF-07	IBM Informix Hochverfügbarkeits-Technologien unter Unix	4 Tage	1.590,00 €	13.08.2012 15.10.2012	
DB-DB2-01	IBM DB2 für Linux/Unix/Windows SQL Grundlagen	5 Tage	1.890,00 €	25.06.2012 10.09.2012 05.11.2012	
DB-DB2-02	IBM DB2 für Linux/Unix/Windows Administration	5 Tage	1.990,00 €	13.08.2012 12.11.2012	
DB-DB2-05	IBM DB2 für Linux/Unix/Windows Monitoring und Tuning	3 Tage	1.290,00 €	16.07.2012 08.10.2012	
DB-MY-01	MySQL Administration	3 Tage	1.190,00 €	16.07.2012 24.09.2012 26.11.2012	
DB-MSSQL-1	Microsoft SQL Server Administration	5 Tage	1.790,00 €	auf Anfrage	
Entwicklung				Juni - Dezember 2012	
P-PHP-01	PHP Programmierung Grundlagen	5 Tage	1.690,00 €	06.08.2012 26.11.2012	
P-PHP-02	PHP Programmierung Aufbau	3 Tage	1.190,00 €	20.08.2012 03.12.2012	
P-PERL-01	Perl Programmierung Grundlagen	5 Tage	1.690,00 €	16.07.2012 15.10.2012	
P-PERL-02	Perl Programmierung Aufbau	5 Tage	1.690,00 €	25.06.2012 20.08.2012 05.11.2012	
P-UNIX-01	Shell Awk und Sed	5 Tage	1.690,00 €	11.06.2012 03.09.2012 12.11.2012	
P-UNIX-01A	Awk Intensiv-Workshop	3 Tage	1.190,00 €	02.07.2012 29.10.2012	
P-XML-01	Einführung in XML	3 Tage	1.190,00 €	04.06.2012 17.09.2012 26.11.2012	
P-XML-02	XML Programmierung unter Java mit DOM und SAX	2 Tage	890,00 €	auf Anfrage	
P-XML-03	Oracle und XML	3 Tage	1.190,00 €	auf Anfrage	
Web- und Applikations-Server				Juni - Dezember 2012	
INT-04	Apache Web-Server Installation und Administration	3 Tage	1.190,00 €	30.07.2012 03.09.2012 15.10.2012 10.12.2012	
INT-07	Tomcat Konfiguration und Administration	3 Tage	1.190,00 €	16.07.2012 24.09.2012 19.11.2012	
INT-08	WebSphere Application Server Installation und Administration	3 Tage	1.390,00 €	04.06.2012 03.09.2012 03.12.2012	
INT-11	Administration und Konfiguration für JBoss	3 Tage	1.190,00 €	02.07.2012 17.09.2012 26.11.2012	

Informationen und Anmeldung

Für Informationen und Fragen zu individuell zugeschnittenen Seminaren, Ausbildungsreihen oder Inhouse-Schulungen stehen wir Ihnen gerne zur Verfügung. Auf Wunsch senden wir Ihnen auch unser komplettes Seminarprogramm zu.

Online-Anmeldung,
aktuelle Seminarinhalte
und Termine unter:

<http://training.ordix.de>



Systemmanagement				Juni - Dezember 2012	
SM-NAG-01	Systemüberwachung mit Nagios Grundlagen	3 Tage	1.190,00 €	18.06.2012 27.08.2012 22.10.2012	
SM-NAG-02	Systemüberwachung mit Nagios Aufbau	2 Tage	890,00 €	21.06.2012 30.08.2012 25.10.2012	
Java-JEE				Juni - Dezember 2012	
E-UML-01	Anforderungsanalyse mit UML	3 Tage	1.190,00 €	04.06.2012 27.08.2012 15.10.2012 19.11.2012	
E-SWA-01	Softwarearchitekturen	5 Tage	1.890,00 €	25.06.2012 24.09.2012 22.10.2012 03.12.2012	
OO-01	Einführung in die Objektorientierte Programmierung	3 Tage	1.190,00 €	02.07.2012 17.09.2012 12.12.2012	
P-JAVA-01	Java Programmierung Grundlagen	5 Tage	1.690,00 €	23.07.2012 08.10.2012 17.12.2012	
P-JAVA-03	Java Programmierung Aufbau	5 Tage	1.690,00 €	11.06.2012 30.07.2012 05.11.2012	
P-JAVA-02	Java GUI Entwicklung mit Swing	5 Tage	1.690,00 €	11.06.2012 10.09.2012 10.12.2012	
P-JEE-01	JEE für Entscheider	1 Tag	590,00 €	16.07.2012 15.10.2012	
P-JEE-02	Einführung in JEE	3 Tage	1.290,00 €	17.07.2012 16.10.2012	
P-JEE-03A	JSP und Servlet Programmierung	5 Tage	1.590,00 €	18.06.2012 27.08.2012 03.12.2012	
P-JEE-04	EJB Programmierung	5 Tage	1.590,00 €	11.06.2012 06.08.2012 22.10.2012	
P-JEE-05	Web-Anwendungen mit JavaServer Faces (JSF)	5 Tage	1.590,00 €	20.08.2012 12.11.2012	
P-JEE-06	Entwickeln mit dem Spring-Framework	3 Tage	1.190,00 €	04.06.2012 27.08.2012 26.11.2012	
INT-05	Java Web Services	3 Tage	1.190,00 €	02.07.2012 08.10.2012 17.12.2012	
J-HIB-01	Hibernate und die Java Persistence API	5 Tage	1.690,00 €	25.06.2012 10.09.2012 19.11.2012	
P-JEE-08	Java Performance Tuning	3 Tage	1.290,00 €	23.07.2012 29.10.2012	
Betriebssysteme				Juni - Dezember 2012	
BS-01	Unix/Linux Grundlagen für Einsteiger	5 Tage	1.690,00 €	18.06.2012 20.08.2012 03.09.2012 05.11.2012	
BS-25	Unix AufbauSeminar für Datenbank- und Applikationsbetreuer	5 Tage	1.790,00 €	25.06.2012 27.08.2012 05.11.2012 19.11.2012	
BS-02	Linux Systemadministration	5 Tage	1.690,00 €	02.07.2012 10.09.2012 08.10.2012 19.11.2012	
BS-09	Linux Hochverfügbarkeits-Cluster	3 Tage	1.290,00 €	20.08.2012 08.10.2012	
BS-19	Linux Cluster mit Pacemaker und Heartbeat 3	3 Tage	1.190,00 €	27.08.2012 29.10.2012 12.11.2012	
BS-16	OpenLDAP - Praxiseinsatz im Netzwerk	4 Tage	1.490,00 €	11.06.2012 27.08.2012 22.10.2012	
BS-03	Solaris Systemadministration Teil I	5 Tage	1.990,00 €	23.07.2012 24.09.2012 03.12.2012	
BS-04	Solaris Systemadministration Teil II	5 Tage	1.990,00 €	06.08.2012 08.10.2012 10.12.2012	
BS-06	Solaris kompakt für erfahrene Unix/Linux-Umsteiger	5 Tage	1.990,00 €	25.06.2012 17.09.2012 17.12.2012	
BS-24	Solaris 11 Administration Neuheiten	5 Tage	1.990,00 €	auf Anfrage	
BS-18	Solaris Virtualisierung mit ZFS und Container (Zonen)	5 Tage	1.990,00 €	11.06.2012 10.09.2012 12.11.2012	
BS-23	Solaris Virtualisierung mit LDOM	3 Tage	1.290,00 €	30.07.2012 15.10.2012	
AIX-01	IBM AIX Systemadministration Grundlagen	5 Tage	1.990,00 €	13.08.2012 22.10.2012	
AIX-02	IBM AIX Installation, Backup und Recovery mit NIM	3 Tage	1.290,00 €	17.09.2012 03.12.2012	
AIX-03	Analyse v. komplexen AIX Performance Problemen	5 Tage	2.300,00 €	20.08.2012 10.12.2012	
Projektmanagement				Juni - Dezember 2012	
PM-01	IT-Projektmanagement - Methoden und Techniken	5 Tage	1.990,00 €	06.08.2012 19.11.2012	
PM-06	System. Projektmanagement - Projektteams souverän führen	4 Tage	1.850,00 €	30.07.2012 08.10.2012	
PM-08	Agiles Projektmanagement mit SCRUM	2 Tage	1.100,00 €	11.06.2012 10.09.2012 06.12.2012	
PM-09	Capability Maturity Model Integration (CMMI)	2 Tage	1.100,00 €	14.06.2012 13.09.2012 26.11.2012	
PM-10	Kennzahlen der IT	2 Tage	1.100,00 €	18.10.2012 13.12.2012	
PM-07	Krisenmanagement in Projekten	2 Tage	1.100,00 €	03.09.2012 12.11.2012	
PM-05	IT-Projektcontrolling	3 Tage	1.290,00 €	23.07.2012 29.10.2012	
PM-11	Konfliktmanagement	2 Tage	1.100,00 €	06.09.2012 15.11.2012	
IT-Management				Juni - Dezember 2012	
MGM-02	IT-Architekturen	3 Tage	1.650,00 €	20.08.2012 05.11.2012	
MGM-01	E-Business	2 Tage	1.100,00 €	23.08.2012 08.11.2012	
MGM-07	IT-Strategien	2 Tage	1.100,00 €	25.06.2012 24.09.2012 10.12.2012	
MGM-08	Strategien effizient entwickeln	3 Tage	1.650,00 €	02.07.2012 17.12.2012	
MGM-03	IT-Management	5 Tage	1.990,00 €	16.07.2012 17.09.2012 03.12.2012	
MGM-05	IT-Risikomanagement	3 Tage	1.650,00 €	18.06.2012 26.09.2012 28.11.2012	
MGM-04	IT-Prozessmanagement	3 Tage	1.650,00 €	13.08.2012 15.10.2012	

Zentrale: **ORDIX AG**
 Westermauer 12 - 16
 33098 Paderborn
 Tel.: 05251 1063-0

Seminarzentrum: **ORDIX AG**
 Kreuzberger Ring 13
 65205 Wiesbaden
 Tel.: 0611 77840-00

Unsere Seminarstandorte sind: Wiesbaden, Bielefeld und Hannover.
 Die Preise gelten pro Seminar pro Teilnehmer in Euro
 zzgl. ges. MwSt., Inhouse-Preise auf Anfrage.



IBM DB2

Indizes und Performance

Die Effizienz bei der Suche nach bestimmten Daten in einer Datenbanktabelle kann durch eine Indexierung deutlich erhöht werden. In einigen Fällen befinden sich die gesuchten Daten bereits im Index, so dass auf die eigentlichen Daten nicht mehr zugegriffen werden muss. Ein Index kann auch optional die Eindeutigkeit einer Spalte sicherstellen. Dieser Artikel zeigt, welche Maßnahmen und Indexierungen sinnvoll sind, um die Performance und nicht die Kosten zu steigern.

Die Anlage von Indizes und die Kosten

Man kann auf einer Tabelle beliebig viele Indizes anlegen, jedoch ist jeder zusätzliche Index mit Kosten verbunden. Jeder Index multipliziert die CPU-Kosten bei den Operationen **insert**, **delete** oder **update**. Bei der Operation **insert** zum Beispiel müssen die Einträge nicht nur für die Zeilen der Tabelle, sondern auch für die korrespondierenden Indizes erstellt werden, was mehr CPU-Leistung erfordert.

Für jeden Index wird zusätzlicher Speicherplatz auf dem Server benötigt. Dies hat zur Folge, dass mehr Zeit beim Reorganisieren der Tabelle, beim Erstellen von Statistiken, beim Laden von Daten und beim Ermitteln von Zugriffplänen benötigt wird.

