Nr. 3 | Juli 2013 | ISSN 09 36-0360 | www.doag.org

DOAG News

Deutsche ORACLE-Anwendergruppe e.V.

Oracle **Infrastrukturen** und **Plattformen**

Solaris

Best Practices für Datenbanken auf ZFS

Was sie von Oracle über ZFS nicht hören werden

Virtual Machine

Oracle VM3 (x86) — was sich getan hat

Backup einer Oracle-VM3-Umgebung

SPARC

Erfahrungen aus den T5-Beta-Tests

OPITZ CONSULTING

Menschen.
Innovationen.
Lösungen.

Wir haben vielleicht keine Superkräfte, aber dafür bieten wir Ihnen Oracle IT-Betrieb in Bestform!

OC|MSI®

Entlasten Sie wertvolle Ressourcen und stellen Sie dennoch die Verfügbarkeit aller Systeme sicher:

Unser OC|MSI® Team unterstützt Sie beim alltäglichen Betrieb Ihrer IT-Landschaft und sorgt für maßgeschneiderte Systemverfügbarkeit, Administration Ihrer IT-Systeme, Pflege, Überwachung, Optimierung – und Sie haben Zeit für Ihre innovativen IT-Projekte!

Managed Services Infrastructure bietet Ihnen herstellerunabhängige Beratung und individuelle auf Sie angepasste Lösungen, remote oder vor Ort, 24/7 oder als Urlaubsvertretung.

Auf Wunsch betreuen wir Ihre IT-Infrastruktur in einer sicheren Service Cloud und sichern so Ihre Business Continuity.

Erfahren Sie mehr über unsere Leistungen im Bereich Managed Services Infrastructure unter www.opitz-consulting.com/msi

Ihr direkter Ansprechpartner ist Najibullah Rajab.
Telefon: +49 2261 6001-0 · E-Mail: Najibullah.Rajab@opitz-consulting.com

Service Cloud
by OPITZ CONSULTING

Wir betreuen Ihre Systeme in einer sicheren Service Cloud.

- Service Cloud Appliance als Hardware oder VM vor Ort
- Hosting der Service Cloud in hochverfügbarem, abgesichertem Rechenzentrum
- Datenübermittlung über ein sicheres und verschlüsseltes Protokoll
- Problemlösung via Incident Management, Service Desk, Request Management und Knowledge Management

Wir sichern Ihre Business Continuity.

Björn Bröhl
Leiter Infrastruktur & Middleware Community

Liebe Mitglieder der DOAG Deutsche ORACLE-Anwendergruppe,
liebe Leserinnen und Leser,

der Schwerpunkt dieser Ausgabe liegt auf „Best Practices“. Sicherlich hat jeder von uns bereits von den Erfahrungen anderer profitiert. Der Erfahrungsaustausch der Anwender ist eines der wichtigsten Ziele der DOAG. Die in dieser Ausgabe enthaltenen Artikel sind aus dem Themenbereich „Infrastruktur“. Neben der neuen Oracle-CPU-Generation T5 finden Sie mehrere Artikel rund um Solaris sowie Oracle VM. Darüber hinaus gibt es als Nachtrag zur letzten Ausgabe noch einen Mini-Schwerpunkt „ETL“.

Best Practices vereinfachen in der Regel kompliziert anmutende Aufgaben. Diese sind bei der Infrastruktur besonders gefragt, wenn Hardware, Betriebssysteme, Virtualisierungs-Technologien sowie die Anwendungen optimal zusammenarbeiten sollen.

Nachdem Solaris 11 mittlerweile seit einem Jahr allgemein verfügbar ist und noch im Jahr 2012 das erste Update-Release ansteht, ist momentan ein guter Zeitpunkt für den Austausch von Erfahrungen bei der Einführung von oder dem Übergang auf Solaris 11. Erfahrungen mit den aktuellen SPARC- und x86-Sytemen sowie den Speicherlösungen sind ebenso wichtig wie der Austausch über die Engineered Systems.

Auch die Administration der Infrastruktur ist nicht zu vernachlässigen. Mit dem Enterprise Manager OpsCenter 12c ist ein Werkzeug verfügbar, das die Einführung neuer Technologien wie IPS und AutoInstall in Solaris 11 mit ihren spezifischen administrativen Konzepten vereinfachen kann. OpsCenter und nicht zuletzt die Komponenten des Enterprise Manager 12c zum Cloud-Management können auch ein weiteres, für viele in der Praxis neuartiges Konzept unterstützen: das Cloud-Computing, auf dem dieses Mal ein besonderer Fokus liegt.

Ich hoffe, dass auch Sie in dieser Ausgabe Best Practices finden, die Ihnen die Arbeit erleichtern und dabei helfen, Zeit und Nerven zu sparen. Viel Spaß beim Lesen wünscht Ihnen

Ihr

Eine schwere Netzwerk-Aufgabe mit VM Server SPARC, Seite 32

Automatische Generierung der ETL-Prozesse, Seite 46

Seit Februar 2013 steht MySQL 5.6 als Produktions-Release zur Verfügung, Seite 55

Einleitung

SPARC

Solaris

Engineered Systems

Virtual Machine

ETL

Aktuell

Best Practice

Tipps und Tricks

DOAG intern

Spotlight

Montag, 8. April 2013

In Denver (USA) kommen rund 7.000 Besucher zur Collaborate 13, eine der weltweit größten Oracle-Konferenzen. Die Vertreter der DOAG nutzen die Gelegenheit, um sich mit den großen amerikanischen Anwendergruppen auszutauschen. Im Gespräch mit der Oracle Applications Users Group (OAUG) geht es vor allem um die Vorbereitung zur DOAG 2013 Applications Konferenz + Ausstellung in Berlin. Steven R. Hughes, Executive Director der OAUG, sichert zu, interessante Speaker zum Thema „Fusion Applications" nach Berlin zu bringen. Beim Meeting mit der Independent Oracle Users Group (IOUG) schließt sich die IOUG der DOAG-Forderung an, die Agenda des International Usergroup Summit, bei dem sich die Repräsentanten der großen und überregionalen Oracle-Usergruppen in den Oracle-Headquarters treffen, müsse wieder mehr im Sinne der Usergroups sein. Dies will man nun gemeinsam bei Oracle adressieren.

Mittwoch, 17. April 2013

Christian Weinberger, Leiter DOAG Business Intelligence & Data Warehouse, eröffnet die DOAG 2013 BI, die zum dritten Mal in Folge in München stattfindet. Die Community Konferenz ist zwischenzeitlich etabliert und hat stabile Teilnehmerzahlen.

Donnerstag, 25. April 2013

Unter dem Motto „Innovation in der Logistik" öffnet in Hamburg die diesjährige DOAG 2013 Logistik Konferenz ihre Pforten. Prof. Dr. Michael ten Hompel appelliert am Ende seiner Keynote an die Zuhörer, das Jahrhundert der Logistik auszurufen. Er ist der Ansicht, dass wir in Deutschland jetzt reagieren müssen, damit nicht andere Länder dieses innovative Feld besetzen.

Dienstag, 30. April 2013

Fried Saacke, DOAG-Vorstand und Geschäftsführer, trifft sich in Berlin mit dem Geschäftsführer des Unternehmens für das Catering der DOAG 2013 Konferenz + Ausstellung. Sie entwickeln gute Ideen, um den Festabend noch gemütlicher und attraktiver zu gestalten.

Freitag, 3. Mai 2013

Im Rahmen der DOAG-Vorstandsitzung werden die Weichen für die Delegiertenversammlung gestellt. Die vier Community-Leiter berichten über ihre erfolgreich abgehaltenen Community Konferenzen beziehungsweise über den Stand der Planung für die Veranstaltungen, die noch folgen.

Dienstag, 14. Mai 2013

Die DOAG 2013 Datenbank in Düsseldorf ist mit rund 250 Teilnehmern ein großer Erfolg. Die Veranstaltung startet mit einer Keynote von Günther Stürner, Vice President Server Technologies und Sales Consulting bei Oracle. Das Urgestein der Oracle-Datenbank in Deutschland blickt visionär in die Zukunft und propagiert einen Data Scientist, der sowohl die Datenbank-Technologie als auch statistische Methoden und Zusammenhänge kennt. Nur so sei ein Unternehmen in der Lage, aus Big Data brauchbare Ergebnisse zu gewinnen.

Mittwoch, 15. Mai 2013

Rund 1.000 Besucher kommen zur Oracle OpenCloud nach München. Der Stand der DOAG verzeichnet einen guten Zulauf. In zahlreichen Gesprächen entsteht der Eindruck, dass die Besucher nur wenige Oracle-Produkte einsetzen und man in die Cloud-Thematik reinschnuppern möchte.

Mittwoch 22. Mai 2013

Fried Saacke, DOAG-Vorstand und Geschäftsführer, und Christian Trieb, Leiter der Datenbank Community und zuständig für internationale Arbeit, vertreten die DOAG auf dem Treffen der EMEA Oracle User Group Community (EOUC). Im Fokus steht der Erfahrungsaustausch der europäischen Anwendergruppen. Die langjährige Forderung der DOAG, die Veranstaltung zum Austausch mit dem europäischen Top-Management von Oracle zu nutzen, wird wieder nicht erfüllt.

DOAG 2013 Konferenz + Ausstellung

Geballtes Know-How zum Greifen nah!

Je mehr Praxisbezug desto besser!

- Erfahrungsbericht
- Projektberichte
- Tipps und Tricks
- Neuerscheinungen oder Features
- Best Practices

Unterstützen Sie als Oracle-Anwender und -Spezialist die DOAG mit Ihrem Vortrag

19.–22. Nov. 2013
Nürnberg NCC Ost

Eine Veranstaltung der DOAG mit

ORACLE · iJUG Verbund · SOUG · AOUG

2013 DOAG Konferenz + Ausstellung
www.doag.org/go/cfp

Andreas Graf (Links) und Dr. Martin Haller (Mitte) im Gespräch mit Björn Bröhl

Fotos: Wolfgang Taschner

„Unser großer Wunsch wäre eine bessere Integration der SAP-Anwendungen …“

Um die vielfältigen Aufgabe einer Kommune zu erledigen, ist eine ausgefeilte Infrastruktur erforderlich. Björn Bröhl, Leiter der Infrastruktur & Middleware Community, und Wolfgang Taschner, Chefredakteur der DOAG *News*, sprachen darüber mit Dr. Martin Haller, Bereichsleiter Systemmanagement, und Andreas Graf, zuständig für Aufbau und Weiterentwicklung der zentralen Systemtechnik bei der Stadt Nürnberg.

Was sind die größten Anforderungen der Stadt Nürnberg an die IT?

Dr. Haller: Die Besonderheit bei einer Kommune ist das vielfältige Anwendungsspektrum. Das kommt daher, dass wir im Vergleich zu einem Unternehmen eine sehr umfangreiche und vielschichtige Palette an Aufgaben in den unterschiedlichsten Bereichen zu erfüllen haben. Darüber hinaus müssen wir vor allem bei den Diensten, die wir für die Bürger bereitstellen, eine hohe Verfügbarkeit bieten. Eine weitere Herausforderung ist in Zeiten der knappen Etats der Blick auf die Wirtschaftlichkeit unserer IT. Hierzu sind insbesondere effiziente Tools zum Managen der Systeme gefragt.

Wie sieht das Spektrum Ihrer Anwendungen aus?

Dr. Haller: Als ich vor fünfzehn Jahren bei der Stadt Nürnberg angefangen habe, gab es noch eine große Programmier-Abteilung, die Anwendungen nach den individuellen Anforderungen der Fachabteilungen hauptsächlich auf einer Mainframe-Umgebung erstellt hat. Heute hingegen setzen wir weitgehend auf Standard-Software, die wir allenfalls individuell konfigurieren. Die Anwendungen umfassen die gesamte Bandbreite einer Kommune, angefangen vom Standesamtswesen über alle Einwohner- und Kfz-Angelegenheiten bis hin zum Betrieb von Bibliotheken.

Die IT der Stadt Nürnberg

- Rund 6.500 Anwenderinnen und Anwender an 125 Standorten sowie rund 200 Fachverfahren
- Mehr als 350 (überwiegend virtuelle) Server in zwei Rechenzentren im Einsatz, davon etwa 40 Prozent unter UNIX-Betriebssystemen (Linux und Solaris) und 60 Prozent unter Windows
- Mehr als 30 Oracle-Datenbank-Server mit mehr als 150 bereitgestellten Oracle-Instanzen
- Storage-Volumen von etwa 140 TB (netto, ohne Spiegel)

Wie ist Ihre IT-Plattform aufgebaut?
Dr. Haller: Wir haben hier einen großen Windows-Bereich mit einer zentralen Active- Directory-Verzeichnisstruktur, auf der unter anderem Mail- und Dokumenten-Dienste abgewickelt werden. Dann gibt es eine große Virtualisierungslandschaft auf VMware-Basis. Datenbanken und die SAP-Lösungen sowie andere zentrale Fachverfahren laufen in einer Solaris-Umgebung. Wir betreiben zwei Rechenzentren, in denen die kritischen Anwendungen gespiegelt ablaufen. Die Hochverfügbarkeit ist durch Cluster-Systeme hergestellt, wobei hier Solaris-Zonen zum Einsatz kommen.

Wie managen Sie diese IT-Plattform?
Dr. Haller: Wir haben ITIL-Prozesse implementiert und ein Konfigurationsmanagement aufgebaut. Das Monitoring erfolgt auf Basis des Open-Source-Tools Icinga, einer Nagios-Variante.

Worin sehen Sie die Vorteile von Solaris?
Graf: Solaris ist schon immer ein bewährtes und stabiles Betriebssystem gewesen. Verglichen mit Linux, wo es eine Vielzahl von Distributionen gibt, ist Solaris ausgereift und vom Hersteller gut unterstützt. Uns kommt es auch entgegen, dass Solaris beispielsweise bereits eine Cluster-Funktionalität bietet. Seit der Version 11 hat sich auch das Update-Management stark verbessert. Im Zuge des Aufbaus einer neuen SAP-Anwendung sind wir dann auch gleich auf Solaris 11 umgestiegen. Ein weiterer Vorteil von Solaris ist die Beständigkeit der administrativen Tools und Konzepte.

Welche Tools nutzen Sie für die Administration Ihrer Umgebungen?
Dr. Haller: Wir setzen Enterprise Manager Cloud Control ein und haben damit gute Erfahrungen gemacht. Wünschenswert wäre hier eine Schnittstelle zu Nagios.

Setzen Sie auch Oracle Enterprise Manager Ops Center 12c ein?
Dr. Haller: Das haben wir bisher nicht in Betrieb, da wir ja bereits andere Tools einsetzen. Wir planen aber demnächst einen Test, da uns das grafische Cluster-Management interessiert, was ja in Solaris 11 nicht mehr enthalten ist. Außerdem wäre es gut, wenn Oracle die Ankündigung, Enterprise Manager Cloud Control und Enterprise Manager Ops Center in einem Produkt zusammenzuführen, bald umsetzen würde.

Setzen Sie Solaris ausschließlich auf der SPARC-Plattform ein?
Graf: Nein. Wir hatten zwar früher eine Sun Fire 12K in Betrieb, die wir durch zwei M5000-Server abgelöst haben, setzen allerdings mittlerweile zunehmend x86-Server ein. Das hat sich in der Praxis gut bewährt. Die SPARC-Systeme sind sehr stark auf Parallelisierung ausgelegt, wohingegen unsere Anwender-Software eher Single-Thread-Performance benötigt, die das Intel-Umfeld besser abdeckt.

Welche Erwartungen haben Sie an die neuen SPARC-Server, die auf dem Mikroprozessor T5 basieren?
Dr. Haller: Oracle hat uns über die neuen Systeme gut informiert und wir beobachten technische Entwicklung, insbesondere die Leistungssteigerung der SPARC-CPUs. Derzeit stehen die Signale bei uns eher in Richtung Ausbau der x86-Server.

Wie beurteilen Sie die SPARC-Server im Vergleich zu den x86-Systemen?
Dr. Haller: Das ist in erster Linie natürlich eine Preisfrage. Die neuen SPARC-Server bieten zwar in mancher Hinsicht eine bessere Integration der Oracle-Software, zum Beispiel durch Optimierung der CPUs für Datenbank-Operationen, doch momentan verzichten wir aus Kostengründen auf diese höhere Performance. Zudem können wir aufgrund der standardisierten x86-Technologie flexibler reagieren.

Haben Sie schon einmal über den Einsatz eines Oracle Engineered System nachgedacht?
Graf: Wir haben uns die Exadata-Maschine angeschaut. Sie passt allerdings nicht so gut in unsere IT-Struktur. Wir

Zur Person: Dr. Martin Haller
Der Diplomphysiker verfügt über dreißig Jahre Berufspraxis in Forschung, Lehre und Datenverarbeitung. Wissenschaftliches Arbeiten hat er unter anderem an den Universitäten Erlangen-Nürnberg und Hamburg sowie am Forschungszentrum DESY durchgeführt. Er ist seit 1998 bei der Stadt Nürnberg, derzeit als Bereichsleiter Systemmanagement im Amt für Organisation, Informationsverarbeitung und Zentrale Dienste zuständig für Planung, Konzeption, Optimierung und Weiterentwicklung der zentralen IT-Systeme in den Rechenzentren der Stadt Nürnberg.

Zur Person: Andreas Graf
Der staatlich geprüfte Elektrotechniker (Datenverarbeitungstechnik) hat 19 Jahre Erfahrung im Bereich UNIX/Linux. Er war bis zum Jahr 2001 mit der Betreuung und Weiterentwicklung von UNIX-Systemen beim Technischen Finanzamt Nürnberg befasst. Danach hat er bei der Stadt Nürnberg zunächst zwei Jahre eine Infrastruktur für E-Government-Anwendungen der Curiavant Internet GmbH mit aufgebaut und ist seit 2003 im Bereich Systemmanagement zuständig für Aufbau und Weiterentwicklung der zentralen Systemtechnik, insbesondere Server, Storage-Systeme, System- und System-nahe Software sowie Datenbanken.

betreiben unsere Datenbanken auf dedizierten Cluster-Systemen, was uns die Möglichkeit bietet, diese flexibel einzusetzen. Außerdem haben wir hinsichtlich Hochverfügbarkeit mit unseren Failover-Clustern gute Erfahrungen gemacht, sodass eine Exadata auch hier nicht optimal ins Konzept passen würde.

Können Sie sich vorstellen, ein Komplettsystem von der Hardware bis zu den Applikationen von einem einzigen Hersteller wie Oracle einzusetzen?
Dr. Haller: Sie sprechen den sogenannten „Red Stack" von Oracle an. Wir sind der Meinung, dass solche performanten Systeme für Unternehmen eine Überlegung wert sind, wenn sie exakt diese Anforderungen haben. Dadurch, dass bei uns – wie bei viele anderen Firmen in Deutschland auch – SAP-Anwendungen im Mittelpunkt stehen, ist der obere Bereich des Red Stack – die Applikationsebene – für uns nicht interessant. Ein weiterer Nachteil einer solchen Komplettlösung besteht in der resultierenden Abhängigkeit von einem einzigen Hersteller. Hier gilt es immer, den Vorteil der abgestimmten Systeme gegen mögliche wirtschaftliche Nachteile abzuwägen. Momentan setzen wir hier lieber auf Standards, um im Interesse der Steuerzahler kostengünstig einkaufen zu können.

Was haben Sie damals empfunden, als Oracle Sun übernommen hat?
Dr. Haller: Das war schon eine turbulente Zeit, da die Ansprechpartner häufig gewechselt haben. Zudem wussten wir nicht, wie es mit der Produkt- und Lizenzierungs-Politik weitergehen wird.

Was hat sich seit dieser Zeit verändert?
Dr. Haller: Wir haben die besondere Offenheit von Sun immer sehr geschätzt. Oracle agiert hier etwas formaler. Dennoch zeigt Oracle großes Interesse am Feedback seiner Kunden.

In welche Richtung wird sich Ihre IT in den kommenden Jahren entwickeln?
Dr. Haller: Die fachliche Vielschichtigkeit wird weiter zunehmen. Eine weitere Herausforderung ist die wachsenden Datenmenge und die damit verbundenen Anforderungen an das Datenmanagement. Auch die Integration mobiler Endgeräte wird eine größere Rolle spielen als heute – einmal als Endgerät in den Fachabteilungen und zum anderen in Form von Apps für die Bürgerinnen und Bürger der Stadt.

Was erwarten Sie dabei von einem IT-Unternehmen wie Oracle?
Dr. Haller: Unser großer Wunsch wäre eine bessere Integration der SAP-Anwendungen. Wir haben bei der Implementierung der Cluster festgestellt, dass diese wegen ihrer Komplexität und der Vielzahl von Services ganz andere Anforderungen stellen als beispielweise eine Oracle-Datenbank. Auch die Management-Werkzeuge könnten durchaus noch etwas effizienter und komfortabler werden. Außerdem käme uns eine moderate Preispolitik entgegen, da wir im öffentlichen Bereich sehr stark auf die Kosten achten müssen.

Wie sehen Sie den Stellenwert einer Anwendergruppe wie der DOAG?
Graf: Die letzte DOAG Konferenz und Ausstellung hat uns sehr gut gefallen. Insgesamt bietet die DOAG viele Möglichkeiten zur Informationsgewinnung und zum Erfahrungsaustausch. Auch die Interessenvertretung gegenüber Oracle hat für mich einen großen Stellenwert.

Neu: Apex 4.2.2 ist da

Mit dem Patchset wird Apex in etwa zehn Minuten auf die Version 4.2.2.00.11 gebracht, wobei das bestehende Datenbank-Schema „APEX_040200" weiterhin seine Verwendung findet. Es empfiehlt sich, vor der Installation eine Sicherung der Datenbank vorzunehmen, da das Patchset nur bei erfolgter Installation erneut eingespielt werden kann.
Neuerungen gibt es vor allem in den Themes 25 und 50. Deren Templates werden in bestehenden Anwendungen durch das Patchset automatisch aktualisiert. Importiert man allerdings eine Apex-4.2.1-Anwendung, sind manuelle Anpassungen notwendig.
Weitere Informationen unter: http://www.doag.org/home/aktuelle-news/article/oracle-application-express-422-ist-da.html

Am 26. März 2013 stellte Oracle die neuen T5-Server vor. Bereits lange vor diesem Termin wurden einige Vorserien-Systeme in einem Beta-Programm ausgiebig getestet. Neben anderen europäischen Anwendern hat auch Swisscom an diesem Programm teilgenommen und eine T5-4 intensiv geprüft. Dieser Artikel beschreibt das Beta-Programm und die Erfahrungen, die die Swisscom IT Services AG mit den neuen Systemen gemacht hat.

Erfahrungen aus den T5-Beta-Tests

Stefan Muehlebach, Swisscom IT Services AG, und Stefan Hinker, ORACLE Deutschland B.V. & Co. KG

Bei Oracle werden schon seit Langem neu einzuführende Server vorab in einem weltweiten Beta-Programm getestet. Dabei stehen zwei wesentliche Ziele im Vordergrund: Einerseits sollen die Systeme in möglichst unterschiedlichen Umgebungen so realitätsnah wie möglich getestet werden. Andererseits erhofft sich Oracle einige herausragende Testergebnisse, sodass bereits bei der Einführung einige Erfahrungsberichte und Referenzen angegeben werden können.

Das Testen durch die Anwender ist in erster Linie für die Entwicklungsabteilungen von Interesse. Selbstverständlich werden die Systeme in jedem Stadium der Entwicklung ständig, intensiv und systematisch erprobt. Doch gerade die absolut notwendige Systematik verhindert, dass ungeplante Dinge passieren. Beim Einsatz in einem Beta-Test ist dies anders. Die Systeme kommen in sehr unterschiedliche, von Oracle nicht kontrollierte Umgebungen. Die Anwender haben noch keine Erfahrung mit diesen Systemen und können daher auch nicht unbewusst Fehler umgehen. Sie sind aufgefordert, die Systeme bis über deren Leistungsgrenzen hinaus zu belasten. Sollten dabei unerwartete Probleme auftreten, ist dies in der Entwicklung hoch willkommen, denn es besteht noch die Möglichkeit, das Produkt gegebenenfalls zu verändern.

Wirklich dramatische Punkte kamen in den Testreihen von T2 bis T5 nicht ans Licht. In dieser vorletzten Entwicklungsstufe der Systeme ist dies auch sehr unwahrscheinlich. Einige kleinere mechanische Unregelmäßigkeiten kamen jedoch ab und zu vor und wurden bis zur Produkteinführung beseitigt. Die Erfahrungen der Anwender in diesen Tests werden in der Entwicklung sehr geschätzt und fließen in zukünftige Entwicklungen und Testverfahren ein.

Für alle Beteiligten nicht weniger wichtig ist der Erfahrungsaustausch während eines Beta-Tests. Dieser ist für einen Anwender mit einigem Aufwand verbunden, müssen doch ein Testplan entwickelt, Ressourcen für die Durchführung und Auswertung zur Verfügung gestellt und ein Abschlussbericht geschrieben werden. Oracle unterstützt diese Aktivitäten durch die Bereitstellung von Performance-Spezialisten mit direktem Draht in die Entwicklung sowie einer speziellen Service-Infrastruktur. Es kommt für die Dauer des Tests zu intensiver Zusammenarbeit. Die dabei geschlossenen oder gefestigten Kontakte bleiben häufig auch nach Abschluss der Tests bestehen, was für beide Seiten vorteilhaft ist.

Für den Anwender selbst ist die Teilnahme am Beta-Programm eine gute Möglichkeit, schon vor der öffentlichen Ankündigung eines neuen Systems dieses auf Herz und Nieren zu testen und für den eigenen Einsatz zu bewerten. Der damit verbundene Aufwand wird oft durch den Zeitvorteil und die bereits erwähnte intensive Zusammenarbeit mehr als ausgeglichen. So ist insgesamt ein solcher Beta-Test für beide Seiten ein Gewinn.

Die SPARC-T5-CPU und die Systeme

Grundlage der SPARC-T5-CPU ist der gleiche Kern, der sich bereits in der SPARC-T4 bewährt hat. Wie bei den CPUs der T-Serie üblich, befindet sich neben den eigentlichen CPU-Kernen jedoch ein Großteil der Infrastruktur ebenfalls auf dem CPU-Chip. Hier erfuhr die T5-CPU die meisten Verbesserungen. Ziel war es, die Gesamtleistung des Chips gegenüber der T4 mindestens zu verdoppeln und gleichzeitig die Skalierbarkeit von vier auf acht Sockel zu verbessern (siehe Abbildung 1).

Die offensichtlichste Maßnahme hierfür war neben der moderaten Erhöhung der Taktrate von 3.0 GHz auf 3.6 GHz die Verdopplung der Kerne

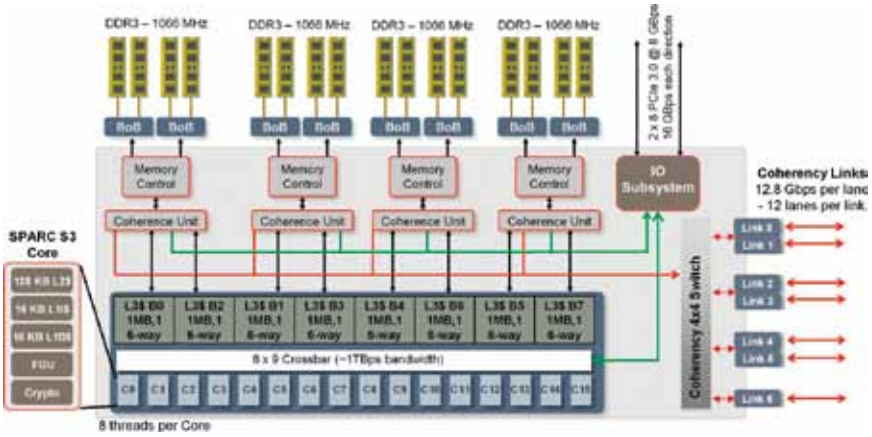

Abbildung 1: Blockschaltbild der T5-CPU (Quelle: [2])

von acht auf sechzehn. Um mit dem damit mehr als verdoppelten Bandbreitenbedarf gegenüber der T4 Schritt halten zu können, wurden nicht nur der L3-Cache entsprechend angepasst, sondern, wichtiger noch, die Memory Controller erneuert. Die bekannte Durchsatzstärke der SPARC-CMT-Architektur erfordert hohe Bandbreiten zum Hauptspeicher, diesem wurde mit einer Steigerung um etwa den Faktor zweieinhalb gegenüber der T4 Rechnung getragen. Für die höhere Skalierbarkeit von bis zu acht CPUs pro System schließlich wurde das Kohärenz-Protokoll völlig neu entwickelt. Das bisherige Snooping-Verfahren wurde durch ein Protokoll auf Verzeichnis-Basis abgelöst, was nicht nur niedrigere Latenzen im Zugriff auf entfernten Speicher ermöglicht, sondern gleichzeitig sehr viel effizienter mit der verfügbaren Bandbreite umgeht.

Auch im IO-Bereich wurde modernisiert. Hier hat man die PCIe-2.0-Controller durch aktuelle PCIe-3.0-Controller ersetzt, was pro Controller eine Verdopplung des Durchsatzes ermöglicht. Die bisher ebenfalls auf dem Chip integrierte 10-Gbit-Ethernet-Schnittstelle hingegen entfällt bei T5. Die bei Einführung der T1 im Jahr 2007 noch neue 10-Gbit-Technik ist inzwischen weit genug fortgeschritten, sodass die Integration in die CPU überflüsig wird.

Auf der Grundlage dieser neuen CPU wurde nun die bereits gut eingeführte Familie der SPARC-T4-Systeme um vier weitere Mitglieder erweitert. Neben einem Ein-Sockel-Blade-System gibt es in nur wenig modifizierten Gehäusen die SPARC T5-2 und SPARC T5-4 mit zwei beziehungsweise vier CPUs. Als neues Spitzen-Modell kommt mit acht CPUs die SPARC T5-8 dazu. Damit umfasst die SPARC-T-Serie nun Systeme von vier (Netra T4-1) bis 128 Kernen (T5-8).

Die T5-4 im praktischen Einsatz

Die Swisscom IT Services Finance AG betreibt eine der größten Banken-Plattformen der Schweiz. Auf den Core-Systemen kommen zwei Standard-Produkte zum Einsatz: Avaloq und Finnova. Im Rahmen eines „Proof of Concept" wurde eine große Finnova-Installation auf einer T5-4 getestet.