Wie viele Indizes sind sinnvoll pro Tabelle?

Oder anders ausgedrückt: An welchem Punkt, wird der erhöhte CPU-Aufwand den Performance-Gewinn für einen neuen Index neutralisieren? Die Lösung ist abhängig von der jeweils vorliegenden Situation. Als Erfahrungswert im DB2-Umfeld gilt, dass nicht mehr als vier bis fünf Indizes für Tabellen bei OLTP-Datenbanken verwendet werden sollten. Bei Datawarehouse-Tabellen (OLAP) sollten es nicht mehr als acht bis zehn Indizes sein.

Werden alle Indizes auf der Tabelle auch genutzt?

Um diese Frage zu beantworten, sollte man zunächst mit einem Index-Upgrade der Datenbank beginnen,

um hier ungenutzte Indizes zu suchen und zu eliminieren.

Ab der DB2-Version 9.7 LUW gibt es in der Katalog-View SYSCAT.INDEXES die Spalte LASTUSED, diese eignet sich gut um eine Aussage über die Verwendung der Indizes machen zu können. Der Default-Wert '0001-01-01' ist in der Spalte LASTUSED vorbesetzt. Bei einem Zugriff auf einen Index, steht in dieser Spalte der Datumswert, zu dem der Index zum letzten Mal für ein statisches bzw. dynamisches DML-Statement oder zum Erzwingen eines Constraint genutzt wurde. Steht nach einer gewissen Zeit des Produktivbetriebes immer noch der Default-Wert oder ein weit zurückliegendes Datum in dieser Spalte, so empfiehlt sich dieser Index als Löschkandidat. Eine endgültige Löschung sollte allerdings erst nach weiteren Prüfungen erfolgen.

Welche Tabellenspalten qualifizieren sich für einen Index?

Tabellenspalten, die oft in WHERE-, ORDER BY- und GROUP BY-Bedingungen stehen, sind gute Indexkandidaten. Um eine optimale Performance zu erzielen, sollten bei der Anlage von Indizes gewisse Regeln beachtet werden. Bevor Indizes erstellt werden, sollte zunächst überprüft werden, ob diese nicht bereits bestehen. Dies kann zum Beispiel ein Primärschlüssel oder Unique Key Constraint sein.

Werden mehrere Tabellen häufig in SQL-Abfragen verbunden, sollte die Spalten mit den JOIN-Bedingungen indexiert sein. Indizes auf Spalten mit geringer Selektivität sollten vermieden werden, dies ist zum Beispiel bei einer Spalte „Status“ der Fall, die den Wert 0 oder 1 annimmt. Durch diese Indexierung steht der zusätzliche Speicherverbrauch in einem ungünstigen Verhältnis zum Performance-Gewinn.

Wichtig ist hier eine genaue Betrachtung der einzelnen Spalte und der in Frage kommenden Indexierungen.

Clustered Index

Ein Clustered Index kann für jede Tabelle nur einmal angelegt werden. Hier liegt nicht nur der Index sortiert vor, sondern auch die Daten (in der Reihenfolge des Clustered Index). Bei einer SQL-Abfrage werden bei einem Clustered Index durch die Sortierung der Daten weniger Seitenzugriffe benötigt. Dies hat eine geringere I/O zur Folge. Der Index, der durch die SQL-Abfragen am meisten verwendet wird, sollte dem zufolge als Clustered Index ausgelegt werden.

für einen mehrspaltigen Index:

```
"STRASSE, HAUSNUMMER"
"WHERE STRASSE = 'KREUZBERGERRING'"
benutzt den Index.
```

Während die Bedingung
"WHERE HAUSNUMMER = 13"
diesen nicht verwendet.

Abb. 1: Beispiel eines Composite Index.

```
SELECT ARTID, BEZEICHNUNG FROM ARTIKEL ORDER BY ARTID
```

Dazu einen Unique-Index erstellen:

```
CREATE UNIQUE INDEX IDXARTID ON ARTIKEL(ARTID)
INCLUDE (BEZEICHNUNG)
```

Abb. 2: Abfrage eines Unique Index.

Composite Index

Bei der Anwendung eines Composite Index sollten die Spalten so sortiert sein, dass die Spalte mit der höchsten Selektivität links im Indexschlüssel steht. Die Spaltenreihenfolge ist hier sehr wichtig. Wird nun eine WHERE-Bedingung ausgeführt, dient die linke Spalte für den Index (siehe Beispiel in Abbildung 1).

Datentypen und Indexspalten

Bei INTEGER-Spalten werden weniger Bytes zum Speichern des Index benötigt. Daher müssen nicht so viele Indexseiten in den Bufferpool geladen werden, wie bei Spalten vom Typ VARCHAR.

Daher sollten Indizes auf Spalten vom Typ INTEGER (4 Bytes Speicherplatz) erstellt werden, statt auf Spalten vom Typ VARCHAR (Anzahl der Charakter + 4 = Anzahl der benötigten Bytes).

Unique Index

Wird bei der Anlage eines Indexes die Option UNIQUE gewählt, so wird die Eindeutigkeit hier nur von den Schlüsselspalten verlangt. Durch das Verwenden der INCLUDE-Klausel bei Unique Indizes werden zusätzliche Spalten zu den Schlüsselspalten angelegt. Die zusätzlichen Spalten vergrößern zwar den Index, es können aber manche Abfragen beschleunigt werden, da die gewünschte

Die Defragmentierung von Tabellen verringert den Speicherplatz und verbessert den Datenzugriff:

REORG TABLE ARTIKEL

Die Verwendung eines temporären Tabellenbereiches mit gleicher Seitengröße:

REORG TABLE ARTIKEL USE MYTEMPSP

Die Sortierung der Zeilen einer Tabelle auf Basis eines bestimmten Index:

REORG TABLE ARTIKEL INDEX IDXID

Im Anschluss sollten die Statistikdaten neu erfasst werden:

RUNSTATS ON TABLE ARTIKEL AND INDEXES ALL

Abb. 3: Die wichtigsten Befehle für die Wartung der Datenbank.

Glossar

OLAP

Online-Analytical-Processing-Datenbanken werden vornehmlich im Datawarehouse-Bereich eingesetzt.

OLTP

Online-Transaction-Processing-Datenbanken, die bei der schnellen Verarbeitung von Transaktionen im Vordergrund steht.

Links

- ▶ [1] IBM DeveloperWorks-Seite:
http://www.ibm.com/developerworks/data/library/dmmag/DMMag_2010_Issue4/DataArchitect/index.html
- ▶ [2] IBM Data Management:
<http://ibmdatamag.com/2010/10/db2-indexes-and-query-performance-part-1/>
<http://ibmdatamag.com/2011/01/db2-indexes-and-query-performance-part-2/>

Bild:

© istockphoto.com | bolide on high-speed road| ssuaphoto
© vectoropenstock | Abstract Color Background|webdesignhot

Information bereits im Index vorhanden ist und somit kein Zugriff auf die Tabellen erfolgen muss. Als Ergebnis der Abfrage in Abbildung 2 sind alle Daten im Index, somit müssen keine weiteren Datenseiten gelesen werden.

Umschreiben von Statements, Vermeiden von Scalarfunktionen

Vermeiden Sie das Verwenden von Scalarfunktionen wie SUBSTR, CHAR, INT in der WHERE-Bedingung

einer Abfrage. Wird eine Scalarfunktion für eine Spalte in einer WHERE-Klausel angewendet, wird der DB2 Optimizer nicht den für die Spalte passenden Index zum Abrufen der Datensätze aus der Tabelle verwenden, sondern einen Tablescan durchführen.

In vielen Fällen ist durch eine Umformulierung der Abfragen eine Performance-Steigerung zu erreichen. Wichtig ist hierbei die Vermeidung von Scalarfunktionen in WHERE-Bedingungen.

Im Folgenden zwei Beispiele:

WHERE substr(Spalte,2,3) = 'ABC'
besser **WHERE spalte like '_ABC%'**

WHERE YEAR(Geburtstag) = 1970
besser **WHERE Geburtstag BETWEEN '1970-01-01' and '1970-12-31'**

Vermeiden Sie die DISTINCT-Klausel beim Select, um Duplikate aus einer Abfrage zu entfernen. Verwenden Sie stattdessen GROUP BY, welches dann auch Indizes verwenden kann. Die DISTINCT-Funktion benötigt viel Zeit, um die Ergebnismenge zu sortieren und ist daher ressourcenintensiv. Bauen Sie Ihre Abfrage so auf, dass die restriktivsten Bedingungen zuerst kommen. Ist eine Bedingung nicht erfüllt, wird sie ausgefiltert und nicht weiter geprüft. Je früher ein Datensatz ausgefiltert wird, desto performanter ist die Abfrage.

Fazit

Eine regelmäßige Wartung der Datenbank, was die Statistiken für den Optimierer und die Reorganisation der Tabellen betrifft (siehe Abbildung 3), ist immens wichtig. Kommt dann noch eine ausgewogene Auswahl der indizierten Spalten und Anzahl der Indizes hinzu, kann eine gute Leistungssteigerung erzielt werden. Dieser Performance-Gewinn erfolgt dann nicht auf Kosten eines höheren CPU-Aufwandes.

Durch eine ausgewogene Auswahl der indizierten Spalten und der Anzahl der Indizes kann eine gute Performance erzielt werden, ohne auf der anderen Seite durch die erhöhten CPU-Kosten diesen Vorteil wieder einzubüßen.



Thomas Schwarz
(info@ordix.de).

Projektmanagement in der Praxis: „Wie aus dem richtigen Leben...“ (Teil V)

Eine Projektkultur schaffen – aber richtig!

Unternehmen steuern ihre strategische und operative Weiterentwicklung zunehmend durch Projekte. Projektstrukturen gewinnen dadurch gegenüber den klassischen Organisationsstrukturen immer mehr an Bedeutung. Die Projektarbeit beschränkt sich dabei nicht mehr nur auf die IT- und Organisationsabteilungen, sondern umfasst inzwischen (fast) alle Bereiche der Unternehmen. Um dieser Entwicklung Rechnung zu tragen, ist es notwendig, eine passende Projektkultur im Unternehmen zu etablieren, die zur vorhandenen Unternehmenskultur passt. Welche Ziele bei der Etablierung einer Projektkultur erreicht werden sollen und welche Maßnahmen für die Einführung notwendig sind, stellt der vorliegende Artikel vor.

Ziele einer Projektkultur

Was soll mit der Einführung einer Projektkultur erreicht werden? Das Hauptziel ist natürlich, dass die Projekte im Unternehmen erfolgreich abgeschlossen werden können. Ein Projekt ist per Definition dann erfolgreich, wenn das geplante Ergebnis

- in der geplanten **Funktionalität**
- in der geplanten **Qualität**
- in der geplanten **Zeit**
- im Rahmen des geplanten **Budgets** und
- im Rahmen der erwarteten **Risiken**

realisiert werden konnte. Insofern geht es in erster Linie darum, Projektkosten zu minimieren, die Termintreue zu verbessern, die Kundenzufriedenheit zu erhöhen, die Projektrisiken so in den Griff zu bekommen, dass es erst gar nicht zu Krisenprojekten kommt.

Weiterhin geht es darum, nicht bei jedem Projekt alles neu zu erfinden. Bewährtes soll wiederverwendet und in der Vergangenheit gemachte Fehler in Zukunft vermieden werden. Die Projektarbeit insgesamt soll professionalisiert, die Zusammenarbeit in Projekten verbessert und der interne Erfahrungsaustausch intensiviert werden.

Elemente einer Projektkultur

Um eine Projektkultur im Unternehmen vollständig zu etablieren, müssen in verschiedenen Bereichen die notwendigen Strukturen und Abläufe geschaffen werden. Die Abbildung 1 zeigt die verschiedenen Elemente einer vollständig etablierten Projektkultur.

PM-Strategie

Zunächst muss eine PM-Strategie entwickelt werden, die festschreibt wann und wie Projekte abgewickelt werden sollen und welche Grundsätze einzuhalten sind. Die PM-Strategie muss natürlich im Einklang mit der vorhandenen Unternehmensstrategie stehen bzw. aus ihr abgeleitet werden. Wichtig ist es auch die bestehende Unternehmenskultur zu berücksichtigen, denn jedes Unternehmen „tickt“ etwas anders.

PM-Organisation

Die definierte PM-Strategie muss anschließend in eine PM-Organisation gegossen werden. Es gibt Unternehmen, die eine eigene Abteilung aufbauen, die als Keimzelle für das Projektmanagement im Hause agiert. Diese Abteilung fungiert zum einen als normierende



Abb. 1: Elemente einer Projektkultur.

Stabstelle, zum anderen als interner Dienstleister für Projektmanagement. Andere Unternehmen etablieren eine rein virtuelle Organisation für das Projektmanagement. Dies bedeutet, dass die PM-Mitarbeiter in ihren Abteilungen organisatorisch verbleiben, aber übergreifend als Know-how-Netzwerk zusammenarbeiten. Die meisten Unternehmen wählen eine Mischung aus beiden Organisationskonzepten.

PM-Prozesse

Abhängig von der gewählten Organisationsform muss nun die Ablauforganisation festgelegt werden, d.h. es müssen Prozesse festgeschrieben werden, wie Projekte beantragt, ausgewählt, genehmigt, aufgesetzt und überwacht werden. Diese PM-Prozesse müssen natürlich zu den Geschäfts- und Entscheidungsprozessen des Unternehmens passen. Neben den Entscheidungs- und Steuerungsprozessen sind die Prozesse für das Assignment Management, Skillmanagement und Knowledge Management für den Bereich Projektmanagement besonders wichtig.

PM-Tools & Methoden

Es folgt der aufwändigste und konzeptionell schwierigste Teil der Etablierung einer Projektkultur – die Auswahl (bzw. die Entwicklung) der zu nutzenden

PM-Tools & Methoden, die den spezifischen Anforderungen des Unternehmens angepasst werden müssen. Da es am Markt eine unüberschaubare Anzahl an Konzepten, Philosophien, Anbietern und Normierungsgremien für Projektmanagement gibt (siehe die kleine Auswahl der weiterführenden Links), ist dieses Vorhaben sehr anspruchsvoll – insbesondere weil üblicherweise viele unterschiedliche Meinungen und Glaubensrichtungen kursieren. Grundsätzlich sollte man die Devise einhalten „Keep it simple“, denn Komplexität entsteht meistens von selbst. Auch muss klar sein, dass man es nicht allen recht machen kann.

Die ausgewählten PM-Werkzeuge, Methoden und Templates sollten in einer zentralen Ablage für alle verfügbar sein (Dateiablage, Intranet oder Wiki). Diese Ablage sollte auch als Plattform, für den generellen Informationsaustausch über Erfahrungen, Aktivitäten und Initiativen aus dem Bereich Projektmanagement dienen.

Einige Unternehmen schreiben die ausgewählten Tools und Methoden verbindlich vor, andere Unternehmen definieren sie nur als ein Angebot, aus dem die einzelnen Projekte auswählen können, was sie für sich benötigen. In den meisten Unternehmen gibt es auch hier eine Mischform aus Vorschriften und Angeboten.

PM-Qualifizierung

Auf Basis der definierten Tools und Methoden muss ein strukturiertes Qualifizierungsprogramm aufgesetzt werden, das die stufenweise Weiterbildung sicherstellt. Es muss eine Vorgehensweise definiert werden, wie die PM-Kompetenz im Hause identifiziert, ausgebaut und gefördert wird. Erfahrene Projektmanager sollten zudem als PM-Coaches für unerfahrene Projektleiter agieren und diese in ihren ersten Projekten begleiten. Auch sollte ein Lessons Learned Workshop [6] am Ende eines jeden Projektes verbindlich vorgeschrieben werden, um einen permanenten Verbesserungsprozess zu etablieren.

PM-Karriere

Um Anreize zu schaffen, dass Mitarbeiter motiviert in Projekten mitwirken bzw. Projekte leiten, muss eine verbindliche PM-Karriere geschaffen werden. Dies bedeutet, dass sich ein Mitarbeiter z.B. vom Projektmitarbeiter, über Projektassistent, Projektleiter, Projektmanager bis zum Senior Projektmanager entwickeln kann. Eine PM-Karriere muss demnach zu einer vergleichbaren Verbindlichkeit, Anerken-

nung und Honorierung führen, wie eine klassische Managementkarriere. Es müssen Kriterien bezüglich Weiterbildung, Kompetenz und Erfahrung definiert werden, die verbindlich festlegen, wann ein Mitarbeiter die nächste Stufe in seiner PM-Karriere erreichen kann. Projektarbeit ist für die Mitarbeiter nur dann dauerhaft attraktiv, wenn sich für sie daraus berufliche Perspektiven und Weiterentwicklungschancen ergeben.

Fazit

Es gibt immer mehr Projekte in den Unternehmen. Gleichzeitig wird die Projektarbeit immer wichtiger und erfolgskritischer. Es lohnt sich also in die Etablierung einer Projektkultur zu investieren. Der entstehende Aufwand wird sich schnell amortisieren. Häufig höre ich aber den Einwand, dass man hierfür keine Zeit habe und dass sich die Projektkultur schon irgendwann von selber einstellen würde.

Mit einer solchen Einstellung kann es Ihnen allerdings leicht passieren, wie dem Fahrradfahrer, der stundenlang neben seinem Fahrrad her rennt. Auf die Frage eines Passanten, warum er das denn mache, antwortet er, dass er es sehr eilig habe und daher keine Zeit hätte auf das Fahrrad zu steigen.



*Benedikt Georgi
(info@ordix.de).*

Glossar

Assignment Management

Die Besetzung der Projekte mit den passenden Mitarbeitern in Abhängigkeit von deren Fähigkeiten, Kompetenzen, Erfahrung und Verfügbarkeit.

Knowledge Management - (Wissensmanagement)

Managementaktivitäten mit dem Ziel, die Wissensbasis im Unternehmen weiterzuentwickeln und diese für alle Mitarbeiter verfügbar zu machen (Steuerung des Wissensaustausches im Unternehmen)

Skillmanagement

Management der Fähigkeiten und Kompetenzen der Mitarbeiter eines Unternehmens, mit dem Ziel deren Fähigkeiten und Kompetenzen weiterzuentwickeln und sie optimal einzusetzen.

Links

- ▶ [1] Gesellschaft für Informatik e.V. / Fachgruppe Projektmanagement:
<http://www.gi.de/>
- ▶ [2] Deutsche Gesellschaft für Projektmanagement e.V.:
<http://www.gpm-ipma.de>
- ▶ [3] PMI - Projekt Management Institute:
<http://www.pmi.org>
- ▶ [4] OGC - Office of Government Commerce in the United Kingdom and other countries - in Deutschland:
<http://www.prince2-deutschland.de>
- ▶ [5] ScrumAlliance:
<http://www.scrumalliance.org>
- ▶ [6] ORDIX news Artikel 4/2011
„Projektmanagement in der Praxis: „Wie aus dem richtigen Leben...“ (Teil III) - Lernen aus Projekten - aber richtig!“:
http://www.ordix.de/ORDIXNews/4_2011/projektmanagement.html



Neue Reihe: Space, the final frontier - die physikalische Speicherverwaltung in Oracle (Teil I)

Tablespaces und schwarze Löcher

In dieser Reihe werden wir zahlreiche Aspekte des physikalischen Speichermanagements von Oracle beleuchten. Dieser Artikel beantwortet einleitend die Frage, wie der Füllgrad von Tablespaces zu berechnen ist, welche Unterschiede es bei den verschiedenen Arten gibt und warum die Trennung von Tabellen und Indizes nicht mehr zeitgemäß ist. Abschließend wird dargestellt, wie viele Tablespaces eine Datenbank überhaupt benötigt.

Tablespaces, Segmente, Extents und Blöcke

Ein Tablespace ist ein Container für Segmente einer Oracle-Datenbank. Ein Segment ist wiederum ein Objekt, welches physikalischen Speicherplatz beansprucht, also z.B. eine Tabelle, ein Index, eine Partition oder ein Cluster. Wie der Name bereits verrät, gehören auch UNDO- und temporäre Segmente hierzu.

Ein Segment besteht aus einem oder mehreren Extents. Ein Extent besteht wiederum aus einer Menge von physikalisch hintereinanderliegenden Blöcken. Auf der einen Seite dienen Extents dazu, die Verkettung zusammengehöriger Blöcke eines Segments zu vereinfachen, auf der anderen Seite ermöglichen sie die Speicherplatzanforderungen von Segmenten auf unterschiedliche physikalische Dateien zu verteilen. So war es schon zu Zeiten von Dateisystemen mit

maximal 2 GB Dateisystemgröße selbstverständlich möglich, Tabellen anzulegen die deutlich größer waren. Da die Blöcke eines Extent physikalisch hintereinanderliegen, muss ein Extent zwangsläufig in genau einer Datendatei liegen.

Ob die Blöcke eines Extent physikalisch auf einem Datenträger hintereinanderliegen, ist aufgrund von zahlreichen Abstraktionsschichten wie RAID und Logical Volumes nicht gewährleistet. Aus Sicht der Datei und des Dateisystems lässt sich das aber vereinfachend annehmen.

Tablespaces und Datendateien

Die Größe eines Tablespace bestimmt sich durch die Größe der ihm zugeordneten Datendateien. Jede Daten-

datei wird mit einer initialen Größe angelegt. Optional hat die Datendatei die Eigenschaft „autoextensible“, mit der sich eine automatische Vergrößerung in definierten Stücken bis hin zu einer maximalen Größe konfigurieren lässt.

Die Zuordnung von Extents zu Datenbankdateien übernimmt Oracle automatisch. Eine manuelle Zuordnung ist zwar möglich, jedoch seit dem Ende des Oracle Parallel Server eher ungebräuchlich. Ob eine Datei in einem herkömmlichen Dateisystem oder in ASM liegt, ist aus Sicht des Speicherplatzverbrauches weitestgehend irrelevant [1].

Datendateien, High Water Marks und Fragmentierung

Datendateien beginnen mit einem Header, der meist 64 KB oder 1 MB groß ist. Dahinter wird der Platz nach und nach mit Extents gefüllt. In jeder Datei zeigt eine High Water Mark an, bis zu welcher höchsten Position in der Datei jemals ein Extent lag. Wird ein Extent oberhalb der High Water Mark angelegt, so wird diese um die Größe des Extent nach oben verschoben. Datendateien können nur bis zum am weitesten entfernten Extent verkleinert werden.

Durch das Löschen oder die Reorganisation von Segmenten entstehen Lücken in der Datendatei, die durch neue Extents gefüllt werden können. Das Resultat wird auch Fragmentierung genannt. Passen Lücken und neue Speicherplatzanforderungen gut zusammen, ist die Fragmentierung i.d.R. unkritisch. Dies wird heute durch standardmäßige Locally Managed Tablespaces erreicht, bei denen alle Extents 8, 128, 1024 usw. Blöcke groß sind. Alle Lücken sowie der Bereich oberhalb der High Water Mark verwaltet die Datenbank als freie Extents.

Der Füllgrad von Tablespaces nimmt natürlich durch viele „Löcher“ ab, aus Sicht des Dateisystems wird der Platz allerdings nicht wieder freigegeben. Beim Backup mit dem Recovery Manager werden diese Löcher mit gesichert, wofür der Recovery Manager nicht die gesamte Datei, sondern nur den Bereich bis zur High Water Mark berücksichtigen muss.