In den letzten fünf bis zehn Jahren gab es im Schweizer Banken-Umfeld einen sehr augenfälligen Trend: Die proprietären Lösungen für das Core-System wurden durch Standard-Produkte abgelöst. Eines dieser Produkte heißt „Finnova", eine Komplettlösung, die vom Kunden (oder dem Betreiber) nur noch parametriert werden muss – eine Erweiterung der Software durch eigenen Code ist nicht erforderlich beziehungsweise nicht vorgesehen.

Im Zentrum einer jeden Finnova-Plattform steht das Core-System. Auf diesem System läuft im Wesentlichen eine Oracle-Datenbank, die sowohl die Daten (Kunden, Konten, Buchungen etc.) als auch die bankfachliche Logik (PL/SQL-Packages) enthält. Neben der Datenbank befinden sich auf dem Core-System zwei weitere Arten von Prozessen (siehe Abbildung 2):

- *Service-Prozesse*
  Über die Service-Prozesse können externe Programme via HTTP auf die angebundene Core-Datenbank zugreifen. Wichtigstes Beispiel eines solchen externen Programms ist der Finnova-Client, ein grafisches User-Interface, das auf den Arbeitsplätzen der Bankmitarbeiter installiert ist. Die Service-Prozesse sind in Java realisiert.
- *Batch-Prozesse*
  Über die Batch-Prozesse läuft die asynchrone Verarbeitung wie zum Beispiel die Verbuchung von Zahlungen, die Berechnung von Gebühren und Zinsen, aber auch die Tages-Endverarbeitung (TEV). Die Batch-Prozesse sind in Perl realisiert und beziehen ihre Aufträge (Jobs) über eine zentrale Queue (Oracle Advanced Queue) aus der Core-Datenbank. Auf einer produktiven Umgebung laufen zwischen 20 und 40 dieser Batch-Prozesse parallel.

Das Core-System muss somit zwei Bedingungen erfüllen:

- Anfragen der Clients schnell verarbeiten können (gute Single-Thread Performance)
- In Hochlast-Situationen mit einer großen Anzahl paralleler Batch-Prozesse umgehen können

Im Jahr 2009 wurde für einen Bankenverbund mit mehr als zwanzig Kunden das Core-System auf eine M8000 migriert. Vier Jahre später steht nun die nächste Migration an und als Zielsystem wurde zunächst eine T4-4 ins Auge gefasst. Um die neue Architektur besser kennenzulernen, wurden eine Testumgebung auf einer T4-4 aufgebaut und die Maschine auf die beiden Hauptkriterien (Single-Thread Performance, Parallel Processing) überprüft. Dabei stellte sich Folgendes heraus:

- Im Bereich der Single-Thread-Performance leistete die T4-4 ungefähr das Gleiche wie die M8000. Das war insofern störend, als der Kunde durch den Wechsel auf eine neue Plattform eine Verbesserung der Antwortzeiten erwartet hatte.
- Im Umgang mit einer großen Anzahl von Batch-Prozessen zeigte die T4-4 jedoch deutliche Vorteile gegenüber der M8000. Mit einer Verdopplung der Batch-Prozesse (von 40 auf 80) konnte die Laufzeit der Quartals-Endverarbeitung (QEV) glatt halbiert werden – dabei war das System im Durchschnitt nur zu 30 Prozent ausgelastet. Zum Vergleich: 80 Batch-Prozesse auf der M8000 würden das System fast zum Stillstand bringen.

Als Oracle mit den Resultaten konfrontiert wurde, bestätigte man diese. Der Verweis auf den wesentlich günstigeren Preis der T4-4 gegenüber der M8000 konnte den Kunden nicht überzeugen: Kürzere Antwortzeiten hatten für ihn höhere Priorität. Um die für den Kunden geforderten Antwortzeiten zu erreichen, wurde das Projekt in das Beta-Testprogramm von Oracle aufgenommen und die Swisscom IT Services AG erhielt im Dezember 2012

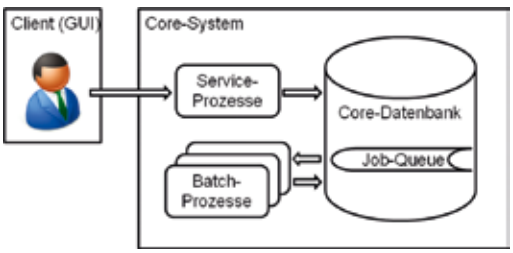

Abbildung 2: Big Picture einer Finnova-Umgebung

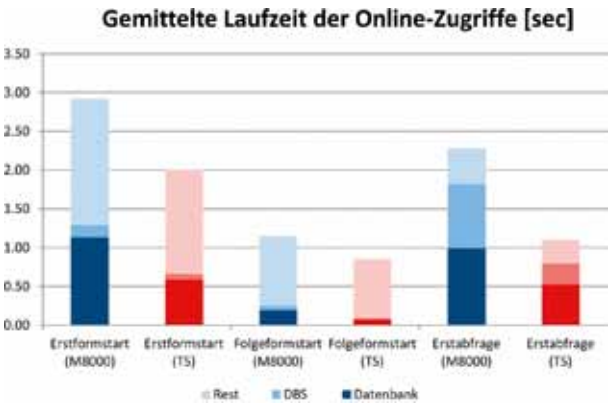

Abbildung 3: Gemittelte Laufzeit der Online-Zugriffe

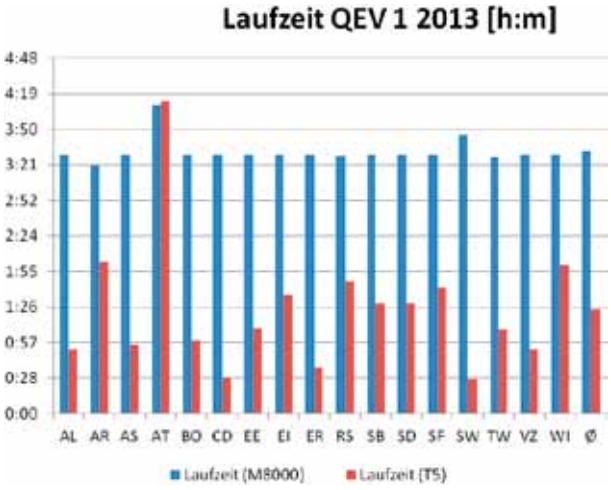

Abbildung 4: Laufzeit der Quartals-Endverarbeitung der einzelnen Mandanten

als eine der ersten Firmen in Europa eine T5-4. Da von Anfang an klar war, dass die T5-4 den finanziellen Rahmen des Kunden sprengen würde, wurde sie durch Deaktivierung von zwei der vier Prozessoren auf eine T5-2 redimensioniert. Im Folgenden sind die Tests beschrieben, die auf der M8000 sowie auf der T5-2 durchgeführt werden. Bei jedem Test wird kurz die Charakteristik des Tests erläutert, anschließend werden die Resultate und ein kurzes Fazit präsentiert. Die große Anzahl von Messwerten und Ergebnissen, die während der Tests entstanden sind, wurde für diesen Artikel auf ein absolutes Minimum eingedampft.

SPECjvm2008

Die SPECjvm2008-Testsuite ist ein synthetischer Test, der aus einem Mix von Code besteht. Im Mittelpunkt der Tests steht die Leistungsfähigkeit der Java Virtual Machine. Die Suite besteht aus verschiedenen Sub-Tests, die jeweils einen bestimmten Aspekt der JVM testen. Die genaue Beschreibung der Sub-Tests kann unter [1] nachgesehen werden.

Resultate: Der Output von SPECjvm2008, der sogenannte „Composite Result", ist ein Durchschnitt aller Sub-Tests. Größere Werte bedeuten ein besseres Resultat. In Tabelle 1 sind die Werte der Sub-Tests sowie der Composite Result aufgeführt (siehe Tabelle 1).

Fazit: Die guten Werte zeigen eindrücklich die Fähigkeit der T5, mit einer großen Anzahl von Threads umzugehen. Zu bemerken ist hierbei, dass diese Werte nicht als offizielle Benchmark-Werte bei „SPEC.org" eingereicht, sondern lediglich zu Vergleichszwecken in diesem Test verwendet wurden. Sie wurden jedoch in Übereinstimmung mit den Regeln für diesen Benchmark ermittelt [3].

Online-Zugriffe

Die Finnova-Software erlaubt es, die Laufzeit einer Aktion durch den Benutzer sehr detailliert zu messen. Dabei wird ermittelt, in welchem Teil der Verarbeitungskette (vom Client über das Netzwerk, die Service-Prozesse bis zur Datenbank) wie viel Zeit verbraucht wird. Der Test ist somit ein guter Mix aus Client- und Service-Prozessen (Java) und der Datenbank.

Resultate: In Abbildung 3 sind die Messwerte grafisch dargestellt. Die Laufzeit jeder Abfrage ist in drei Teile unterteilt: „Rest" (Client + Netzwerk, nicht vom Core-System abhängig), „DBS" (Service-Prozesse) und die reine Zeit in der Datenbank.

Fazit: Die Resultate der Online-Zugriffe lassen sich sehen. Die Verbesserung in diesem Bereich gegenüber der M8000 betragen zwischen 26 und 52 Prozent (siehe Abbildung 3).

Batch-Verarbeitung

Die Batch-Verarbeitung ist üblicherweise ein probates Mittel, um ein Core-System an seine Leistungsgrenze zu bringen. Über eine zentrale Oracle-Queue wird eine große Anzahl von Perl-Prozessen (Batch-Prozesse) mit sogenannten „Jobs" versorgt. Diese werden dann parallel in der Datenbank ausgeführt. Mit diesem Test wird die Fähigkeit der Maschine getestet, mit einer großen Anzahl aktiver Prozesse umzugehen. Als Testfall wurde die so-

Sub-Test	M8000	T5-2
Compiler	266.03	867.39
Compress	729.38	1222.98
Crypto	343.62	2481.84
Derby	600.51	3396.24
Mpegaudio	536.81	1571.27
Scismark.large	80.2	432.94
Scismark.small	720.6	1432.34
Serial	479	1851.12
Startup	7.4	15.78
Sunflow	259.8	722.59
Xml	841	2688.61
Composite result	283.8	937.96

Tabelle 1: Resultate von SPECjvm2008

Software-Update	M8000	T5-2
Laufzeit der Installation [h:m:s]	7:24:21	3:24:23

Tabelle 2: Laufzeit der Installation

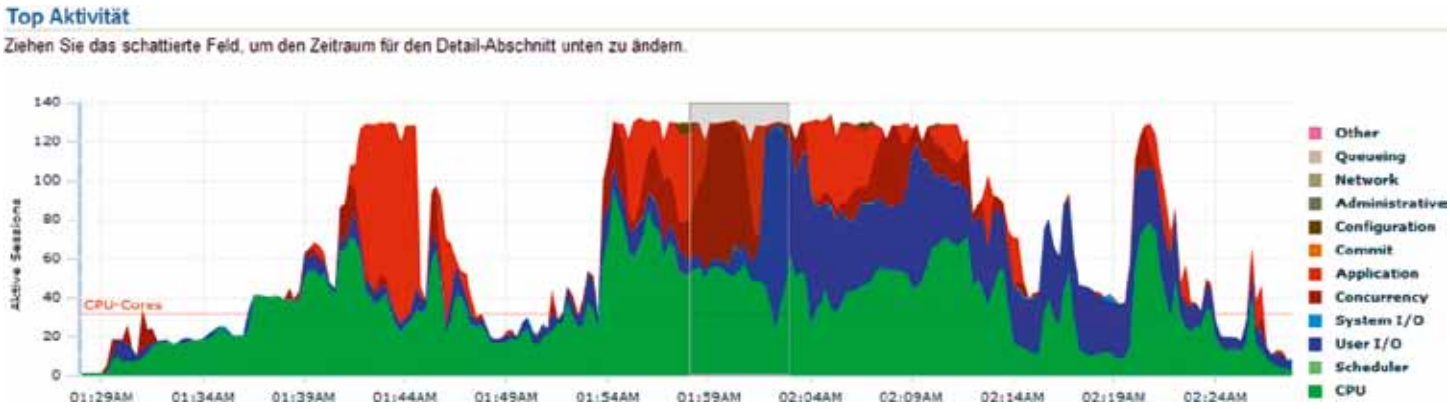

Abbildung 5: Ablauf der QEV (T5); Darstellung im OEM

genannte „Quartals-Endverarbeitung“ (QEV) per Ende März durchgeführt.

Resultate: Die QEV der meisten Mandanten war in weniger als der Hälfte der Zeit durchgeführt (siehe Abbildung 4). Die Verarbeitung erfolgte mit 128 parallelen Batch-Prozessen. Interessant ist, wie stark die Maschine dadurch belastet wurde (siehe Abbildungen 5 und 6). Hier zeigt die T5 ihre wahre Stärke, den Umgang mit Hunderten von parallelen Prozessen. Es gelang in keinem Testfall, die Maschine in eine Situation zu bringen, in der ein Arbeiten nicht mehr möglich war.

Update der Finnova-Software

Updates der Finnova-Software werden viermal im Jahr als sogenannte „Quartals-Patches“ ausgeliefert und auf der produktiven Umgebung eingespielt. Der große Teil der Updates betrifft Objekte in der Datenbank (Anpassungen an Tabellen, Laden von neuen Packages etc.). Die Updates werden seriell über eine einzige Session in die Datenbank geladen, daher gibt dieser Test die Veränderung der Single-Thread-Performance sehr gut wieder.

Als Resultat wird die gesamte Installations-Dauer gewertet. Die Installation ist in 41 Schritte unterteilt, deren Laufzeiten hier nicht weiter dargestellt werden (siehe Tabelle 2). Auch dieser Test bestätigt einmal mehr die Beobachtung, dass sich die Performance gegenüber der M8000 rund verdoppelt hat.

Fazit

Die T5-2 hat sich in den Tests als vollwertiger Ersatz für die M8000 erwiesen. Sie bringt im Bereich der Single-Thread-Performance rund die doppelte Leistung und zeigt im Bereich Multi-Processing/Multi-Threading Leistungen, die wirklich erstaunlich sind. Tabelle 3 zeigt die eingesetzten Systeme und Software.