UNDO und temporäre Tablespaces

Im Gegensatz zu permanenten Tablespaces beinhalten Tablespaces vom Typ UNDO und TEMP nur Extents, die zur Laufzeit benötigt werden. Die Extents werden automatisch von der Datenbank allokiert und wieder freigegeben. Aber auch hier fällt die High Water Mark der zugehörigen Dateien nicht zurück. Bei Backups werden temporäre Tablespaces über-

sprungen. Mit ein paar administrativen Mitteln ist es möglich, die Tablespaces im laufenden Betrieb zu wechseln und somit den Platzverbrauch zu reduzieren, wenn diese zur Laufzeit aufgrund eines besonderen Umstands sehr groß geworden sein sollten.

Trennung von Daten und Indizes

Aus Sicht des I/O-Systems gibt es bei Tablespaces nur zwei Formen von lesenden Zugriffen. Dies sind Singleblock-I/Os und Multiblock-I/Os. Während die Ersten meist aus Lesen über ROWID resultieren, entstammen Multiblock-I/Os aus Full-Table-Scans. Bei Multiblock-I/Os sollte die Mehrzahl der betroffenen Extents größer als die maximale I/O-Größe sein, da ein I/O maximal bis zum Ende des Extent geht. Bei einer Extent-Größe von 64 KB und einer maximalen I/O-Größe von 1 MB sind gegenüber einem 1 MB Extent 15 I/O-Operationen zusätzlich notwendig. Weiterhin ist natürlich wichtig, dass die I/O-Aufträge über möglichst viele Platten verteilt werden. Daraus resultiert der Ansatz „stripe and mirror everything“.

```

SELECT  t.tablespace_name
,        t.contents
,        round(( tum.used_space * t.block_size ) / 1024 / 1024, 2 )
        current_mb
,        ts.bytes_alloc / 1024 / 1024 allocated_mb
,        round(( tum.used_space * t.block_size ) /
ts.bytes_alloc * 100, 2 ) allocated_pct_used
,        round( tum.tablespace_size * t.block_size / 1024 / 1024, 2 ) max_mb
,        round( tum.used_percent, 2 ) max_pct_used
FROM    dba_tablespaces t
,        dba_tablespace_usage_metrics tum
,        (
        SELECT  tablespace_name, sum( bytes ) AS bytes_alloc
        FROM    dba_data_files
        GROUP BY tablespace_name
        UNION ALL
        SELECT  tablespace_name, sum( bytes ) AS bytes_alloc
        FROM    dba_temp_files
        GROUP BY tablespace_name
        ) ts
WHERE   ts.tablespace_name = t.tablespace_name
AND     ts.tablespace_name = tum.tablespace_name
ORDER BY t.tablespace_name;

```

Abb. 1: SQL-Kommando zur Ermittlung des Füllgrades eines Tablespace.

TABLESPACE_NAME	CONTENTS	CURRENT_MB	ALLOCATED_MB	ALLOCATED_PCT_USED	MAX_MB	MAX_PCT_USED
SYSAUX	PERMANENT	398,25	600	66,38	4865,89	8,18
SYSTEM	PERMANENT	577,94	700	82,56	4965,89	11,64
TEMP	TEMPORARY	30	59	50,85	4325,38	0,69
T10M	PERMANENT	1	10	10	10	10
UNDOTBS1	UNDO	3,19	425	0,75	4691,38	0,07
USERS	PERMANENT	4,56	7,5	60,83	4273,39	0,11

Abb. 2: Ausgabe der Füllgrade verschiedener Tablespaces.

Glossar

ASM

Automatic Storage Management - Ist ein Logical Volume Manager, der mit der Version 10g Release 1 zunächst nur zur Speicherung von Datenbankdateien eingeführt wurde.

Block

Kleinste Oracle-Speichereinheit. Die Größe beträgt zwischen 2 KB und 32 KB, in der Regel 8 KB. Daten eines Segments werden blockweise in den Shared Memory der Instanz geladen.

Dateien

Datendateien oder temporäre Dateien dienen zur Ablage von Extents. Die Datendatei ist die Schnittstelle zwischen der logischen Speicherstruktur (Tablespace) und der physikalischen Speicherstruktur (Dateisystem).

Extent

Eine Menge mehrerer, physikalisch hintereinanderliegender Blöcke. Ein Segment besteht aus 1-n Extents. Jedes Extent ist dabei exakt einem Segment zugeordnet.

Logical Volumes

Abbildung von physikalischen Festplatten auf Dateisysteme, im Gegensatz zu RAID ohne Datenduplizierung, also ohne Erhöhung der Verfügbarkeit.

RAID

Redundant Array of Independent Disks, Abbildung von physikalischen Festplatten auf logische Festplatten zur Erhöhung der Verfügbarkeit und der Performance.

Segmente

Begriff für alle Oracle-Objekte, die dedizierten Speicherplatz benötigen. Hierzu gehören Tabelle, Indizes, Cluster, Partitionen, temporäre und UNDO-Segmente.

Tablespace

Container zur Ablage von Segmenten. Die Größe eines Tablespace bestimmt sich durch die Größe der ihm zugewiesenen Dateien.

Da heute zwischen dem Dateisystem und den physikalischen Festplatten nahezu immer eine oder mehrere Abstraktionsschichten wie RAID und Logical Volumes liegen, sind die I/O-Charakteristika einzelner Dateien in der Regel gleichartig. Aus diesem Sachverhalt erschließt sich keine vernünftige Begründung, warum heute Daten von Indizes in unterschiedliche Tablespaces getrennt werden sollten [2].

Füllgrad von Tablespaces

Der aktuell maximale Platz eines Tablespace berechnet sich aus der Summe der Bytes der zugehörigen Daten- oder temporären Dateien. Der maximal mögliche Platz berechnet sich, unter Berücksichtigung der Eigenschaft der automatischen Erweiterbarkeit, aus dem freien Platz im Dateisystem und der maximalen Größe der Datendateien. Der belegte Platz berechnet sich aus dem Header der Datenbankdatei und den belegten Extents. Die Abbildung 1 zeigt ein relativ einfaches SQL-Kommando zur Ermittlung des Füllgrades. Die Ergebnisse sind in Abbildung 2 aufgeführt.

Wie viele Tablespaces?

Die folgenden Tablespaces sind für jede produktive Datenbank obligatorisch:

- SYSTEM
- SYSAUX
- TEMP
- UNDO

Darüber hinaus benötigen die Anwendungen auf einer Datenbank lediglich einen Tablespace! Von dieser Regel sollte nur Abstand genommen werden, wenn dafür nachvollziehbare Gründe vorliegen. Denkbar sind dabei:

- sehr viele Read-Only-Objekte, die in ein Read Only Tablespace ausgelagert werden, um die Backup-Zeiten zu reduzieren
- Datenbewirtschaftung mit Transportable Tablespaces
- bei sehr großen Datenmengen, um den Restore einzelner Datenbankdateien zu beschleunigen

Fazit

In der Praxis lässt sich fast immer beobachten, dass die Summe des vorgehaltenen Datenvolumens für das Wachstum von Tablespaces nahezu proportional mit der Anzahl der Tablespaces wächst. Ihr Hersteller von Speichersystemen wird sich darüber freuen. Allerdings befindet sich das schwarze Loch dann schließlich in Ihrem Budget.

Links

- ▶ [1] ORDIX news Artikel 03/2004
„Oracle 10g (Teil II): Tablespaces unter 10g: größer - schneller - flexibler“:
http://www.ordix.de/ORDIXNews/3_2004/tablespaces_10g.html
- ▶ [2] Richard Foote's Oracle Blog:
<http://richardfoote.wordpress.com/2008/04/16/separate-indexes-from-tables-some-thoughts-part-i-everything-in-its-right-place/>

Bild: © flickr.com | Lego!!! | Glenn Calvin



Martin Hoermann
(info@ordix.de).

Neue Reihe: Solaris 11 - Neue Funktionen (Teil I)

Boot-Umgebungen

Für das Erscheinen des Betriebssystems Solaris 11 wählte Oracle ein sehr passendes Datum: November 2011. Zusätzlich zum einprägsamen Datum wurde die neue Version von Solaris mit zahlreichen Funktionen ausgestattet, welche die Administratoren vor neue Herausforderungen stellen. Ähnlich wie bei Solaris 10 wurden Teile des Betriebssystems dermaßen verändert, dass es schon fast einem Neueinstieg gleich kommt. In diesem Artikel möchten wir über die Solaris 11 Boot-Umgebungen berichten, welche den Nachfolger der Live-Upgrade-Funktion älterer Versionen darstellen.

Boot-Umgebungen in Solaris 11

Eine Boot-Umgebung in Oracle Solaris 11 stellt eine vollständig boot-fähige Instanz eines Betriebssystemabbildes mit verschiedenen Patch-Leveln oder unterschiedlicher Anwendungssoftware dar.

Der Administrator besitzt hierbei die Möglichkeit, mehrere Boot-Umgebungen auf einem System zu verwalten und bei Bedarf zu aktivieren. Bei der Neuinstallation eines Oracle Solaris 11 Release wird auf dem System automatisch eine Boot-Umgebung mit dem Namen „solaris“ erstellt.

Mit einem einzigen Kommandozeilenwerkzeug namens `beadm` können zusätzliche Boot-Umgebungen auf dem System eingerichtet und verwaltet werden. Das Werkzeug `beadm` nutzt hierbei die Funktionalität der ZFS-Snapshots. UFS-Dateisysteme können bei der Erstellung von Boot-Umgebungen nicht berücksichtigt werden.

Ein Kommando zur Administration

Für die Administration und Verwaltung von Boot-Umgebungen wird das Kommando `beadm` verwendet, welches über untergeordnete Argumente und Optionen beeinflusst werden kann.

Die Abbildung 1 zeigt das Kommando `beadm` mit seinen Argumenten. Für die Erstellung einer Boot-Umgebung fasst `beadm` alle Datensätze zu einer Umgebung zusammen und führt an ihr gleichzeitig Aktionen aus. Der Begriff „Datensatz“ ist hierbei eine

allgemeine Bezeichnung für ZFS-Objekte wie Clones, Dateisysteme oder Snapshots. Im Kontext der Boot-Umgebungen bezieht sich der Begriff Datensatz speziell auf die Dateisystemspezifikationen für eine bestimmte Boot-Umgebung bzw. einen bestimmten Snapshot.

Bei der Verwendung von `beadm` sind keine ZFS-Befehle mehr notwendig, um einzelne Datensätze zu verändern. Das Programm verwaltet und aktualisiert das GRUB-Menü (bei x86-Systemen) oder das Boot-Menü (bei SPARC-Systemen) automatisch. Wird mit dem Programm eine neue Boot-Umgebung erstellt, wird diese automatisch dem GRUB-Menü bzw. Boot-Menü hinzugefügt. `beadm` beherrscht aktuell folgende Möglichkeiten:

- Anzeigen von Boot-Umgebungen, Clones und Snapshots
- Erstellen einer auf der aktiven Boot-Umgebung basierenden neuen Boot-Umgebung
- Erstellen einer auf einer inaktiven Boot-Umgebung basierenden neuen Boot-Umgebung
- Erstellen eines Snapshot von einer vorhandenen Boot-Umgebung
- Erstellen einer neuen Boot-Umgebung auf der Basis eines vorhandenen Snapshot
- Erstellen einer neuen Boot-Umgebung und Angabe eines spezifischen Namens für das GRUB-Menü bzw. SPARC-Boot-Menü
- Aktivieren einer inaktiven Boot-Umgebung


```
Usage:
  beadm subcommand cmd_options
subcommands:
  beadm activate beName
  beadm create [-a] [-d description]
                [-e non-activeBeName | beName@snapshot]
                [-o property=value] ... [-p zpool] beName
  beadm create beName@snapshot
  beadm destroy [-fF] beName | beName@snapshot
  beadm list [[-a] | [-d] [-s]] [-H] [beName]
  beadm mount beName mountpoint
  beadm rename beName newBeName
  beadm unmount [-f] beName
```

Abb. 1: Das Kommando `beadm` mit seinen möglichen Argumenten.

```
# beadm create -d "Zweite Boot-Umgebung" Solaris11-backupBE
```

Abb. 2: Erstellen einer neuen Boot-Umgebung aus der aktuell aktiven.

```
# beadm list
BE          Active Mountpoint Space  Policy Created
--          -
KopieVonBackupBE -      -      55.0K  static 2012-02-01 19:39
Solaris11-backupBE R      -      4.58G  static 2012-02-01 19:39
solaris     N      /      48.24M static 2011-12-18 08:39
```

Abb. 3: Anzeige der vorhandenen Boot-Umgebungen.

- Mount einer Boot-Umgebung
- Unmount einer Boot-Umgebung
- Entfernen einer Boot-Umgebung
- Löschen eines Snapshot, der mit einer Boot-Umgebung verknüpft ist
- Umbenennen einer inaktiven Boot-Umgebung

Erstellen einer Boot-Umgebung

Sollen Änderungen an einem Oracle Solaris 11 Betriebssystem durchgeführt werden, beispielsweise Änderungen an Dateien oder das Einspielen neuer Pakete, kann mit `beadm create` eine Kopie der aktuellen Umgebung erstellt werden.

Der Administrator hat nun die Wahl, seine Änderungen in der aktuellen Boot-Umgebung oder der neu erstellten durchzuführen. Bei eventuellen Problemen kann nun relativ einfach durch Wechseln der Boot-

Umgebung der ursprüngliche Zustand wieder hergestellt werden.

Wird mit `beadm` eine neue Boot-Umgebung erstellt, wird automatisch im GRUB-Menü bzw. im Boot-Menü ein Eintrag erstellt. Mit der Option `-d` kann Einfluss auf den Titel des Boot-Eintrages genommen werden. Der in Abbildung 2 aufgezeigte Befehl erstellt eine neue Boot-Umgebung aus der augenblicklich aktiven und gibt dieser einen Namen.

Boot-Umgebungen aus inaktiven Umgebungen, den sogenannten Clones, können mit `beadm create -e` erstellt werden. Soll der aktuelle Zustand gesichert werden, besteht zudem die Möglichkeit, einen Snapshot der aktuellen Umgebung zu erstellen. Ein solcher stellt hier ein schreibgeschütztes Abbild eines Datensatzes oder einer Boot-Umgebung zu einem bestimmten Zeitpunkt dar. Snapshots an sich sind nicht boot-fähig. Allerdings kann aus einem Snapshot wiederum eine neue Boot-Umgebung erstellt werden, die dann boot-fähig ist.

Anzeigen, was es so gibt

Alle mit dem Kommandozeilenbefehl `beadm` erstellten Snapshots, Boot-Umgebungen und Datensätze können mit dem Befehl `beadm list` angezeigt werden. Im Normalfall werden Snapshots und Boot-Umgebungen mit dem Befehl `beadm create` erstellt.

Es ist aber trotzdem möglich, Snapshots und Boot-Umgebungen mit anderen Dienstprogrammen zu erstellen. Beispielsweise werden bei der Benutzung des Befehls `pkg` automatisch Clones einer Boot-Umgebung erstellt, wenn Pakete installiert oder vorhandene aktualisiert werden. Der Befehl `beadm list` zeigt zudem Boot-Umgebungen an, die mit dem Befehl `pkg` erstellt wurden. Abbildung 3 zeigt die Ausgabe von `beadm list`.

Aktivieren von Boot-Umgebungen

Soll beim nächsten Neustart eine andere Boot-Umgebung gebootet werden, muss diese zunächst mit `beadm activate` aktiviert werden. Dabei kann immer nur eine Boot-Umgebung aktiv sein. Die neu aktivierte Boot-Umgebung wird beim Systemneustart zur Standard-Boot-Umgebung. Diese stellt jetzt die Standardeinstellung im GRUB-Menü bzw. im SPARC-Boot-Menü dar.

Wenn das starten mit der Boot-Umgebung fehlschlägt, kann das System neu gestartet und manuell eine andere Umgebung aus dem Boot-Menü ausge-

wählt werden. Dabei lassen sich aber nur Boot-Umgebungen und Clones aktivieren. Wird versucht einen Snapshot zu aktivieren, schlägt der Vorgang fehl.

Fazit

Die Boot-Umgebungen in Oracle Solaris 11 stellen eine wesentliche Verbesserung des Live-Upgrade-Mechanismus früherer Versionen dar. Mit nur einem Befehl kann die Administration durchgeführt werden.

Durch die Nutzung der Funktionen des ZFS-Dateisystems lassen sich platzsparend Boot-Umgebungen anlegen. Einzig und allein bei der Installation von Software und Updates sollte der Administrator hin und wieder einen Blick auf automatisch erstellte Boot-Umgebungen werfen, da sonst irgendwann unzählige Umgebungen existieren und der Plattenplatz ausgeht.

In unserem neuen Seminar „Solaris 11 Administration Neuheiten“ erfahren Sie weitere interessante Details zum Thema Boot-Umgebungen.



Christian Fertsch
(info@ordix.de).

Glossar

Journaling

Unter Journaling versteht man die Aufzeichnung des Dateisystems mit allen Änderungen in einem Logfile. Die Einträge erfolgen vor Änderung des Dateisystems, wodurch die Konsistenz eines Dateisystems nach einem Systemcrash durch Auslesen des Logfile wieder hergestellt werden kann.

Snapshot

Abbild eines Dateisystems zu einem bestimmten Zeitpunkt. Kann nur lesend verwendet werden.

UFS-Dateisysteme

Bisheriges Standarddateisystem unter Solaris vor Solaris 10. Solaris bietet ab der Version 7 eine Option an, die ein Journaling in UFS (Unix File System) implementiert.

ZFS-Dateisysteme

Zeta File System - Ein von Sun Microsystems entwickeltes transaktionales Dateisystem, welches zahlreiche Erweiterungen für die Verwendung im Server- und Rechenzentrumsbereich enthält.

ZFS-Objekte

Komponenten die im ZFS-Umfeld anzutreffen sind, wie ZFS-Clones, Dateisysteme oder Snapshots.

ZFS-Snapshots

Ein Snapshot stellt ein Abbild eines ZFS-Dateisystems zu einem bestimmten Zeitpunkt dar.

Seminarempfehlung: Solaris 11 Administration Neuheiten

► **Informationen/Online-Anmeldung:** <http://training.ordix.de/siteengine/action/load/kategorie/Betriebssysteme/nr/1111/index.html>

Nach vielen Jahren wurde Ende 2011 die neue Version von Solaris 11 veröffentlicht. In diesem Seminar werden wir die Neuerungen und Änderungen bei der Administration aufzeigen und durch Übungen vertiefen.

Seminarinhalte

- Neuerung bei der Installation: AI
- Neue Befehle und Tools bei der Netzwerkadministration
- Crossbow: virtuelle Netzwerke
- Softwaremanagement jetzt anders: IPS
- Erstellung von alternativen Boot-Umgebungen
- Neuigkeiten bei ZFS und Zonen
- Comstar: Feature für den Einsatz als Filer
- Kooperation mit Windows
- Vertiefung der Theorie durch praktische Übungen und Beispiele

Termine

20.08.2012 - 22.08.2012 in Wiesbaden
08.10.2012 - 10.10.2012 in Wiesbaden
03.12.2012 - 05.12.2012 in Wiesbaden

Seminar-ID: BS-24

Dauer: 3 Tage

Preis pro Teilnehmer: 1.290,00 € (zzgl. MwSt.)

Frühbucherpreis: 1.161,00 € (zzgl. MwSt.)

SubSelect oder BatchSize?

Das ist hier die Frage

Im ersten Teil dieser Reihe [4] haben wir die Möglichkeit betrachtet, das „N+1-SELECT-Problem“ mittels der Hibernate Batch-Fetching-Strategie in den Griff zu bekommen. Mit der SubSelect-Fetching-Strategie steht dem Entwickler eine zweite Möglichkeit zur Verfügung, um die Performance seiner Applikation zu verbessern. Doch wann sollte man die eine Möglichkeit verwenden und wann die andere? Die Antwort auf diese Frage geben wir im folgenden Artikel.

Kurzer Rückblick

Das N+1-SELECT-Problem ergibt sich immer dann, wenn mehrere Objekte in der Applikation verarbeitet werden, die in einer 1:n- oder n:m-Relation zu weiteren Objekten stehen, die ebenfalls in der Verarbeitung berücksichtigt werden müssen. Bei einem nicht optimierten Mapping generiert Hibernate N+1-SELECT-Anweisungen.

Wir haben dieses Problem an dem Beispiel aus Abbildung 1 untersucht und dabei gezeigt, dass durch die Angabe einer BatchSize die N+1-SELECT-Befehle auf N/BatchSize+1 reduziert werden und welche zusätzlichen Probleme dabei auftreten können.

Und nun mit SubSelect

Wie bei der Batch-Fetching-Strategie werden bei der SubSelect-Fetching-Strategie ebenfalls referenzierte Objekte mit einem IN-Statement eingeladen. Allerdings enthält dieses Statement keine Schlüsselliste sondern eine Unterabfrage.

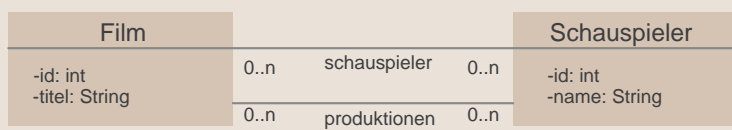


Abb. 1: Das Objektmodell, dem die Beispiele zu Grunde liegen.

Um diese Fetching-Strategie bei der Schauspieler-Relation in unserem Beispiel zu verwenden, muss zunächst das Mapping angepasst werden (siehe Abbildung 2). Doch egal, ob das Mapping mit BatchSize oder mit SubSelect annotiert wird, es ergeben sich nahezu die gleichen Ausgaben (vergleiche Abbildung 3 mit Abbildung 6 des vorherigen Artikels). Sie unterscheiden sich nur in folgendem Statement:

Mit BatchSize:

```
select
  s.*
from
  join_film_schauspieler jfs,
  schauspieler s
where
  jfs.fk_schauspieler_id=s.id and
  jfs.fk_film_id in (
    1, 6, 10, 12, 3
  )
```

Mit SubSelect:

```
select
  s.*
from
  join_film_schauspieler jfs,
  schauspieler s
where
  jfs.fk_schauspieler_id=s.id and
  jfs.fk_film_id in (
    select
      f1.id
    from
      join_film_schauspieler jfs1,
      film f1
    where
      jfs1.fk_film_id =f1.id and
      jfs1.fk_schauspieler_id=2
  )
```


Anstelle der Auflistung aller benötigten IDs wird eine Unterabfrage von Hibernate generiert. Diese Unterabfrage entspricht dem Statement, mit dem die Filmobjekte eingeladen wurden (oder allgemeiner formuliert: mit dem die Parent-Objekte der Relation eingeladen wurden).

Komplexität der Unterabfragen

Hibernate verwendet, wie bereits erwähnt, für die Unterabfrage exakt das Statement, mit dem die Parent-Objekte eingeladen wurden. Wenn das ursprüngliche Statement bereits komplex war, dann ist die Unterabfrage umso komplexer.