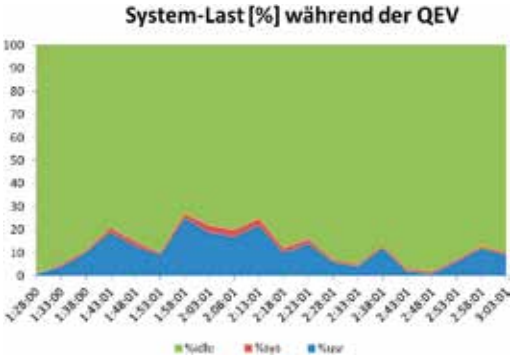

Abbildung 6: Systemlast (sar) während der QEV (T5)

Referenzen

[1] SPECjvm2008: http://www.spec.org/jvm2008/docs/RunRules.html

[2] Oracle SPARC T5-2, SPARC T5-4, SPARC T5-8, and SPARC T5-1B Server Architecture: http://www.oracle.com/technetwork/server-storage/sun-sparc-enterprise/documentation/o13-024-sparc-t5-architecture-1920540.pdf

[3] Offizielle Benchmark-Ergebnisse der T5-Systeme: http://www.oracle.com/us/solutions/performance-scalability/sun-sparc-enterprise-t-servers-078532.html

	M8000	T5-2
CPU	8 SPARC VII, 2520 MHz	2 SPARC-T5, 3600 MHz
RAM	160 GB	1 TB
OS-Version	Solaris 10	Solaris 10
Datenbank-Version	Oracle 11.2.0.2, PSU 6	Oracle 11.2.0.2, PSU 6
Java-Version	1.6.0_14-b08	1.6.0_14-b08
Perl-Version	5.8.8	5.8.8

Tabelle 3

Stefan Mühlebach
stefan.muehlebach@swisscom.ch

Stefan Hinker
stefan.hinker@oracle.com

Wer sich jemals mit dem inzwischen in die Jahre gekommenen Jumpstart Server von Solaris beschäftigt hat, kennt die damit verbundenen gewaltigen Nachteile und Probleme sehr gut.

Solaris 11 Deployment – willkommen in der Neuzeit

Ralf Germann, Trivadis AG

Selten funktioniert eine Installation von Anfang an wie gewollt. Unzureichende Mechanismen für Post-Installation-Tasks führen dazu, dass die meisten Administratoren eine eigene Lösung dafür entwickeln oder auf Produkte von Drittherstellern zurückgreifen müssen. An eine Einheit ist und war nicht zu denken. 300 Solaris-Server sollen mit einem neuen Software-Agenten bestückt werden, doch wo ist das Hilfsmittel dazu? Wie können wir unsere Software auf den neuesten Stand bringen?

Mit den neuen Deployment-Mechanismen von Solaris 11 hat Oracle einen Quantensprung in Richtung Neuzeit gemacht und das Betriebssystem mit Möglichkeiten ausgestattet, die von einem modernen Betriebssystem erwartet werden. Dieser Artikel zeigt die Unterschiede zwischen Solaris 10 und 11 hinsichtlich des Deployment und erklärt die neuen Mechanismen. Bei Jumpstart wird jeweils Bezug auf den puren Mechanismus ohne allfällige Hilfsmittel wie JET oder dergleichen genommen.

Die wesentlichen Unterschiede

Auf den ersten Blick lässt sich feststellen, dass das Deployment von Solaris 10 ein Sammelsurium diverser Komponenten und Tools ist. Einiges davon wurde nachträglich entwickelt und hinzugefügt, um etwas Erleichterung im Deployment-Alltag zu schaffen. Bestes Beispiel dafür ist das Jumpstart Enterprise Tool (JET), das ursprünglich als internes Projekt einiger Sun-Entwicklern gestartet wurde, die mit den Funktionen ihres hauseigenen Deployment nicht zufrieden waren.

Funktionen, die zusammengehören sollten, befinden sich in getrennten Komponenten. Dies hat zwangsweise zur Folge, dass sich die Administratoren in wesentlich mehr „Produkte“ und deren Eigenheiten sowie Kommando-Strukturen einarbeiten müssen. Zudem sind diese von ihrer Logik her meistens auch nicht konsistent aufgebaut.

Schon wesentlich angenehmer sieht es unter Solaris 11 aus. Die Deployment-Struktur wurde von Grund auf überarbeitet und konsolidiert. Mit durchgehendem ZFS und den Boot-Environments ist eine gute Basis entstanden, die auch bei Notfällen optimal unterstützt. Dies war unter Solaris 10 mit Live-Upgrade beziehungsweise den „lu“-Kommandos bereits möglich, allerdings nicht so komfortabel gelöst wie mit dem einheitlichen „bootadm(1M)“-Befehl unter Solaris 11.

Das Packaging System (IPS) erhielt endlich Komponenten, die viele Administratoren unter Solaris 10 stark vermisst haben (siehe Abbildung 1). Mit Repositories ist nun auch die zentrale Verwaltung von Software möglich. Die Installation und Deinstallation sowie das Aktualisieren von Produkten ist wesentlich vereinfacht. Wer sich mit gängigen Unix- und Linux-Paketmanagern auskennt, stellt mit Freude fest, dass die Bedienung sehr intuitiv und ohne große Einarbeitung funktioniert.

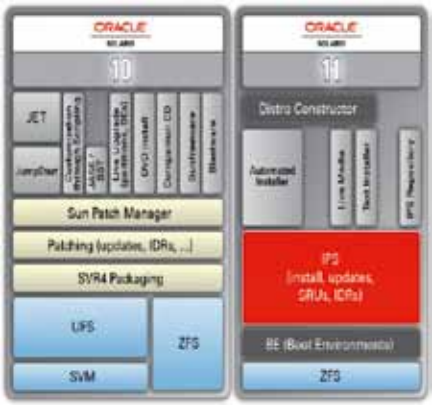

Abbildung 1: Vergleich der Deployment-Lösungen von Solaris 10 und Solaris 11 (Quelle: Oracle)

Auch die Basis-Installation mit dem Automated Installer ist um einiges angenehmer, übersichtlicher und einfacher geworden und arbeitet sehr gut mit den IPS-Repositories zusammen. Es wurden zudem Bestrebungen unternommen, Kommandos einheitlicher zu konzipieren. Aufbauend auf der Kommando-Struktur, die wir bereits von der ZFS-Verwaltung her kennen, ist dies Oracle auf den ersten Blick auch gelungen. Leider sind die Befehle zum Teil etwas verwirrend und es besteht Verwechslungsgefahr. Einige Funktionen, die bisher nicht zusammengeführt wurden, haben sehr ähnliche Namen.

Der Automated Installer

AI steht nicht für den englischen Begriff „Artificial Intelligence“, was so viel wie künstliche Intelligenz bedeutet. Soweit ist Oracle mit ihrem automatisierten Installationswerkzeug Automated Installer leider noch nicht. Auch wenn man neidlos zugeben muss, dass damit ein guter Job gemacht wurde und eine gewisse Intelligenz des Produkts nicht abzustreiten ist (siehe Abbildung 2).

Die AI-Software muss auf einem zentral zugänglichen Server installiert werden. Darin enthalten sind das IPS-Paket „installadm“ sowie dessen direkte Abhängigkeiten. Ein DHCP-Server und die Dienste TFTP und HTTP/S sind ebenfalls unerlässlich. Mittels sogenannter „Manifeste“ kann man seine Clients und die eigentliche Installation vorbereiten.

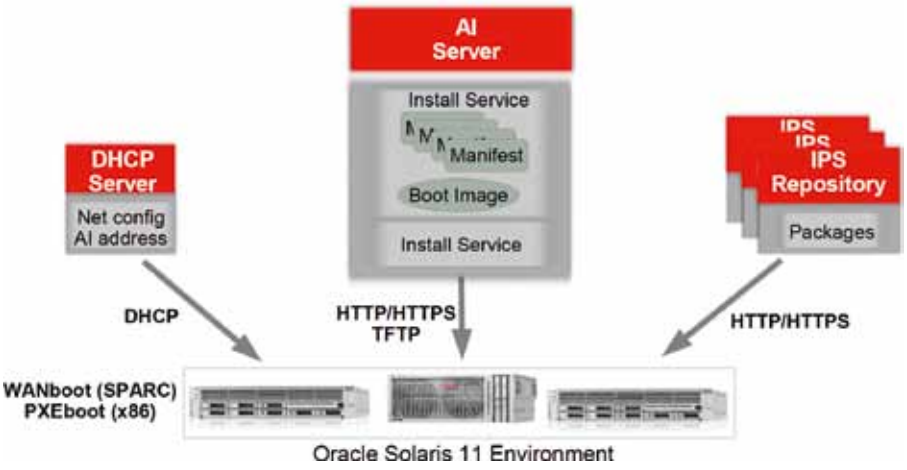

Abbildung 2: Funktionsübersicht des Automated Installer (Quelle: Oracle)

Manifeste sind vergleichbar mit Kickstart-Files aus den Linux-Distributionen. Partitionierungen, zusätzliche Repositories und Pakete sind nur einige der Konfigurationsmöglichkeiten innerhalb eines Manifests. Sollen auf dem physischen Server auch gleich eine oder mehrere Zone(n) mitinstalliert werden, ist das Manifest ebenfalls der richtige Ort, um dies zu definieren. Mit den mitgelieferten Bordmitteln von Jumpstart unter Solaris 10 war eine automatische Zonen-Installation nach der Client-Installation beispielsweise nicht möglich.

Ohne ein oder mehrere Repositories geht natürlich gar nichts. In diesen Repositories befinden sich alle benötigten System-Pakete sowie zusätzliche Produkte (von Drittherstellern oder eigene Pakete), die man gern mitinstallieren möchte. Die benötigten Komponenten können selbstverständlich auf einen oder mehrere Server mit unterschiedlichen Funktionen verteilt sein. Voraussetzung ist die reibungslose Kommunikation dieser Server untereinander.

Als zentrales Tool für die Verwaltung von AI-Tasks dient uns das Kommando „installadm(1M)“, das von der Struktur her auf der bereits von Solaris 10 bekannten ZFS-Verwaltung basiert. Beinahe alle benötigten Schritte lassen sich damit ausführen, was die Handhabung sehr vereinfacht. Dazu ein Beispiel für die Erstellung eines Clients (siehe Listing 1), ein Beispiel für die Erstellung und Zuweisung eines Manifests (siehe Listing 2) sowie die Anzeige der Zusammenfassung (siehe Listing 3).

Es besteht die Möglichkeit, Manifeste an einen oder mehrere Server zu knüpfen (siehe Beispiel). Die Steuerung übernehmen die sogenannten „Criteria“. Es können Werte wie „Architektur“, „MAC-Adressen“, „Memory“, „Plattformen“, „Hostnamen“ oder „IP-Adressen“ definiert werden. Es sind sowohl einzelne Werte als auch komplette Ranges möglich. Kennt man also seine zukünftige Umgebung und nimmt sich die notwendige Zeit für eine Planung und gute Struktur, spart man für die Umsetzung einiges an Zeit ein. Dies gilt insbesondere für sehr homogene Server-Umgebungen.

```
# installadm create-client -e
00:0C:29:31:AD:9F -n sol1111x86
```

Listing 1

```
# installadm export -n sol1111x86 -m orig_default -o /export/
manifest/TVD-SOL-11-CLIENT.xml
# installadm create-manifest -n sol1111x86 -f /export/manifest/
TVD-SOL-11-CLIENT.xml -m TVD-SOL-11-CLIENT -c mac="00:0C:29:31:AD:9F"
```

Listing 2

```
# installadm list -n sol1111x86 -m -c -p

Service Name Client Address     Arch   Image Path
------------ --------------     ----   ----------
sol1111x86   00:0C:29:31:AD:9F  i386   /install/sol1111x86

Service/Manifest Name    Status   Criteria
---------------------    ------   --------
sol1111x86
   TVD-SOL-11-CLIENT              mac      = 00:0C:29:31:AD:9F
   orig_default          Default  None

Service/Profile Name     Criteria
--------------------     --------
sol1111x86
   TVD-SOL-11-CLIENT     mac      = 00:0C:29:31:AD:9F
```

Listing 3

Mit einem optionalen „Sysconfig“-Profil lassen sich die nach einer Installation notwendigen Einstellungen wie „Benutzer“, „Passwörter“, „Zeitzonen“, „Hostnamen“, „Netzwerk-Konfigurationen“ etc. im Voraus definieren (siehe Abbildung 3). Dies hat den Vorteil, dass man sich nach der Installation den Gang zur Konsole spart. Der Befehl „sysconfig create-profile“ ruft ein interaktives Menü auf, das der Eingabemaske einer manuellen Text-Installation entspricht und die Einstellungen in ein vordefiniertes File speichert. Listing 4 zeigt ein Beispiel für die Erstellung eines Sysconfig-Profils.

Für Konfigurationsaufgaben, die sich nicht während der Installation erledigen lassen, eignet sich der sogenannte „First Boot SMF Service“. Dieser ist optional und entspricht ungefähr der „Post Install Script“-Funktion unter Jumpstart. Nicht unerwähnt sei das Tool „js2ai(1M)“. Wer seine alten „sysidcfg“-Dateien in AI-Manifeste umwandeln möchte, hat damit das geeignete Werkzeug. Bei komplizierteren Gebilden wird nicht immer zu 100 Prozent die Nacharbeit erspart, jedoch ist diese nach der Umwandlung in das XML-Format um einiges einfacher, fehlerfreier und komfortabler als bei der manuellen Übertragung in eine

```
# sysconfig create-profile -o /export/profiles/TVD-SOL-11-CLIENT.xml
```

Listing 4: Beispiel für die Erstellung eines Sysconfig-Profils

Datei. In Tabelle 1 sind die einzelnen Aufgaben zwischen Jumpstart und Automated Installer direkt miteinander verglichen. Nebenbei bemerkt: Installationen mit dem Automated Installer lassen sich sowohl für SPARC- als auch für x86-Systeme durchführen, egal, auf welcher Architektur der AI-Server und dessen Komponenten basieren.

IPS – Paketmanagement unter Solaris 11

Wer sich zum Beispiel mit dem Paketmanager „yum" aus den Red-Hat-Derivaten oder anderen ähnlichen Verwaltungs-Werkzeugen auskennt, wird sich mit dem Imaging Packaging System (IPS) von Solaris 11 schnell zurecht finden. Mit einem oder mehreren Software-Repositories, die entweder direkt übers Internet bezogen oder auf Servern in der eigenen Umgebung abgelegt werden können, lassen sich sehr einfach Software-Komponenten installieren, deinstallieren oder aktualisieren. Da es sich um ein zentrales Paket-Management handelt, ist die Belieferung mehrerer Server kein Problem mehr. Die Verwendung von einem oder mehreren Spiegel-Servern ist ebenfalls möglich.
Auf grafischen Systemen kann direkt mit dem Paketmanager (siehe Abbildung 4) oder dem webbasierten Tool gearbeitet werden. Server ohne grafische Komponenten benutzen das „pkg(1M)"-Kommando für ihre Tasks. Pakete lassen sich versionieren und die benötigten Abhängigkeiten werden automatisch mit installiert.

Standardmäßig wird das Public-Repository von Oracle vorkonfiguriert (siehe http://pkg.oracle.com/solaris/release/en/index.shtml), Vertragskunden erhalten für ihre Updates ein separates Repository (siehe https://pkg.oracle.com/solaris/support).

Der Distribution Constructor

Mit dem Distribution Constructor (siehe Abbildung 5) lassen sich angepasste Installations-Images oder virtuelle Maschinen erstellen. Die Oracle-Solaris-Entwickler benutzen dieses Tool übrigens auch, um ihre Produkte zu erstellen.