Werden in unserer Beispielapplikation alle Filme ausgegeben, bei denen ein Schauspieler mitwirkt, der den Film auch produziert, so führt dies beim Zugriff auf die Film-Schauspieler-Relation zu dem SELECT aus Abbildung 4. Wird anschließend die Schauspieler-Produktionen-Relation verwendet, die ebenfalls mit SubSelect annotiert ist, so werden (Sie ahnen es sicher schon) beide Anweisungen kombiniert zu dem Statement aus Abbildung 5.

Jede weitere SubSelect-annotierte Relation verlängert dieses Statement. Bei komplexen Relationsgeflechten wird so recht schnell die Datenbanklast eher erhöht als verringert, was ja das ursprüngliche Ziel war.

Die Komplexität der generierten SELECT-Befehle ist meiner Ansicht nach der größte Nachteil dieser Technik. Theoretisch könnte das Datenbanksystem zwar die Daten, welche die Unterabfrage selektiert, noch im Cache haben, so dass in der Praxis die Laufzeiten nicht allzu schlecht werden. Doch das wäre eher dem Zufall überlassen.

BatchSize vs. SubSelect

Durch die Verwendung der `subselect`-Annotation werden die ehemals N+1-Statements auf zwei reduziert. Diese Vorgehensweise hat sogar den Vorteil, dass bei unglücklichen Konstellationen nicht zu viel Speicher verbraucht wird, so wie es bei der BatchSize-Technik leicht passieren kann (dies gilt zumindest für die hier betrachteten Anwendungsfälle).

Ein weiterer Vorteil der SubSelect-Fetching-Strategie ist die konstante Anzahl der Statements, während bei der BatchSize-Variante die Anzahl der Statements von der BatchSize und der Menge der Daten abhängt. In unserem Beispiel würden bei einer BatchSize von 30 und einem Schauspieler, der in 121 Filme mitgewirkt hat, 5 Statements an die Datenbank geschickt. Wohin-

```
@ManyToMany(cascade = CascadeType.ALL)
@JoinTable(name = "JOIN_FILM_SCHAUSPIELER", ...)
@Fetch(FetchMode.SUBSELECT)
private Collection<Schauspieler> schauspieler;
```

Abb. 2: Das Mapping der Schauspieler-Relation.

```
Tom Hanks (2)

Produktionen
select f.* from join_schauspieler_produktionen jsp, film f
where jsp.fk_film_id=f.id and jsp.fk_schauspieler_id=2

    The Pacific (3)
    Band of Brothers (6)

Mitwirkungen
select f.* from join_film_schauspieler jfs, film f where jfs.
fk_film_id=f.id and jfs.fk_schauspieler_id=2

    Apollo 13 (1)

select s.* from join_film_schauspieler jfs, schauspieler s
where jfs.fk_schauspieler_id=s.id and jfs.fk_film_id in (se-
lect f1.id from join_film_schauspieler jfs1, film f1 where
jfs1.fk_film_id=f1.id and jfs1.fk_schauspieler_id=2)

    Tom Hanks (2)
    Kevin Bacon (9)
    Illuminati (10)
    Tom Hanks (2)
    Ewan McGregor (11)
    Sakrileg (12)
    Tom Hanks (2)
```

Abb. 3: Die Ausgaben mit SubSelect.

```
select
s.*
from
    join_film_schauspieler jfs,
    schauspieler s
where
    jfs.fk_schauspieler_id=s.id and
    s.fk_film_id in (
        select
            f1.id
        from
            film f1,
            join_film_schauspieler jfs1,
            schauspieler s1,
            join_schauspieler_produktionen jsp,
            film f2
        where
            f1.id=jfs1.fk_film_id and
            jfs1.fk_schauspieler_id=s1.id and
            s1.id=jsp.fk_schauspieler_id and
            jsp.fk_film_id=f2.id and
            f1.id=f2.id
    )
```

Abb. 4: Komplexeres Statement mit Unterabfrage.

```

select
  f1.*
from
  join_schauspieler_produktionen jsp1,
  film f1
where
  jsp1.fk_film_id=f1.id and
  jsp1.fk_schauspieler_id in (
    select
      s1.id
    from
      join_film_schauspieler jsf1,
      schauspieler s1
    where
      jsf1.fk_schauspieler_id=s1.id and
      jsf1.fk_film_id in (
        select
          f2.id
        from
          film f2,
          join_film_schauspieler jfs2,
          schauspieler s2,
          join_schauspieler_produktionen jsp2,
          film f3
        where
          f2.id=jfs2.fk_film_id and
          jfs2.fk_schauspieler_id=s2.id and
          s2.id=jsp2.fk_schauspieler_id and
          jsp2.fk_film_id=f3.id and
          f2.id=f3.id
      )
    )
)

```

Abb. 5: Komplexeres Statement mit zwei Unterabfrage.

Links

- ▶ [1] ORDIX news Artikel 03/2008 "First-/Second Level und Query Cache - schneller als die Polizei erlaubt ...":
http://www.ordix.de/ORDIXNews/3_2008/Java_J2EE_JEE/Performance_steigern_Query_Cache.html
- ▶ [2] ORDIX news Artikel 02/2009 „Easy Hibernate Cache - mehr PS oder mehr Sicherheit?“:
http://www.ordix.de/ORDIXNews/2_2009/Java_JEE/easy_hibernate_cache.html
- ▶ [3] ORDIX news Artikel 03/2009 „Lazy und Eager Loading - ein guter Cache lädt nicht mehr als er muss“:
http://www.ordix.de/ORDIXNews/3_2009/Java_JEE/hibernate_lazy_eager_loading.html
- ▶ [4] ORDIX news Artikel 01/2012 „Hibernate für Fortgeschrittene (Teil I) - Die Batch-Fetching-Strategie für Realationen richtig anwenden“:
http://www.ordix.de/ORDIXNews/1_2012/hibernate_batch_fetching_strategie.html
- ▶ [5] Leseempfehlung:
Java Persistence mit Hibernate von Gaving King, Christian Bauer, Jürgen Dubau



Andreas Maßmann
(info@ordix.de).

gegen bei der Verwendung von SubSelect im gleichen Beispiel immer nur ein Statement benötigt wird, um die Daten der Relation einzuladen.

Dem gegenüber steht die höhere Komplexität der erzeugten Statements. Die Datenbanksysteme können in der Regel Daten, die über den Primärschlüssel selektiert werden, sehr schnell laden. Dies gilt natürlich auch für Listen von Primärschlüsseln, so wie sie durch die Angabe einer BatchSize von Hibernate generiert werden. Wobei zu beachten ist, dass die Laufzeiten solcher Statements natürlich abhängig von der Länge der Listen sind.

Die Kosten von Statements mit Unterabfragen hängen von den Suchbedingungen ab, je komplexer die Suchkriterien, desto länger die Laufzeiten. Ob diese Komplexität bei der SubSelect-Fetching-Strategie zu einer schlechteren Performance als bei der Batch-Fetching-Strategie führt, müssten Tests und Laufzeitmessungen in den einzelnen Anwendungsfällen zeigen.

Fazit

Dieser Artikel hat gezeigt, dass das „N+1-SELECT-Problem“ auch durch die Verwendung der SubSelect-Fetching-Strategie gelöst werden kann. Weiterhin garantiert diese Technik, dass maximal 2 Statements nötig sind, um die Parent- und alle Child-Objekte einzuladen.

Leider erhöht sich die Komplexität der SQL-Anweisungen, insbesondere dann, wenn von den Child-Objekten weiter navigiert wird, da dann die Unterabfragen geschachtelt werden.

Wie in den meisten Fällen, kann keine klare Empfehlung à la „verwende nur die Batch-Fetching-Strategie“ ausgesprochen werden, da die Probleme in der Praxis vielfältig sind. Aus diesem Grund bietet Hibernate auch beide Möglichkeiten an.

In einem aktuellen Projekt verwenden wir ausschließlich die Batch-Fetching-Strategie und haben damit sehr gute Resultate erzielt. In anderen Projekten muss dies aber nicht so sein. Vielleicht ist es in bestimmten Situationen sogar ratsam, beide Techniken zu kombinieren, um eine optimale Performanz zu erzielen. Der Artikel hat auf mögliche Gefahren bzw. Konsequenzen der verschiedenen Techniken hingewiesen. Ob diese bei Ihnen in der Praxis auftreten können, müssen Tests und Analysen zeigen.

Active Data Guard - Praxisbericht

Aufbau einer Lesefarm mit Active Data Guard

Mit der Option Active Data Guard kann ab Oracle 11g eine Standby-Datenbank „Read Only“ geöffnet werden, trotzdem wird sie von der Primary-Datenbank aktualisiert. Mit Oracle Active Data Guard können somit Standby-Datenbanken als Lesedatenbanken oder ganze Lesefarmen für Abfragen, Berichte und webbasierte Zugriffe konfiguriert werden. Der Aufbau von Lesefarmen ist für Kunden geeignet, die eine hohe Leselast haben, die diese von der Primary-Datenbank nehmen wollen und dennoch aktuelle Daten benötigen. Zusätzlich kann mit einer Fast-Start-Failover-Konfiguration ein Ausfall der Primary-Datenbank mit einer Standby-Datenbank abgesichert werden.

Realisierte Lesefarm mit Active Data Guard unter RedHat 5.7 64bit mit Oracle 11.2.0.3

In einem über 9 Monate laufenden Projekt wurde eine Active Data Guard - Lesefarm mit 10 Standby-Datenbanken konzipiert und realisiert. Dieser Bericht beschreibt das grundsätzliche Konzept und den Aufbau der Active Data Guard - Lesefarm.

Die Oracle Enterprise Edition ist die Voraussetzung für den Einsatz von Data Guard. Mit Active Data Guard kann eine Standby-Datenbank „Read Only“ geöffnet werden und wird dennoch von der Primary-Datenbank aktualisiert. Für den Einsatz von Active Data Guard muss keine zusätzliche Software installiert werden.

Ab Oracle 11.2. kann in einer Data-Guard-Konfiguration mit bis zu 30 Standby-Datenbanken gearbeitet werden. Alle Datenbanken innerhalb der Data-Guard-Konfiguration müssen dabei lizenziert werden. Die Option „Active“ muss extra lizenziert werden. Die Abbildung 1 zeigt die im Projekt realisierte Active-Data-Guard-Lesefarm.

Die Primary-Datenbank befindet sich im primären Rechenzentrum und versorgt die Lesedatenbanken mit aktuellen Daten. Zudem wird die K-Fall-Standby-Datenbank und werden die Lesedatenbanken im K-Fall-Rechenzentrum von der Primary-Datenbank mit Daten versorgt.

In einer Fast-Start-Failover-Konfiguration werden die Primary- und die K-Fall-Standby-Datenbank mit dem Observer überwacht. Bei einem Katastrophenfehler der Primary-Datenbank sorgt der Observer für einen automatischen Failover auf die K-Fall-Standby-Datenbank und macht sie zur Primary-Datenbank. Die Client-Anbindung, z.B. über Webserver, erfolgt über konfigurierte Loadbalancer-Pools.

Konfiguration Data Guard

Für die einfache Administration einer so komplexen Lösung mit Data Guard ist der Einsatz des Data Guard Broker unerlässlich. Mit dem Data Guard Broker kann die Konfiguration zentral administriert und überwacht werden. Die Administration kann im Line-Mode (`dgmgr1`) oder im Oracle Enterprise Manager mit dem Data Guard Manager erfolgen. Mit dem Befehl `show configuration` werden alle Datenbanken der Data-Guard-Konfiguration aufgeführt (siehe Abbildung 2).

Transportmodis

Bei der Konfiguration der Data-Guard-Umgebung kann zwischen zwei Transportmodi für die Übertragung von Redo-Log-Einträgen zwischen Primary- und Standby-Datenbank unterschieden werden.

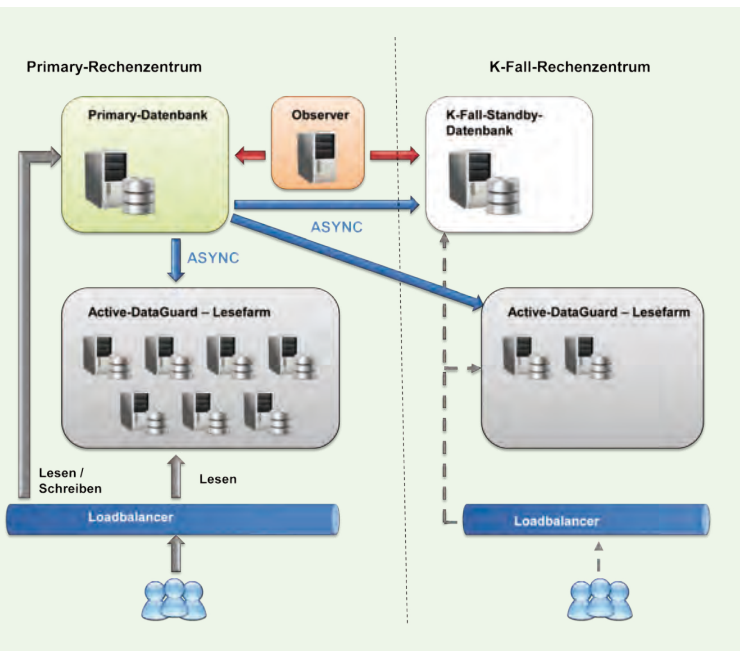


Abb. 1: Aufbau der Active-Data-Guard-Lesefarm.

```
DGMGRL> show configuration;
```

```
Configuration - DG_PROD
```

```
Protection Mode: MaxPerformance
Databases:
DB1_ST1      - Primary database
DB2_ST2      - (*) Physical standby database
DB3_ST1      - Physical standby database
DB4_ST1      - Physical standby database
DB5_ST1      - Physical standby database
DB6_ST1      - Physical standby database
DB7_ST1      - Physical standby database
DB8_ST1      - Physical standby database
DB9_ST1      - Physical standby database
DB10_ST2     - Physical standby database
DB11_ST2     - Physical standby database
```

```
Fast-Start Failover: ENABLED
```

```
Configuration Status:
SUCCESS
```

Abb. 2: Anzeige aller Datenbanken der Data-Guard-Konfiguration.

ASYNC

Bei dem Transportmodus ASYNC wird eine Transaktion (Übertragung von Redo-Log-Einträgen) verzögert durch die Standby-Datenbank an die Primary-Datenbank bestätigt. Der ASYNC-Modus ist die Default-Einstellung. Die Standby-Datenbank ist nicht absolut synchron mit der Primary-Datenbank. Daher wird es einen Datenverlust beim Crash der Primary-Datenbank und einem automatischen Failover auf die

Standby-Datenbank geben, auch wenn dieser nur minimal ausfällt. Im Wesentlichen ist der Datenverlust abhängig von der Netzwerklatenz zwischen der Primary- und Standby-Datenbank.

SYNC

Bei dem Transportmodus SYNC wird erst nach der Übertragung bzw. dem Anwenden der Redo-Log-Einträge auf die Standby-Datenbank die Transaktion abgeschlossen. Die Primary-Datenbank wartet somit auf die Bestätigung der Standby-Datenbank.

Beim Transportmodus SYNC wird die Performance der Primary-Datenbank eingeschränkt, insbesondere dann, wenn die Netzverbindung zwischen Primary- und Standby-Datenbank unterbrochen oder die Netzwerklatenz schlechter wird.

Ab Oracle 11.2 wird die Übertragung bzw. das Schreiben der Redo-Log-Einträge in die Primary- und Standby-Datenbanken parallelisiert. Dies gilt sowohl bei SYNC als auch bei ASYNC. Das Hinterherlaufen der Standby-Datenbanken im ASYNC-Modus ist daher hauptsächlich von der Transaktionslast auf der Primary-Datenbank und der Netzwerklatenz zwischen Primary- und Standby-Datenbank abhängig. Abbildung 3 zeigt, wie die Transport- und Apply-Lücke einer Standby-Datenbank mit dem Befehl `dgmgrl` überprüft werden kann.

Wenn die Primary-Datenbank nicht performance-mäßig durch Data Guard eingeschränkt werden soll, ist die Wahl des ASYNC-Modus zwischen der Primary- und den Standby-Datenbanken empfehlenswert. Ein Datenverlust muss allerdings bei einem automatischen Failover in Kauf genommen werden. Beide Transportmodi können aber auch kombiniert werden, z.B.:

- ASYNC zwischen der Primary- und den Lesedatenbanken. Zwischen der Primary-Datenbank und den Lesedatenbanken ist meistens eine kurze Asynchronität tolerierbar.
- SYNC zwischen der Primary- und der K-Fall-Standby-Datenbank. Diese Variante ist zu wählen, wenn im Failover-Fall kein Datenverlust für die neue Primary-Datenbank toleriert werden kann.

Fast Start Failover (Observer)

Der Observer wurde mit Oracle 10.2 eingeführt. Er überwacht auf einem separaten Server die Primary- und eine definierte Standby-Datenbank. Dabei sorgt er nach einem Katastrophenfehler auf der Primary-Datenbank für einen automatischen Failover auf die Standby-Datenbank. Optimalerweise sollte der

Observer an einem dritten Standort stehen, damit er auch einen Failover nach einem Komplettausfall des Primary-Rechenzentrums durchführen kann. Der Observer hat in unserer Konfiguration folgende Vorteile:

- Bei einem Katastrophenfehler auf der Primary-Datenbank führt der Observer einen automatischen Failover auf die Standby-Datenbank durch. Die neue Primary-Datenbank versorgt nun die Lesedatenbanken mit Daten.
- Sollten bei einem Katastrophenfehler der Primary-Datenbank eine oder mehrere Lesedatenbanken weiter sein (höhere SCN, z.B. bei der Verwendung vom Transportmodus ASYNC), als die K-Fall-Standby-Datenbank, werden vom Observer automatisch alle betroffenen Lesedatenbanken auf den Stand der neuen Primary-Datenbank zurückgesetzt.

Zwei wichtige Parameter in der Fast-Start-Failover-Konfiguration sind die Parameter `Threshold` und `LagLimit`.

`Threshold` bestimmt, wann nach einem Störungseintritt auf der Primary-Datenbank ein automatischer Failover auf die definierte Standby-Datenbank durch den Observer initiiert wird. `Threshold 600` bedeutet, dass nach einem Störungseintritt auf der Primary-Datenbank 600 Sekunden vergehen, bis der Observer einen Failover initiiert. Der Parameter kann über folgenden Befehl geändert werden:

```
dgmgrl> edit configuration set property
FastStartFailoverThreshold = 600;
```

Mit dem Parameter `LagLimit` wird der mögliche, tolerierte Datenverlust bei einem Failover durch den Observer auf die Standby-Datenbank angegeben. `LagLimit 600` (10 Minuten) bedeutet, dass der Observer einen Failover auf die Standby-Datenbank nur dann durchführt, wenn Primary- und Standby-Datenbank kleiner oder gleich 10 Minuten asynchron sind.

Ist das `LagLimit` zwischen Primary- und Standby-Datenbank größer als 10 Minuten wird durch den Observer kein automatischer Failover durchgeführt. Geändert werden kann der Parameter mit `dgmgrl`:

```
dgmgrl> edit configuration set property
FastStartFailoverLagLimit = 600;
```

Mit dem Befehl `show fast_start failover` im `dgmgrl` kann die Fast-Start-Failover-Konfiguration angezeigt werden (siehe Abbildung 4).

Flashback Database

Mit „Flashback Database“ können Datenbanken schnell auf einen Stand aus der Vergangenheit gesetzt

```
DGMGR> show database 'DB2_ST3';

Database - DB2_ST3

Role:                PHYSICAL STANDBY
Intended State:      APPLY-ON
Transport Lag:       0 seconds
Apply Lag:           0 seconds
Real Time Query:    ON
Instance(s) :       DBPROD

Database Status:
SUCCESS
```

Abb. 3: Überprüfen der Standby-Datenbank.

```
DGMGR> show fast_start failover
Fast-Start Failover: ENABLED

Threshold:           600 seconds
Target:              DB2_ST2
Observer:            observ001
Lag Limit:           600 seconds
Shutdown Primary:   TRUE
Auto-reinstate:     TRUE

Configurable Failover Conditions
Health Conditions:
Corrupted Controlfile  YES
Corrupted Dictionary  YES
Inaccessible Logfile   NO
Stuck Archiver         NO
Datafile Offline      YES

Oracle Error Conditions:
(none)
```

Abb. 4: Fast-Start-Failover-Konfiguration.

werden. Für den Einsatz des Observer ist die Konfiguration von Flashback Database Voraussetzung. Für die Lesedatenbanken sollte der Flashback-Zeitraum auf mindestens 2 Stunden eingestellt werden. Der Flashback-Zeitraum wird durch den SPFILE-Parameter `db_flashback_retention_target` bestimmt (Angabe in Minuten).

Anbindung der Clients

Die Anbindung der Clients, z.B. der Webserver, kann über einen Loadbalancer erfolgen. In einem Loadbalancer-Pool können die Lesedatenbanken nach Aufgaben gruppiert werden, zum Beispiel Batch-Verarbeitung oder normale Lesezugriffe der Clients.

Der Loadbalancer übernimmt die Aufteilung der Client-Anfragen auf die Lesedatenbanken. Diese kann nach verschiedenen Algorithmen realisiert werden, z.B. Round Robin oder nach Lastverhalten der Lesedatenbankenserver. In der `tnsnames.ora` auf dem

Glossar

Active Data Guard

Option in der Enterprise Edition. Eine Standby-Datenbank kann lesend geöffnet werden und wird dennoch von der Primary-Datenbank aktualisiert.

Data Guard Broker

Die Management-Schnittstelle zu Oracle Data Guard.

Fast Start Failover (Observer)

Bei einem Ausfall der Primary-Datenbank oder des Netzwerkes initiiert der Observer automatisch einen Failover zu einer vorher definierten Standby-Datenbank (Failover Target).

Flashback Database

Mit dieser Technik können Datenbanken schnell in einen Zustand der Vergangenheit zurückgesetzt werden, ohne zuvor ein Backup einzuspielen.

Loadbalancer

Der Loadbalancer ist ein Lastverteiler, der mit unterschiedlichen Verfahren, z.B. nach dem Round Robin-Verfahren (Rundlauf-Verfahren), eine Aufteilung von Client-Anfragen auf Servern vornimmt.

Round Robin (Rundlauf-Verfahren)

Round Robin ist ein Verfahren zur Lastverteilung beziehungsweise für ein Scheduling.

SCN

System Change Number - Eindeutiger Befehlszähler in der Datenbank.

Switchover

Geplanter Rollenwechsel der Primary- und Standby-Datenbank, z.B. bei Wartungsarbeiten.

Client/Webserver wird anschließend nur noch die IP-Adresse des Loadbalancer-Pool angegeben.

Aber auch die Primary- und K-Fall-Standby-Datenbankserver können in einem Loadbalancer-Pool konfiguriert werden. Mit einer eingebauten Abfragelogik werden Client-Anfragen über den Loadbalancer-Pool immer nur auf die Primary-Datenbank geleitet. Zudem können im Loadbalancer Prioritäten vergeben werden, z.B. wenn beide Datenbanken des Loadbalancer-Pool sich in der Primary-Rolle befinden, soll bevorzugt auf die Datenbank im primären Rechenzentrum zugegriffen werden.