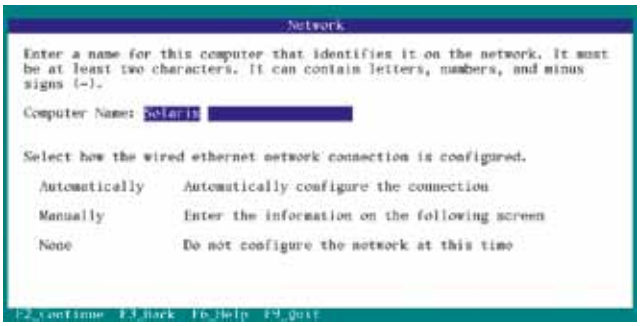

Abbildung 3: Einstiegsbild des System Configuration Tool

Mithilfe von spezifizierten Parametern in den Distribution-Constructor-Manifesten (XML-Format) erstellt das Tool ISO-Images oder Images für virtuelle Maschinen. Basierend auf x86-ISO-Images kann auch ein bootbares Image für USB-Geräte erzeugt werden. Beim Erstellen von Images besteht die Möglichkeit, den Prozess an verschiedenen Stellen zu stoppen und später wieder zu starten. Für Überprüfungen und Debugging ist dies unerlässlich. Diese Funktion nennt sich „checkpointing".

Das Boot Environment

Obwohl die Technologie nur entfernt etwas mit dem Deployment zu tun hat, wird das Thema trotzdem kurz gestreift. Unter Solaris 10 waren Boot Environments ebenfalls möglich. Ohne einen „ZFS-rpool2" musste man allerdings einige Einschränkungen in Kauf nehmen, die unter Solaris 11 mit dem durchgängigen ZFS nicht mehr vorhanden sind (siehe Abbildung 6).

Mit dem zentralisierten „bootadm(1M)"-Befehl lassen sich Boot Environments (BE) schnell und unkompliziert erstellen. Weil die Technologie mit ZFS-Funktionen arbeitet, wird bei neuen BEs meist nur wenig zusätzlicher Speicherplatz für ein Delta belegt.

Wer Updates installieren möchte, aber keine große Downtime dafür zur Verfügung hat, der wird Boot Environments ebenfalls sehr mögen. Es besteht die Möglichkeit, Updates/Installationen in einem neuen BE auszuführen, das erst zum Zeitpunkt der Umschaltung aktiv wird. Somit ist das produktive System von der Installation nicht betroffen, solange man nicht das neue BE aktiv nimmt. Nebenbei bemerkt: Boot Environments sind auch in Zonen möglich.

Neuerungen unter Solaris 11.1

Es wird nicht auf alle Neuerungen eingegangen, die das Update 11.1 enthält, sondern nur auf die wesentlichen Änderungen im Deployment. Eine der spürbarsten Verbesserungen ist ganz

Aufgabe	Jumpstart	Automated Installer
Installations-Server bereitstellen	Es wird das „setup_install_server(1M)"- Kommando benötigt	Mittels „installadm create-service" wird ein neuer Installationsdienst bereitgestellt
Clients einer Installation hinzufügen	Durch das Kommando „add_install_client(1M)"	Geht ganz einfach mit „installadm create-client"
Installationsbedingungen definieren	Mit den sogenannten „Profile"-Dateien realisierbar	Über ein AI-Manifest-File
Spezifizieren von Client-Regeln	Wird mit den „Rules"-Dateien realisiert, die einer „Profile-Datei" zugewiesen werden müssen	Mit dem Kommando „installadm set-criteria" lassen sich Clients mit AI-Manifesten verknüpfen
Konfigurationen nach der Installation ausführen	Es werden das „sysidcfg"-File sowie „finish"-Skripte benötigt	Mittels der Manifeste und eines First-Boot-SMF-Service umsetzbar

Tabelle 1

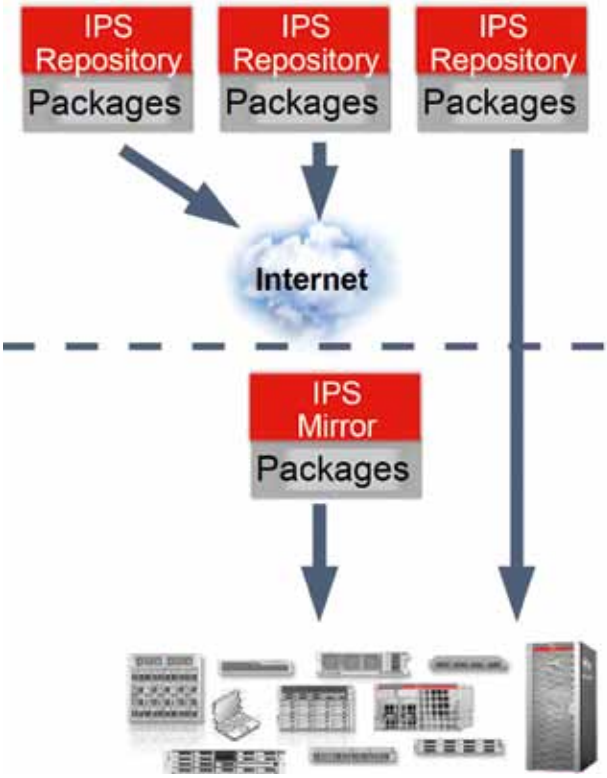

Abbildung 4: Überblick Paket-Management (Quelle: Oracle)

klar der Paket-Manager (IPS). Oracle hat diesem einen gewaltigen Performance-Schub verpasst. Unter 11.0 dauerten zum Beispiel Installationen und Updates unnachvollziehbar lange, was viele Administratoren als störend empfunden haben. Mit 11.1 ist die Performance deutlich akzeptabler.

Die neuen Sub-Kommandos „update-service", „update-profile" und „set-service" des „installadm(1M)"-Befehls vereinfachen die Anpassung von Installations-Services und bieten Administratoren mehr Flexibilität. Neu ist auch die direkte Installation mit der „interactive text"-Methode und dem „live media installer" auf iSCSI-LUNs. Die Fortschrittsausgabe während des Deployment ist viel ausführlicher und zuverlässiger geworden.

Wer nicht gern mit XML-Files arbeitet, dem wird die nächste Neuerung sicherlich gefallen. Mit dem Kommando „svcbundle(1M)" lassen sich neue SMF-Manifeste und -Profile ohne XML-Kenntnisse erstellen. Auch beim Update von Zonen hat sich in Sachen „Performance" gewaltig etwas geändert. Dies ist vor allem der Möglichkeit zu verdanken, mehrere Zonen auf einem Server gleichzeitig aktualisieren zu können. Gemäß Oracle ist das Updaten von 20 Zonen somit vier Mal schneller als bisher. Zudem wurde die Zeit für eine Zonen-Installation um 27 Prozent verbessert.

Wer sich über weitere Neuerungen und Verbesserungen unter Solaris 11.1 informieren möchte, dem kann das offizielle Dokument von Oracle unter http://www.oracle.com/technetwork/server-storage/solaris11/documentation/solaris11-1-whatsnew-1732377.pdf weiterhelfen.

Fazit

Oracle hat mit dem Deployment unter Solaris 11 sehr viel richtig gemacht. Es scheint, als wäre der Neuaufbau ein guter Schritt, der sich zwangsweise durch die vielen Abgänge ergab. Vieles wirkt nun modern, zeitgemäß und durchdacht. Wenn Oracle sich nun die Zeit nimmt und weitere Optimierungen am Paket-Manager vornimmt oder die Kommando-Struktur noch mehr zu vereinfachen beziehungsweise zu konsolidieren versucht, kann man schlussendlich mit gutem Gewissen sagen, dass Solaris – vor allem was das Deployment angeht – endlich in der Neuzeit angekommen ist.

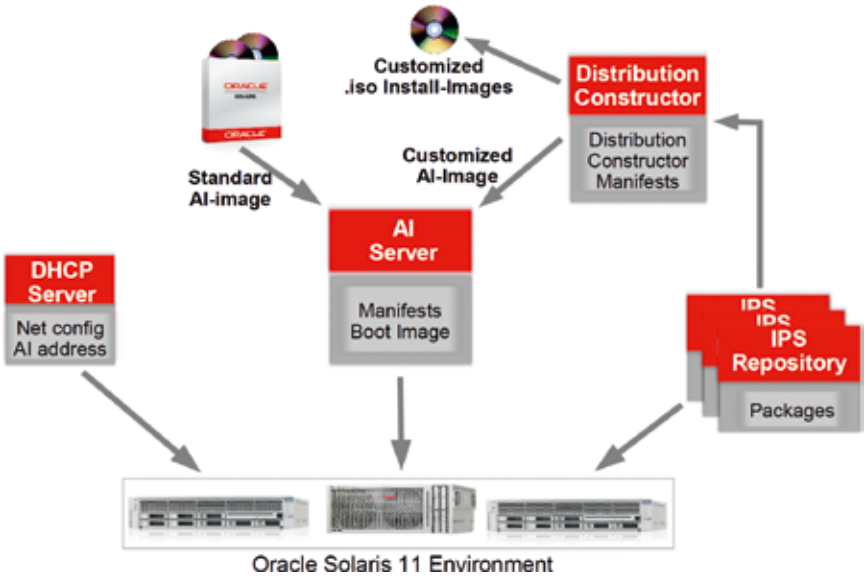

Abbildung 5: Überblick über den Distribution Constructor (Quelle: Oracle)

Abbildung 6: Boot-Environments-Beispiel (Quelle: Oracle)

Ralf Germann
ralf.germann@trivadis.com

Zur Nutzung und Verwaltung direkt oder über ein Storage-Area-Network (SAN) angebundener Platten für eine Oracle-Datenbank wird ASM auf Raw-/Block-Devices empfohlen, was von vornherein auf die I/O-Charakteristika der Datenbank abgestimmt ist. Nicht zuletzt aus administrativen Überlegungen heraus sind aber weiterhin auch Dateisysteme populär, wobei deren Voreinstellungen meist nicht den spezifischen I/O-Charakteristika von Datenbanken entsprechen, die sich ja von sonstigen Lastprofilen signifikant unterscheiden.

Best Practices für Datenbanken auf ZFS

Franz Haberhauer, ORACLE Deutschland B.V. & Co. KG

ZFS ist das moderne, innovative lokale Dateisystem von Oracle Solaris und kann als gängiges Dateisystem von Datenbanken genutzt werden. Darüber hinaus verfügt es über effiziente Snapshot- und Cloning-Funktionen. Andererseits dient es als internes Dateisystem der Network-Attached-Storage-Familie (NAS) von Oracle – der ZFS Storage Appliances (SA) – und wird damit indirekt für Datenbanken auf „(d)NFS“ genutzt. Der Artikel gibt Konfigurationshinweise und geht auf deren technische Hintergründe ein. Abschließend werden einige Tools angesprochen, die auf der ZFS SA verfügbar sind.

Oracle Solaris ZFS

Vor mittlerweile gut sieben Jahren wurde mit ZFS in Solaris 10 ein neues lokales Dateisystem integriert, das die klassischen Aufgaben eines Dateisystems – Daten ohne großen administrativen Aufwand schnell und effizient zu schreiben, sicher zu speichern und wieder auszulesen – mit einer innovativen Architektur grundsätzlich anders löste als traditionelle Dateisysteme [1]. Dateisystem und Volume-Management sind integriert, was die starken Mechanismen zur Sicherung der Datenintegrität ermöglicht, die ZFS besonders auszeichnen.

Stille Datenkorruption, teilweise durch transiente Hardware-Defekte oder Software-Fehler entstanden, ist ein reales Risiko, das angesichts der heutigen Datenvolumina kein unwahrscheinliches Ereignis mehr ist [2]. Über konsequent genutzte Prüfsummen in baumstrukturierten Dateien und Meta-Informationen kann ZFS korrupte Blöcke erkennen und gegebenenfalls aus vorhandenen redundanten Daten (Spiegel, RAID-Z, RAID-Z2) den korrekten Inhalt rekonstruieren. Die konsequente Nutzung des Copy-on-Write-Paradigmas und des Transaktionskonzepts sorgen dafür, dass das On-Disk-Image auch nach einem Systemabbruch immer konsistent ist, wobei Write-Caches moderner Platten und Speichersysteme genutzt werden können. ZFS erlaubt es zudem, relativ langsame Platten mit hoher Kapazität mit einem Level-2-Cache (L2ARC) aus schnelleren Laufwerken – insbesondere SSDs – für Anwendungen transparent zu leistungsfähigen „hybriden Storage Pools“ zu kombinieren. Leichtgewichtige Snapshots mit minimalem Overhead und der Möglichkeit, daraus beschreibbare Clones zu machen, Kompression und in Solaris 11 Verschlüsselung und Deduplikation sind nur einige der umfangreichen Funktionalitäten, die ZFS so attraktiv machen.

Als Hintergrund für die folgenden Empfehlungen zur Nutzung für Datenbanken ist von besonderem Interesse, wie ZFS Daten ausschreibt. ZFS überschreibt Daten nicht unmittelbar, sondern folgt dem Prinzip „Copy on write“. Geänderte Blöcke werden wie neue Blöcke an eine neue Stelle auf der Platte geschrieben. Dabei werden variable Blockgrößen von bis zu 128 KB verwendet und wahlfreie Schreibzugriffe in effiziente sequenzielle Schreiboperationen umgewandelt, was den Zugriffs-

Charakteristika moderner Platten entgegenkommt.

Änderungen werden von den Blättern zum Wurzelknoten, dem sogenannten „Uberblock“, propagiert. Das Ausschreiben des Uberblock bildet den atomaren Zustandsübergang, der das Ende einer Transaktion definiert. Aus Performance-Gründen werden Änderungen zu Transaktionsgruppen zusammengefasst. Periodisch (voreingestellt sind fünf Sekunden) wird eine solche Transaktionsgruppe abgeschlossen und alle geänderten Blöcke werden eine Ebene nach der anderen bis zum Uberblock ausgeschrieben, wobei die übergeordneten Knoten jeweils Prüfsummen für die darunterliegenden Knoten enthalten.

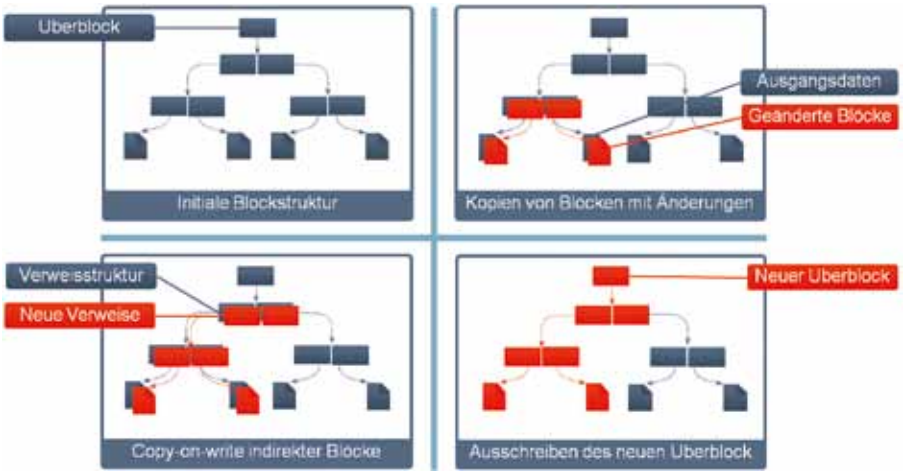

Abbildung 1: Copy-on-write: Update-Operation in ZFS

Währenddessen laufen weitere Änderungen in der nächsten Transaktionsgruppe. Wird der ersetzte Wurzelknoten nicht freigegeben, sondern sichtbar gemacht, erhält man mit minimalem Aufwand eine Snapshot-Funktionalität. Um synchrone Operationen, wie sie in DBMS oder von NFS-Servern genutzt werden, schnell abwickeln zu können, wird zudem ein Intent Log (ZIL) genutzt, über den Änderungen sofort auf stabilen Speichern gesichert werden, bevor sie sich mit dem Ausschreiben des Uberblock der Transaktionsgruppe auf den Datenplatten selbst manifestieren. Für eine optimale Performance kann man den ZIL von den Datenplatten separieren und auf dedizierten, besonders schnellen stabilen Speichern ablegen, wie etwa Flash-Karten oder SSDs, die zum Schutz vor Datenverlusten ebenfalls gespiegelt sein sollten.