Je nach Loadbalancer können diese Statusabfragen, z.B. mit JDBC an die Datenbank gestellt werden. D.h. beim Einsatz von Loadbalancer-Pools für die Client-Anbindung kann, auf die sonst übliche Service-Trigger-Erstellung innerhalb der Data-Guard-Datenbanken verzichtet werden.

Probleme während der Realisierung

Bei der Realisierung dieser komplexen Data-Guard-Lösung mit mehreren Standby-Datenbanken mussten

mit Oracle 11.2.0.3 auf RedHat 5.7 64bit eine Reihe von Problemen bewältigt und Workarounds durchgeführt werden:

- Damit ein manueller Switchover auf die K-Fall-Standby-Datenbank überhaupt funktionierte, musste der Patch 12829021 eingespielt werden. Mit diesem Patch wird folgender BUG gefixt:
BUG 12829021: SWITCHOVER FAILS WITH ORA-1093
- Bei der Verwendung der Flash Recovery Area für die Archive-Logs, gab es das Problem, dass sich die `database role` des Parameters `log_archive_dest_1 (valid for)` nach einem Switchover auf die neue Primary-Datenbank auf `standby role` änderte. D.h. die neue Primary-Datenbank konnte ohne manuellen Eingriff nicht mehr die Archive-Logs archivieren, da sie sich in der Primary-Rolle befand. Der Parameter `log_archive_dest_1` musste daher ohne den `db_unique_name` konfiguriert werden, siehe MOS ID: 1364467.1 Broker overrides local LOG_ARCHIVE_DEST_n destination on Bystander Standby.
- Ein weiteres Problem ist in BUG 13355993: SWITCHOVER HANGS USING DATA GUARD BROKER ON SWITCH BACK beschrieben. Durch das Setzen des Parameters `_smu_debug_mode=134217728` konnte ein Switchover und ein Failover nun ohne Probleme durchgeführt werden.
- Wurde die K-Fall-Standby-Datenbank „Read Only“ geöffnet, funktionierte der Switchover/Failover auf diese Datenbank nicht, wenn mit XDB/XML ein HTTP-Port konfiguriert wurde, hierfür ist auch BUG 12829021: SWITCHOVER FAILS WITH ORA-1093 verantwortlich.

Praxistipps

Zur Performance-Analyse können Statspack- und AWR-Reports nicht auf einer Read-Only-Datenbank erstellt werden. Oracle stellt ab der Version 11g für Read-Only-Standby-Datenbanken ein eigenes Standby-Statspack zur Verfügung, siehe MOS-Note: Installing and Using Standby Statspack in 11g [ID 454848.1].

Bei der Installation des Standby-Statspack wird in der Primary-Datenbank ein Datenbank-Link und ein Package `STATSPACK_<db_unique_name>_<db_name>` erstellt. Achten sie bei der Wahl des `db_unique_name` darauf, dass die Gesamtlänge des Package-Namen 30 Zeichen nicht überschreitet, sonst droht der Fehler ORA-972 ("identifier is too long").

Bei dem Einsatz von mehreren Standby-Datenbanken müssen die Archive-Prozesse erhöht werden, der

Default-Wert ist 4. Die Archive-Prozesse schließen die Lücke zwischen der Primary- und Standby-Datenbank und sorgen für den Heartbeat der Datenbanken innerhalb der Data-Guard-Konfiguration.

Die Archive-Prozesse sollten auf allen Datenbanken der Data Guard-Konfiguration auf mindestens $\langle \text{Anzahl aller Datenbanken} + 1 \rangle$ erhöht werden:

```
dgmgr1> edit database 'DB3_ST1' set property 'LogArchiveMaxProcesses'= 12;
```

Die Übertragung der Redo-Einträge erfolgt auf den Standby-Servern in die sog. Standby-Logs. Spiegeln sie die Standby-RedoLogs nicht. Verwenden sie für die Standby-Logs nur ein Member. So wird unnötige I/O-Last auf den Standby-Servern vermieden.

Der Primary- und die Standby-Server sollten das gleiche I/O-Subsystem haben, z.B. ein SAN (Storage Area Network). Wählen Sie auf den Standby-Datenbanken den Memory-Parameter `db_cache_size` größer als auf der Primary-Datenbank.

Fazit

Mit Active Data Guard können Lesefarmen einfach mit dem Data Guard Broker konfiguriert und verwaltet werden. Der Einsatz des Observer war in unserer Konfiguration empfehlenswert. Neben einen automatischen

Links

- ▶ [1] My Oracle Support (MOS): <https://support.oracle.com>

Failover, kümmert sich der Observer auch darum, dass alle Standby-Datenbanken synchron mit der neuen Primary-Datenbank sind und setzt die Lesedatenbanken notfalls automatisch auf den Stand der neuen Primary-Datenbank zurück.

Bei so komplexen Data-Guard-Umgebungen sollten immer ausführliche Tests mit der speziellen Konfiguration eingeplant werden. Profitieren Sie von unseren Erfahrungen beim Aufbau von komplexen und großen Data-Guard-Umgebungen. Gerne beraten und unterstützen wir Sie bei der Planung und Umsetzung. Sprechen Sie uns an.



*Michael Skowasch
(info@ordix.de).*

Seminarempfehlung: Oracle Data Guard

- ▶ **Informationen/Online-Anmeldung:** <http://training.ordix.de/siteengine/action/load/kategorie/Datenbanken/nr/493/index.html>

In diesem Seminar lernen Sie, mit Hilfe von Oracle Data Guard Ihr Datenbanksystem gegen ungeplante Ausfälle zu schützen und geplante Wartungsfenster zu minimieren. Die Data-Guard-Architektur sowie das Aufsetzen einer physikalischen und einer logischen Standby-Datenbank sind die zentralen Themen dieses Workshops.

Seminarinhalte

- Data Guard: Ideen, Konzepte
- Physical Data Guard: Darstellung und Umsetzung der unterschiedlichen Möglichkeiten, Einsatzgebiete, Beispiele
- Logical Data Guard: Darstellung und Umsetzung der unterschiedlichen Möglichkeiten, Einsatzgebiete, Beispiele
- Data Guard Broker: Die Administrationsschnittstelle
- Fast Start FailOver (Observer)
- Monitoring, Performance-Analyse, Troubleshooting
- RMAN-Sicherungen unter Data-Guard-Aspekten
- Vertiefung der Theorie durch praktische Übungen und Beispiele

Termine

- 02.07. - 05.07.2012 in Wiesbaden
- 24.09. - 27.09.2012 in Wiesbaden
- 22.10. - 25.10.2012 in Bielefeld
- 26.11. - 29.11.2012 in Wiesbaden

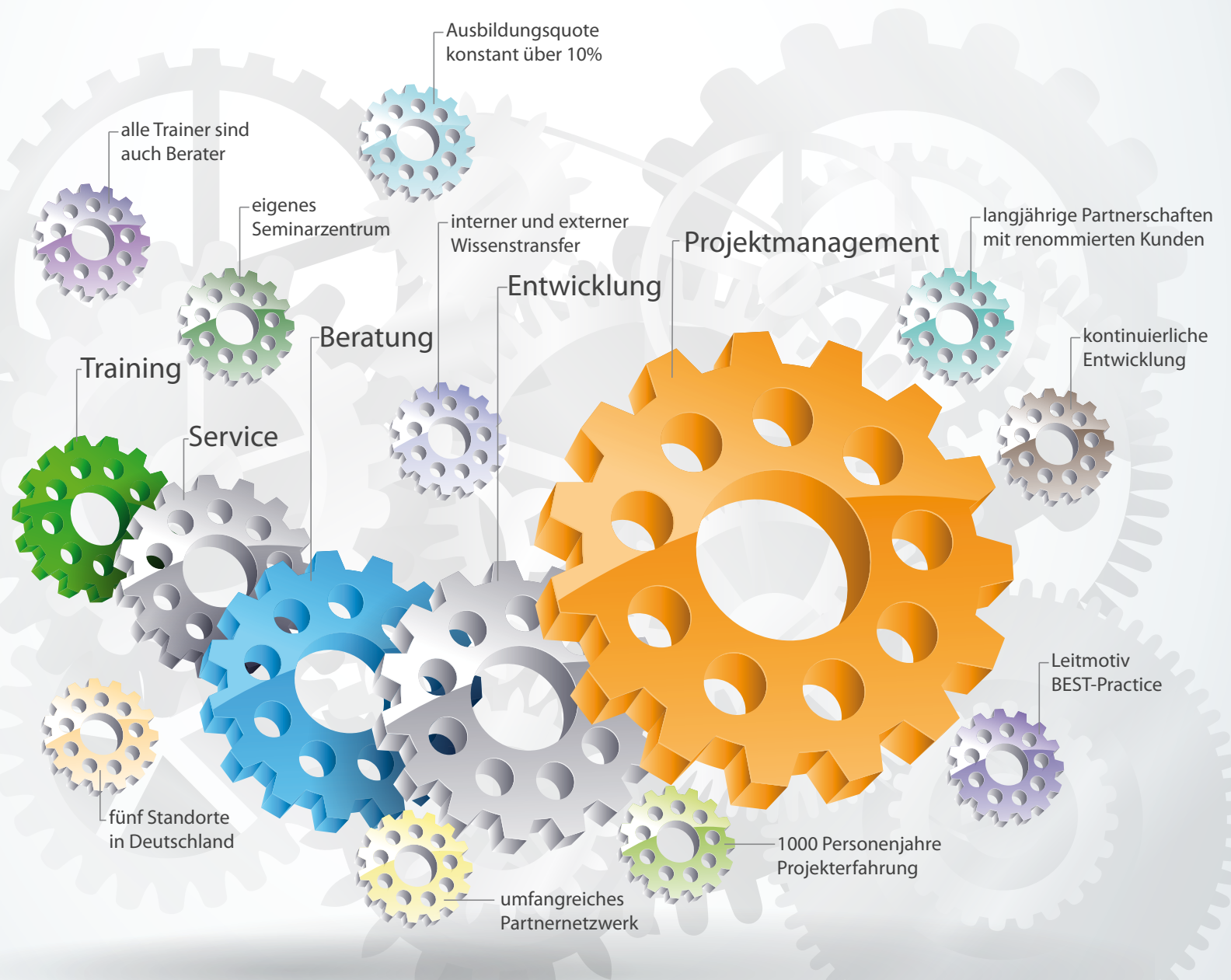
Seminar-ID: DB-ORA-31

Dauer: 4 Tage

Preis pro Teilnehmer: 1.590,00 € (zzgl. MwSt.)

Frühbucherpreis: 1.431,00 € (zzgl. MwSt.)

VERTRAUEN ist die BASIS für den gemeinsamen ERFOLG!



IT-Projekte in externe Hände zu legen, erfordert großes Vertrauen.

Dieses Vertrauen verdienen wir uns durch langfristig ausgerichtete, verlässliche Partnerschaften mit unseren Kunden.

Für viele unserer Kunden ist ORDIX seit mehr als 20 Jahren ein bevorzugter Partner. Wir liefern, was wir versprechen.

- Professionelles IT-Projektmanagement – Ihre Projekte kompetent gesteuert
- Kompetente IT-Beratung – Richtige Entscheidungen steigern Ihren Erfolg
- Hochqualifizierte IT-Entwicklung – Lösungen so individuell wie Ihr Unternehmen
- Zuverlässiger IT-Service – Reibungsloser IT-Betrieb als Standard
- Individuelles Training schafft Mehrwert

Weitere Informationen unter:

<http://www.ordix.de/>

ORDIX AG