Die I/O-Charakteristika der Oracle -Datenbank unterscheiden sich von denen sonstiger Lasten auf Dateisystemen. Während normale Dateien vom Dateisystem im Hauptspeicher gepuffert sind und – oft kleine – I/Os der Anwendungen vom Dateisystem asynchron beim Ausschreiben auf Platte in größere I/Os kumuliert oder beim Lesen vorausgelesen werden, puffert die Oracle-Datenbank Daten selbst in der SGA und schreibt sie bei Bedarf in einer definierten Blockgröße synchron aus. Dabei sind nicht alle I/Os gleichermaßen kritisch für die Performance. Die auf der Dateisystem-Ebene synchronen I/Os in die Daten- und Index-Bereiche sind aus Datenbank-Sicht asynchron, nur die Zeit für das Ausschreiben des Commit-Record in den Redo Log geht unmittelbar in die Antwortzeit einer Transaktion ein.

Bei älteren Dateisystemen wie Solaris UFS wurden einige Optimierungen insbesondere für Datenbanken in einer Option unter dem Begriff „Direkt I/O“ zusammengefasst, mit der Datenbank-Dateien optimiert geöffnet oder Dateisysteme gemountet werden können. Dadurch wird insbesondere beim UFS ein Single-Writer-Lock zur Sicherung der POSIX-Semantik, die für Datenbank-Dateien nicht relevant ist, zwecks Erhöhung des Durchsatzes umgangen und die (Doppel-)Pufferung im Dateisystem deaktiviert.

ZFS adressiert parallele I/Os POSIX-konform und performant durch Byte-Range-Locking. Die Pufferung im ZFS-Puffer des Hauptspeichers, dem ARC, kann gezielt durch ein Dataset-Property „primarycache“ mit den Werten „all“, „none“ oder „metadata“ gesteuert werden. Bei ZFS können in einem Storage Pool, der einem „logical Volume“ eines traditionellen Volume Manager entspricht, mehrere Datasets („Dateisysteme“) angelegt werden, die unterschiedliche Properties haben können.

Best Practices für Oracle-Datenbanken

Eine wesentliche Einstellung ist die Anpassung der ZFS „recordsize“ für Tabellen und Indizes an die Blockgröße der Datenbank („db_block_size, allerdings nicht kleiner als die Seitengröße des Servers, bei SPARC 8 KB). Für andere Bereiche (Redo Logs, Undo, Temp und Archived Redo Logs) passt hingegen die Voreinstellung von 128 KB. Um für diese Bereiche jeweils spezifische Parameter setzen zu können, sollten sie in separate Datasets gelegt werden.

„recordsize“ definiert man vorzugsweise gleich beim Anlegen eines Dataset, da eine Änderung nur für danach angelegte Dateien zum Tragen kommt. Um sie für eine bereits vorhandene Datei zu ändern, kann man diese kopieren. Durch Kopieren kann man auch einen Seiteneffekt von Copy-on-write wieder eliminieren: Wegen der wahlfreien Änderung einzelner Blöcke wird eine physische Nachbarschaft auf der Platte mit der Zeit aufgelöst, da ja geänderte Blöcke nicht an die ursprüngliche Stelle zurückgeschrieben, sondern an eine neue ausgeschrieben werden, was einerseits das Schreiben effizienter macht, andererseits bei nachfolgenden Tabellen-Scans dann zu wahlfreien statt sequenziellen Zugriffen führen kann, denn eventuell wird aus einem großen logischen Read der Datenbank eine Anzahl kleinerer IOs auf der ZFS-Ebene. Dieser Effekt macht sich auch bei Sicherungen über RMAN bemerkbar. Bei hohen Änderungsraten kann es angesichts der Laufzeit von Backups angebracht sein, auf „recordsize“ von 32 KB oder 64 KB zu gehen.

Ein weiterer Faktor, den man berücksichtigen sollte, kann eine minimale interne I/O-Größe von Speichersystemen sein. Diese resultiert meist aus der

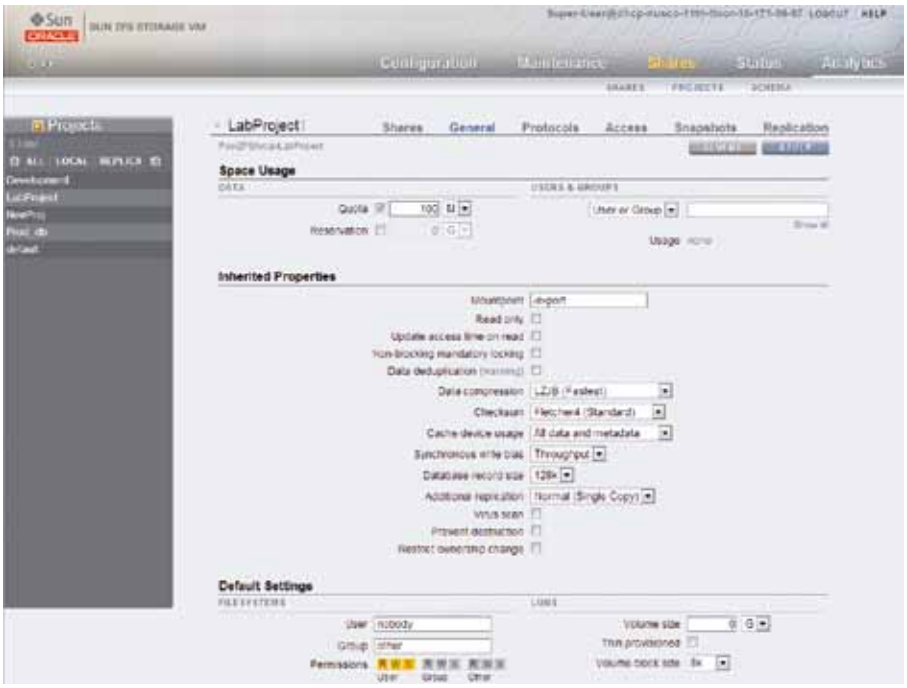

Abbildung 2: Web-GUI der ZFS Storage Appliance

internen Verwendung von RAID-Schemata. Generell ist unter Performance-Gesichtspunkten für die Datenbank die Spiegelung gegenüber RAID-5/-6 oder auch RAID-Z/-Z2 vorzuziehen – und zwar im Hinblick auf die stundenlangen Resilvering-/Wiederherstellungszeiten nach dem Ausfall einer heutigen Platte mit mehreren Terabyte Kapazität vorzugsweise mit mehrfacher Redundanz, um das Risiko eines weiteren Ausfalls in dieser Phase abzudecken.

Spiegel haben den Vorteil, dass durch sie die maximale Schreib-Rate (IOPS) erhalten bleibt und durch (Mehrfach-)Spiegel die maximale Lese-Rate entsprechend vervielfacht wird, was bei RAID-Verbünden nicht in gleichem Maße der Fall ist, wobei sich insbesondere der Aspekt des Schreibens bei Hardware-RAIDs mit nichtflüchtigen Caches (NVRAM) relativiert.

Bei Copy-on-write ist die Preallocation von Dateien nicht sinnvoll – ausreichend Platz lässt sich besser über Quotas und insbesondere Reservierungen im ZFS freihalten. Generell sollte für eine optimale Performance die Kapazität von Dateisystemen nicht voll ausgenutzt werden, da bei hohem Füllgrad auf Algorithmen umgeschwenkt wird, die Speichereffizienz über Performance stellen. Während traditionell für Datenbanken auf ZFS empfohlen wurde, unter 80 Prozent zu bleiben [3, 4], wurde für Solaris 11.1 nach Optimierungen diese Empfehlung auf 90 Prozent angehoben [5].

Zudem sollte für Anwendungen, die wie die Oracle-Datenbank sehr viel Hauptspeicher nutzen, der ZFS-Puffer (ARC) von vornherein entsprechend über den Parameter „zfs_arc_max im / etc/system“ beschränkt werden, um zu vermeiden, dass ZFS beim (Neu-)Start der Datenbank erst Speicher freigeben muss oder im Betrieb darum konkurriert. Wichtig ist, dass er ausreichend groß bleibt für die Metadaten des Workingsets (etwa 1,5 Prozent der aktiven Daten bei ZFS „recordsize“ von 8 KB, bei einer größeren entsprechend weniger). Hier ein Beispiel für die Beschränkung auf 2GB im „/etc/system“: §set zfs:zfs_arc_max=2147483648“ oder „set zfs:zfs_arc_max=0x80000000“. Für die Datasets der Undo-Daten und Archived Redo Logs kann man über „primarycache=metadata“ die Puffe-rung im ARC gleich auf die Metadaten beschränken.

Da Datenbank-I/Os für das Dateisystem synchron sind, kommt dem ZFS Intent Log (ZIL) eine besondere Bedeutung zu, weshalb er auf ein separiertes Log-Device aus (gespiegelten) SSDs gelegt werden sollte. Der ZIL kennt zwei Modi, die über den Parameter „logbias“ gesteuert werden. Während bei „latency“ (Voreinstellung) die Daten aus dem ARC zunächst in den ZIL und mit der Transaktionsgruppe auf die Datenplatten geschrieben werden, werden im Modus „throughput“ die Daten direkt auf die Datenplatten geschrieben und nur Verweise darauf in den ZIL (die dann mit dem Ausschreiben des neuen Uberblock obsolet werden). Da hier weniger in den ZIL geschrieben wird, lässt sich mit einer etwas höheren Latenz der einzelnen Writes ein größerer Gesamtdurchsatz erreichen – und zudem Bandbreite für wirklich Performance-kritische Writes bereitstellen, nämlich jene in die Redo-Logs. Insofern lautet die Empfehlung, für alle Bereiche außer den Redo Logs „logbias=throughput“ zu setzen.

Die automatische und transparente Kompression der Archived Redo Logs ist eine weitere Möglichkeit, spezifische Funktionalitäten von ZFS zu nutzen. Kompression spart übrigens nicht nur Plattenplatz, sie kann auch die Anzahl von I/Os reduzieren und sich darüber positiv auf die Performance auswirken. Kompression wird durch Setzen des Dataset-Property „compression=on“ aktiviert. In Tabelle 1 sind noch einmal die Empfehlungen tabellarisch zusammengefasst

Diese Empfehlungen, die in einem Best-Practices-Dokument auf „solarisinternals.com“ gesammelt wurden, haben inzwischen Eingang in das „So-

Dataset	**ZFS recordsize**	**logbias**	**primarycache**
Daten	**db_block_size*)**	**throughput**	all
Indizes	**db_block_size**	**throughput**	all
Redo	128KB	latency	all
Undo	**db_block_size*)**	**throughput**	**metadata**
Temp	128KB	**throughput**	all
Archive	128KB **compression=on**	**throughput**	**metadata**

*) Datawarehouse-artige Anwendungen, Large Objects (LOBs): 128KB (Default)

Tabelle 1: Abweichungen von den Voreinstellungen sind fett dargestellt

laris Tunable Parameters Reference Manual“ [4, 5] und in ein im September 2012 aktualisiertes Whitepaper [3] gefunden, die jeweils ausführliche Beispiele enthalten. [3] enthält zudem einen Abschnitt mit Empfehlungen zur Formatierung von LUNs und Hinweisen zur Nutzung von Caches in Platten und Plattensystemen, die für ZFS ganz allgemein gelten.

MySQL

Für MySQL wird ebenfalls empfohlen, „recordsize“ an die Blockgröße der Storage Engine anzupassen, bei InnoDB 16 KB für die Data Files und die Voreinstellung von 128 KB für die Log Files, wobei Data und Log Files in separate Pools gelegt werden sollten [4, 5].

ZFS Storage Appliance

Auch bei der Nutzung von Datenbanken über „(d)NFS“ spielt ZFS unter Umständen eine Rolle, nämlich als internes Dateisystem der ZFS Storage Appliances (SA), einer Produktlinie leistungsfähiger NAS-Systeme mit einem Einstiegssystem mit elf Platten bis hin zu hochverfügbaren Konfigurationen und einer Kapazität von mehreren Petabyte, jeweils mit integriertem Flash/SSDs für separierte ZFS Intent Logs [6]. Die Systeme zeichnen sich durch ein im Vergleich zum Wettbewerb sehr gutes Preis-Leistungs-Verhältnis aus sowie durch eine intuitive, webbasierte Administrationsoberfläche, in der sich die oben erläuterten Parameter (ZFS Properties) wiederfinden – teilweise unter logischen Bezeichnungen: „logbias“ als „Synchronous write bias“ oder „recordsize“ als „Database record size“ (siehe Abbildung 2).

Eine weitere interessante Funktionalität der ZFS SA ist Analytics. Sie erlaubt tiefgehende Performance-Analysen – sehr fein granuliert bis hin zu Offsets in einzelnen Dateien und mit einer grafischen Visualisierung über der Zeitachse (siehe Abbildung 3).

Die ZFS SA wird auch als integrierter Fileserver in den Engineered Systems Exalogic und SPARC SuperCluster verbaut und ist als ZFS Backup Appliance als leistungsstarke Backup-Lösung für die Engineered Systems verfügbar. Zwei Whitepaper [7, 8] beschreiben ausführlich Best Practices der Nutzung

Abbildung 3: I/O-Profil eines „CREATE TABLESPACE” in ZFS Storage Appliance Analytics

von ZFS SA für Backup und Recovery der Exadata, wobei vieles davon auf die Sicherung von Datenbanken auf anderen Servern übertragbar ist.

Wie gesagt sind mit ZFS sehr effiziente Snapshots und Clones möglich, was sich hervorragend nutzen lässt, um etwa schnell und platzsparend Entwicklungs- oder Test-Datenbanken anzulegen [9]. Mit dem neuen Snap Management Utility [10] kann das Erzeugen von Datenbank-Snapshots und -Clones in Verbindung mit ZFS Storage Appliances weitgehend automatisiert und über eine webbasierte Oberfläche verwaltet werden. Weitere Einsatzfelder ergeben sich daraus, dass auf der ZFS SA Hybrid Columnar Compression (HCC) auch außerhalb der Exadata nutzbar ist.

Allein in der Oracle IT und in den Cloud Datacentern werden inzwischen ZFS Storage Appliances mit einer Gesamtkapazität von über 200 Petabyte eingesetzt. In [11] sind daraus Anwendungsszenarien beschrieben, in denen Best Practices exemplarisch umgesetzt sind.

Literaturhinweise

[1] Oracle Solaris ZFS Technology: http://www.oracle.com/technetwork/server-storage/solaris11/technologies/zfs-338092.html

[2] Robin Harris: CERN's Data Corruption Research, 17.09.2007: http://storagemojo.com/2007/09/19/cerns-data-corruption-research

[3] Cyndi Swearingen, Roch Bourbonnais, Alain Chéreau: Configuring ZFS for an Oracle Database, Oracle White Paper, September 2012: http://www.oracle.com/technetwork/server-storage/solaris10/config-solaris-zfs-wp-167894.pdf

[4] Oracle Solaris 10 1/13 1 Tunable Parameters Reference Manual – Tuning ZFS for Database Products: http://docs.oracle.com/cd/E26505_01/html/E37386/chapterzfs-db1.html#scrolltoc

[5] Oracle Solaris 11.1 Tunable Parameters Reference Manual – Tuning ZFS for Database Products: http://docs.oracle.com/cd/E26502_01/html/E29022/chapterzfs-db1.html#scrolltoc

[6] Oracle Sun ZFS Storage Appliances: http://www.oracle.com/us/products/servers-storage/storage/nas

[7] Protecting Oracle Exadata with the Sun ZFS Storage Appliance: Configuration Best Practices, Updated Oracle White Paper, March 2013: http://www.oracle.com/technetwork/server-storage/sun-unified-storage/documentation/zfssa-exadata-rman-v1-3-1926901.pdf

[8] Backup and Recovery Performance and Best Practices using the Sun ZFS Storage Appliance with the Oracle Exadata Database Machine, Oracle White Paper April 2012: http://www.oracle.com/technetwork/database/features/availability/maa-wp-dbm-zfs-backup-1593252.pdf

[9] Database Cloning using Oracle Sun ZFS Storage Appliance and Oracle Data Guard, Oracle White Paper, December 2011: http://www.oracle.com/technetwork/database/features/availability/maa-db-clone-szfssa-172997.pdf

[10] Snap Management Utility for Oracle Database: http://www.oracle.com/us/products/servers-storage/storage/nas/snap

[11] Sun ZFS Storage Appliance and Oracle IT: Use Cases and Benefits, Oracle White Paper, September 2012: http://www.oracle.com/us/products/servers-storage/storage/nas/resources/zfssaoracle-it-whitepaper-100812gc-1875031.pdf

Franz Haberhauer
franz.haberhauer@oracle.com

Wenn sich ein Solaris-System-Administrator an die Worte „don't change a running system" erinnert, ist es meistens schon zu spät. Eine einfache Umkehr ist nicht immer möglich. Der Autor hat routinemäßig ein Solaris „CPU OS Patchset" eingespielt und ist danach in Teufels Küche geraten.

Was sie von Oracle über ZFS nicht hören werden

Roman Gächter, Trivadis AG

Dieser Artikel beschreibt die Erfahrungen, die der Autor bei einem Kunden mit dem „Zetabyte File System" (ZFS) gemacht hat. ZFS wurde von Sun Microsystems für Solaris entwickelt und gehört heute Oracle. Der Kunde muss anonym bleiben und kann nicht erwähnt werden.

ZFS ist unbestritten ein außerordentlich gutes Produkt. Eine der innovativsten Erfindungen in diesem Bereich der letzten Jahre. Auch wenn das Aufsetzen von ZFS aus der Sicht des Administrators unglaublich leicht vonstattengeht, darf nicht vergessen werden, dass in einem Enterprise-Datenbank-Umfeld zusätzliches Tuning notwendig ist und man von Anfang an ein gutes Konzept für das Storage-Layout ausarbeiten muss.

Eine Banken-Plattform, die seit knapp einem Jahr in Betrieb war und problemlos lief, bekundete drei Wochen nach dem Einspielen des Solaris CPU OS Patchset vom Januar 2012 massive Performance-Probleme. Diese führten zu Verbindungsabbrüchen zum SWIFT-Netzwerk, dem Worst Case für das Banken-Business. SWIFT steht für „Society for Worldwide Interbank Financial Telecommunication" und ist eine Plattform, über die Banken untereinander Finanztransaktionen abwickeln.

Die einzige Änderung, die zuvor am System vorgenommen wurde, war das Solaris-Upgrade. Für das Business, den Software-Lieferanten und das Management war somit die Ursache des Problems gefunden. Nun wurde der Ball den Solaris-System-Administratoren zugespielt, die gewaltig unter Druck gerieten, das Performance-Problem zu lösen.

Die System-Architektur

Verteilt auf zwei Rechenzentren, ist die Plattform redundant aufgebaut. Es handelt sich um mehrere SPARC Enterprise M4000 Server, auf denen Solaris 10 installiert ist. Es wird die Solaris-Virtualisierungs-Lösung mit Solaris Zonen genutzt. Die verschiedenen Systeme der Applikationen laufen alle in Solaris Containern.

Die Daten der Banken-Applikationen befinden sich in einer Oracle-Datenbank 11g. Die gesamte Oracle-Datenbank-Installation ist in einem Zpool konzentriert. Dieser wird mit der Replikations-Lösung SNDR von Solaris synchron in das andere Rechenzentrum repliziert. Es ist bekannt, dass sich für solche Zwecke Oracle Data Guard besser eignen würde, die Lösung mit SNDR wurde jedoch vom Software-Hersteller empfohlen und vom Kunden gewünscht.

Die Daten liegen auf einem SAN. Es wird ausschließlich das Zetabyte File System (ZFS) verwendet. Die zu replizierende Datenmenge ist relativ klein (50 GB). Die Verteilung zwischen Schreib- und Lese-Operationen auf der Datenbank ist ausgeglichen. Die Antwortzeiten der Applikationen für die Benutzer verhalten sich in etwa linear zu den Antwortzeiten der Datenbank. Die Systeme sind in Bezug auf CPU nur wenig ausgelastet. Aufgrund der synchronen Replikation war früher die Netzwerk-Verbindung der Flaschenhals zwischen den Rechenzentren.

Eingrenzung des Problems

Es kristallisierte sich schnell heraus, dass der Engpass auf der I/O-Seite der Oracle-Datenbank lag. Die aufgezeichneten „System Activity Report"-Daten (5-Minuten-Mittelwerte) zeigten zum Teil 100 Prozent „busy"-Werte auf den Device-Files des Oracle Zpool und entsprechend auf dem Replikations-Device von SNDR. Obwohl der gesamte Daten-Durchsatz bescheiden war, zeigten die Oracle AWR-Reports sehr schlechte Antwortzeiten („AVERAGE WAIT") – zum Teil über 50 Millisekunden. Auch mit „iostat" (Mess-Intervall 1 Sekunde) waren viele hohe durchschnittliche Antwortzeiten zu beobachteten, also Werte von „asvc_t" („average service time of active transactions in milliseconds") von über 50. Weil für die Plattform eine dedizierte Netzwerk-Verbindung zwischen den Rechenzentren verfügbar war und die gemessenen Bandbreiten nur einen Teil der möglichen Kapazität ausmachten, war ein Problem mit der synchronen Replizierung unwahrscheinlich. Man konnte die weitere Analyse also auf Oracle und das ZFS fokussieren.

Übersicht ZFS

Auszug aus Wikipedia: „ZFS ist ein von Sun Microsystems entwickeltes transaktionales Dateisystem, welches zahlreiche Erweiterungen für die Verwendung im Server- und Rechenzentrums-Bereich enthält. Hierzu zählen die enorme maximale Dateisystemgröße, eine einfache Verwaltung selbst komplexer Konfigurationen, die integrierten RAID-Funktionalitäten, das Volume-Management sowie der prüfsummenbasierte Schutz vor Datenübertragungsfehlern."

Die Vorteile von ZFS sind unbestritten und der Autor möchte sie nicht

mehr missen. Insbesondere die einfache Administration, die optimale Integration mit Solaris und die eingebaute Funktionalität von Snapshots sind von großem Nutzen.

Man sollte sich der diversen ZFS-Filesystem-Chaches bewusst sein. Diese können durch geschickte Konfiguration und Nutzung von schnellen Devices wie „solide state disks“ die Performance erheblich verbessern. Sie können sich aber auch kontraproduktiv auswirken:

- Behinderung anderer Applikationen durch Speicher-Verbrauch des ARC-Cache
- Verdopplung der Schreib-Operationen für „synchrone writes“ im ZIL Log
- Abbremsen von Datenbanken durch „file level prefetching“- und „device level read ahead“-Mechanismen beim Lesen

Nachfolgend eine Liste von ZFS-Caches:

- Der „first level cache” (ARC cache) befindet sich im Memory. Es handelt sich um eine Variante des ARC-Algorithmus (Adaptive Replacement Cache)
- Optional können „second level disk caches” definiert werden:
  - Dafür eignen sich schnelle Disks wie SSD
  - Der „read cache“, als „L2ARC“ bezeichnet, wird über das Zpool-Property „cachefile“ aktiviert und über das ZFS-Property „secondarycache (all | none | metadata)“ beeinflusst
  - Der „write cache“ wird als „ZFS Intent Log“ (ZIL) bezeichnet und befriedigt POSIX-Requirements für synchrone Transaktionen. Ist kein separates ZIL-Device definiert, wird der ZIL ein Teil des Zpool. „zpool status” zeigt die „log devices” an.
  - Die „second level“-Chaches lassen sich während des Betriebs einfach hinzufügen, konfigurieren oder entfernen

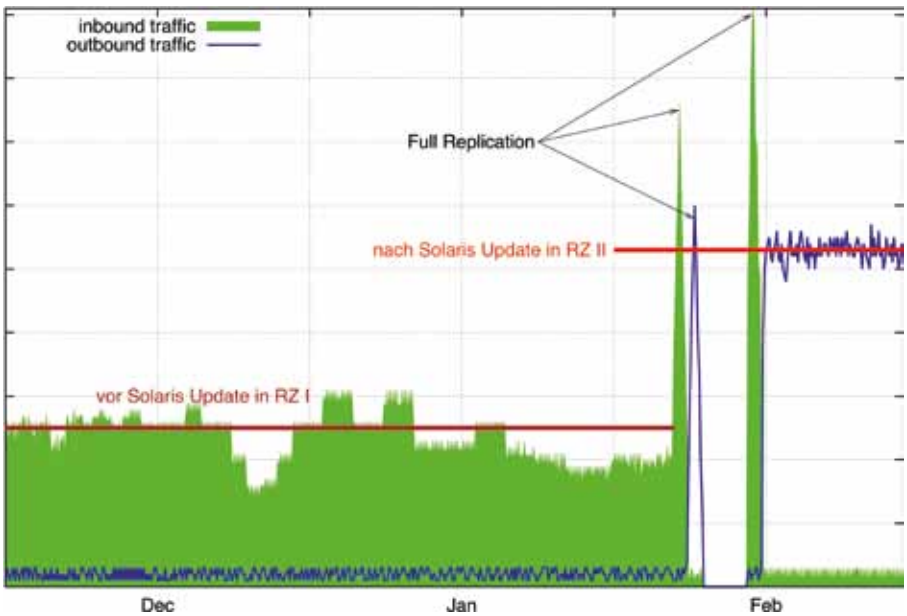

Abbildung 1: Replikations-Durchsatz auf der Leitung zwischen den Rechenzentren

ZFS und Oracle

In den Anfangszeiten von ZFS gab es Statements von Sun Microsystems, die darauf hinausliefen, ZFS nicht für Oracle-Datenbanken zu verwenden. Dies hat sich schon seit einiger Zeit geändert. Heute ist ZFS offiziell von Oracle für Datenbanken unterstützt und auch empfohlen. Wichtig ist jedoch, das ZFS nach „best practice“ aufzusetzen. Oracle hat diverse Whitepapers dazu verfasst – und diese sollten auch berücksichtigt werden. Die wichtigsten Punkte sind:

- ZFS „record size“ an „db block size“ anpassen
- Überwachen des „ARC cache“ im Memory und, wenn nötig, begrenzen
- ZFS intend log (ZIL) für „Oracle data files“ umgehen
- Den ZFS-Füllgrad (Usage) überwachen, immer mindestens unter 80 Prozent oder besser noch darunter bleiben
- Wenn möglich, bei Oracle-Datenbanken immer dedizierte Zpools für „redo logfiles“, „data files“ und „archivelog files“ mit dedizierten SAN LUNs verwenden
- Für Zpools nur ganze LUNs verwenden, keine Partitionen

Analyse von ZFS-Performance-Problemen

Wie bereits erwähnt, startet man hier am besten damit, die Oracle-„Best Practice“-Whitepaper zu studieren und zu untersuchen, ob die eigene Installation davon abweicht. Das beste Solaris-Bordmittel für die Analyse der I/O-Performance ist „iostat“. Man sollte ein Skript aufsetzen, das die „extended“-iostat-Werte im 1-Sekunden-Intervall rund um die Uhr aufzeichnet. Zudem sollte man sich protokollieren lassen, wie viel Memory sich der ZFS-ARC-Cache reserviert und ob das Memory auch schnell wieder freigegeben wird – sofern anderweitig gebraucht. Mit einer Beschränkung des ARC-Cache bewegt man sich auf der sicheren Seite. Der Befehl, um den ARC-Cache anzeigen zu lassen, lautet „root# echo „::memstat“ | mdb –k“.

Wichtige Informationen gewinnt man durch Analysieren. Das im Solaris-Betriebssystem eingebaute Dynamic Tracing (Dtrace) ist ein sehr mächtiges Tool. Es bietet die Möglichkeit, in laufenden Prozessen den Arbeitsspeicher, die Prozessorzeit, das Dateisystem und die Netzwerk-Ressourcen zu untersuchen. Im Zusammenhang mit der ZFS-Performance interessieren folgende Dtrace-Ergebnisse:

- Welche Programme verursachen “top writes“ und ”top reads“?
- Wie sieht die “byte size“-Verteilung aus?
- Sind „ganging“-Operationen zu beobachten? Dies wäre ein Zeichen von fragmentierten Zpools.

In der Oracle-Datenbank kommt man nicht darum herum, AWR Reports zu generieren, um aussagekräftige Bewertungen zur I/O-Performance machen zu können. Bei ZFS spricht man von „ganging“, wenn die Daten nicht mehr an einen zusammenhängenden Platz im ZFS geschrieben werden können und kleinere Gaps verwendet werden müssen. Die Anzahl der „ganging“-Operationen beim Schreiben ist linear zum Fragmentierungsgrad eines Zpool. Besteht der Verdacht auf eine Fragmentierung in einem Zpool, sollte man das entsprechende Tracing durchführen. Die Dtrace-Syntax “dtrace -qn ‚fbt::zio_gang_tree_issue:entry { @[pid]=count(); }‘ -c “sleep 300““ zeigt die „ganging“-Operationen in einem System an.

Problemlösung

Zurück zum Performance-Problem: Es wurde festgestellt, dass sich die über die Netzwerk-Verbindung zum anderen Rechenzentrum replizierte Datenmenge nach dem Solaris-Update sprunghaft fast um den Faktor zwei erhöht hatte. Abbildung 1 zeigt den Replikations-Durchsatz auf der Leitung zwischen den Rechenzentren. Grüne Kurve „inbound“, blaue Kurve „outbound“. Vor und nach dem Upgrade wurden Failover-Tests durchgeführt und jeweils eine volle Replikation (Spitzen) gefahren. Nach dem Upgrade liefen die Applikationen auf der anderen Seite also unter der blauen Kurve; die Richtung der Replikation hatte sich geändert.

Die Kernel Patches 147440-10 und 144500-19 des Januar-Patch-Bundle führten im ZFS neue Properties ein, unter anderem das Property „logbias“. Für Datenbank-Files sollte dieses auf „throughput“ gesetzt sein, es war jedoch nach der Patch-Installation auf dem Default-Wert „latency“. „throughput“ bedeutet: Der ZFS Intend Log wird für synchrones Schreiben nicht verwendet, womit sich die „writes“ quasi um die Hälfte reduzieren. Anmerkung: In Oracle Solaris10/08- bis 10/09-Installationen wurde ein ähnliches Verhalten durch das Setzen des Kernel-Parameters im „/etc/system“ durch „set zfs:zfs_immediate_write_sz=8000“ erreicht. Dieser Kernel-Parameter war hier jedoch nicht gesetzt. Nach der Änderung von „logbias“ auf „througput“ im ZFS mit den Oracle-Daten-Files hat sich tatsächlich die replizierte Bandbreite wieder auf den normalen Wert eingestellt. Leider war das Performance-Problem damit aber noch nicht gelöst.

Es wurde festgestellt, dass das ZFS mit den „redolog files“ fälschlicherweise auf eine „recordsize“ von 8 K anstatt 128 K eingestellt war. Zudem waren die Messungen der „bitesize“-Verteilung im Oracle-Zpool anders als erwartet. Die größte Verteilung sollte bei 8 K, entsprechend zur „recordsize“ des ZFS mit den Oracle „datebase files“ sein. Tabelle 1 zeigt jedoch ein ganz anderes Bild, wie mit dem Dtrace-Toolkit-Programm „bitesize.d“ gemessen wurde.

6. May		
371 zpool-Oracle_R\0		
value	**----- Distribution -----**	**count**
256		0
512	@@@	4498
1024	@@@@@	7236
2048	@@@@@@@@@@	14340
4096	@@@@@@	8388
8192	@@@@@	7461
16384	@@@@@	7710
32768	@@@@@	7068
65536		425
131072		40
262144		0

Tabelle 1

11. May		
371 zpool-Oracle_R\0		
value	**----- Distribution -----**	**count**
256		0
512	@	4498
1024	@	7236
2048	@@	14340
4096	@@@@@	8388
8192	@@@@@@@@@@@@@@@	7461
16384	@@@@@@	7710
32768	@@@@@@	7068
65536	@@	425
131072	@@	40
262144		0

Tabelle 2

Daraufhin wurden der „recordsize“-Wert im „redolog file“ ZFS auf 128 K geändert sowie anschließend in einem Servicefenster der ganzen Oracle-Zpool gesichert und wiederhergestellt, um diese Änderung wirksam zu machen. Danach kam das große Aufatmen. Die Performance bewegte sich wieder in einem Bereich wie vor dem Upgrade. Die WAR-Werte für „AVERAGE WAIT“ lagen bei 14 Millisekunden. Nun sahen auch die Resultate der Bitesize-Verteilung besser aus (siehe Tabelle 2).

Leider gab es bald eine bittere Enttäuschung – die Lösung war immer noch nicht gefunden, denn drei Wochen später war das Performance-Problem zurückgekehrt. Es blieb nur der bekannte Workaround: Servicefenster beantragen, Datensicherung des Oracle Zpool und Wiederherstellen der Daten. Das brachte wieder für drei Wochen Ruhe.

Abbildung 2 zeigt, wie sich die Performance-Werte innerhalb von zwei Wochen verschlechtert haben. Die Grafik zeigt die Werte für „AVERAGE WAIT (ms) for LOG FILE SYNC (time to wait before writing into RedoLog files)“.

Ein Oracle-Consultant erklärte vor Ort, dass das Performance-Problem durch die fortschreitende ZFS-Fragmentierung des Oracle-Zpool verursacht wird. Verstärkt wird das Problem, weil das Storage-Layout nicht den Best-Practice-Richtlinien folgt und keine dedizierten Zpools für eine Separierung von „redolog files“ und „database files“ verwendet werden. Bei dieser Umgebung wird wegen der SNDR-Replikation absichtlich nur ein Oracle-Zpool verwendet, da die Dauer einer vollständigen SNDR-Replikation abhängig von der Anzahl und der Größe der zu replizierenden Zpools ist.

ZFS überschreibt nie Daten, sondern folgt dem Copy on Write-Prinzip. Dies ist optimal für die Daten-Integrität und auch notwendig, damit Techniken wie „snapshots“, „cloning“, „shadow copy“ und „zfs send/receive“ überhaupt funktionieren. Leider handelt sich ZFS damit das Fragmentierungsproblem ein, sobald in einem Zpool regelmäßig geschrieben wird.

Die Empfehlung von Oracle war: „Storage-Layout ändern und Trennen beziehungsweise Verteilen von Oracle

„redolog files“ und „database files“ auf mehrere Zpools mit dedizierten LUNs.“ Diese Umorganisation wurde in die mittelfristige Planung aufgenommen.

Durch Benchmarks auf Test-Systemen und „ganging“-Messungen mit Dtrace wurde festgestellt, dass sich die Fragmentierungs-Problematik in der Umgebung durch die Erhöhung des ZFS-„freespaces“ entschärfen ließ. Die Oracle-Administratoren haben daraufhin zunächst die Daten im Oracle-Zpool so weit wie möglich reduziert. Dadurch verlängerte sich die Zeitspanne auf zwei Monate, bis eine neue Defragmentierungsaktion mit Service-Fenster notwendig wurde.

Mit einer Usage von 82 Prozent im Oracle-Zpool dauerte es drei Wochen, bis die Fragmentierung die Performance beeinträchtigte, mit 67 Prozent Usage acht Wochen. Abbildung 3 zeigt sehr schön die Auswirkung der schleichenden Fragmentierung. Erster Knick am 6. Juni durch Defragmentierung, zweiter Knick am 13. Juni durch Defragmentierung und mehr freien Platz im Zpool. Abbildung 4 zeigt die „ganging“-Operationen vor (rote Kurve) und nach einer Defragmentierung des Oracle-Zpool.

Nun fiel der Entschluss, den Oracle-Zpool in dem Maße zu vergrößern, dass die Zeit für eine volle Replikation gerade noch akzeptabel war, und den Freespace auf 50 Prozent heraufzusetzen. Mit dieser Aktion war das Fragmentierungsproblem gelöst. Einerseits sind keine „ganging“-Operationen mehr zu beobachten, andrerseits hat sich auch die Performance massiv verbessert. Die AWR-Werte „AVERAGE WAIT“ lagen bei 6 Millisekunden. Abbildung 5 zeigt die Entwicklung der „average wait times“. Links bis zum 13.6. mit einer ZFS-Usage von 80 Prozent, in der Mitte nach dem Löschen von Daten, „file relocation“ und ZFS-Usage von 67 Prozent und ganz rechts nach einer weiteren „file relocation“ und mit einer Usage von nur noch 50 Prozent.

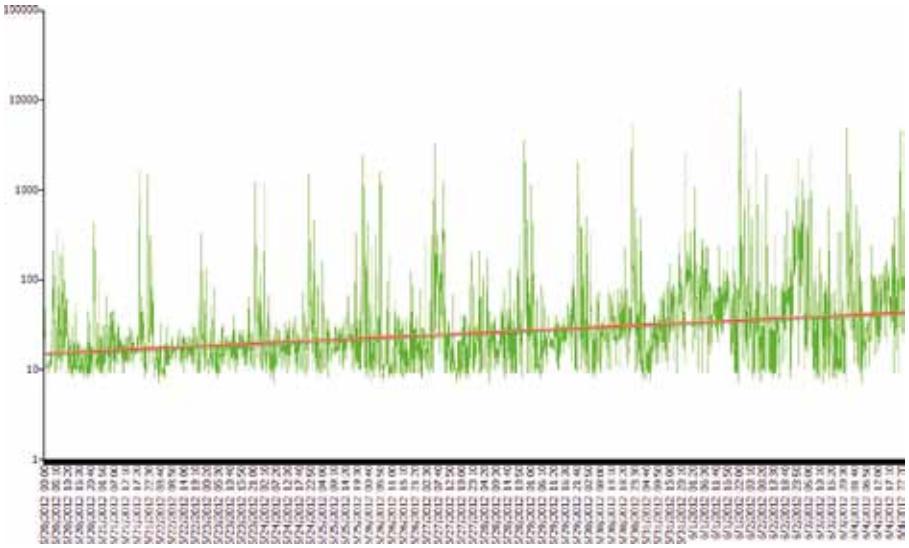

Abbildung 2: Verschlechterung der Performance-Werte innerhalb von zwei Wochen

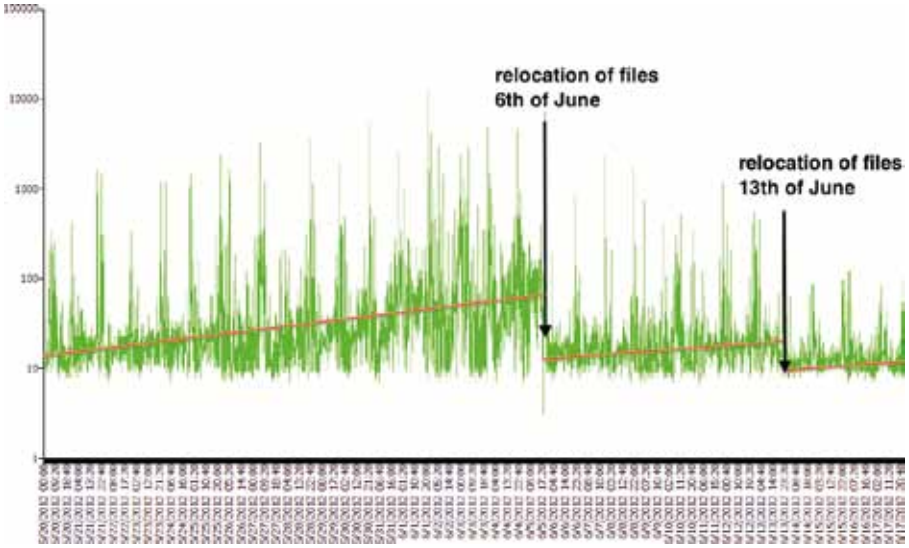

Abbildung 3: Die Auswirkung der schleichenden Fragmentierung

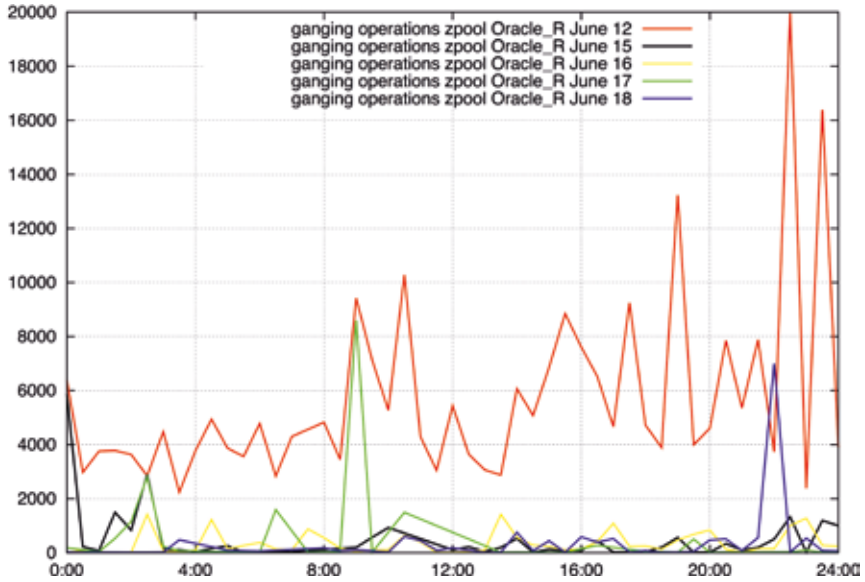

Abbildung 4: „ganging“-Operationen vor (rote Kurve) und nach einer Defragmentierung des Oracle-Zpool

Fazit

Die Gretchenfrage, die sich stellt: Standen die Performance-Probleme im Zusammenhang mit den applizierten Solaris-Patches oder entstanden sie durch eine langsame Reduktion des freien Plat-

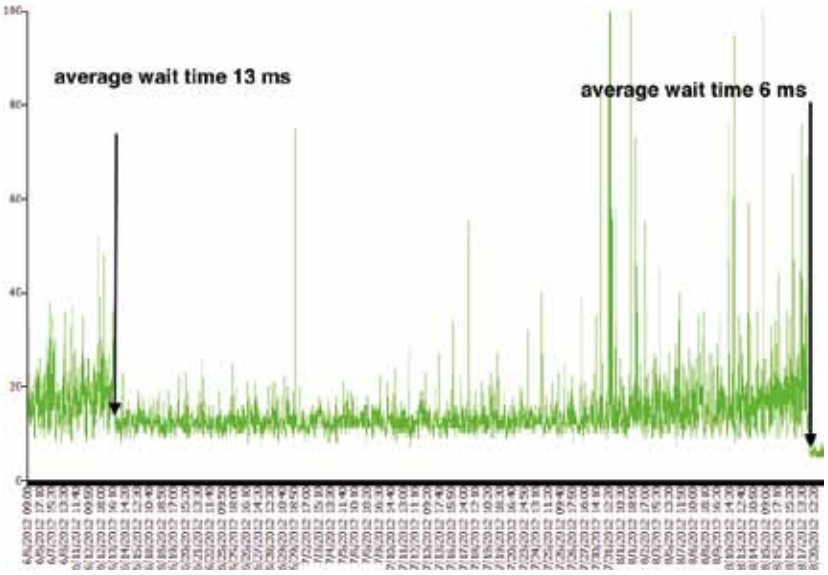

Abbildung 5: Die Entwicklung der „average wait times“

zes im Zpool? Gemäß den Aufzeichnungen hat sich der Füllgrad des Zpool nur minimal verändert. Die Vermutung liegt nahe, dass mit den Patches 147440-10 und 144500-19 Änderungen im ZFS vorgenommen wurden, die die Fragmentierungsproblematik in unserer spezifischen Umgebung akzentuiert haben.

Im Nachhinein betrachtet ist die Lösung des ZFS-Performance-Problems einfach: Reduktion des Füllgrades von ZFS von 80 auf unter 50 Prozent. Damit hat man die vorher immer wiederkehrende Fragmentierung des Oracle-Zpool, verbunden mit Performance-Problemen in der spezifischen Umgebung, gänzlich eliminiert und macht damit regelmäßige Defragmentierungs-Aktionen mit Service-Fenstern unnötig. Leider musste die Lösung selbst gefunden werden. Die Durchführung ist in dieser Umgebung mit relativ kleinen Datengrößen möglich, wäre hingegen bei größeren Datenmengen nicht praktikabel.

Es gab mehrere Service Requests beim Oracle Support – kein Wort über ZFS-Fragmentierung. Erst ein Oracle-Consultant vor Ort nahm das Wort in den Mund. Zudem muss man sehr lange suchen, um die Fragmentierungs-Problematik in der Oracle-Dokumentation zu finden – in den ZFS-Manuals herrscht hier großes Schweigen. Schön wäre es, wenn Oracle ein Tool bereitstellen könnte, das ähnlich wie ein ZFS-„Scrubbing“ im Hintergrund im Online-Betrieb laufen und den Zpool automatisch defragmentieren würde.

Literatur und Links

- Oracle ZFS Whitepaper: http://www.oracle.com/technetwork/server-storage/solaris/config-solaris-zfs-wp-167894.pdf
- Dtrace: http://www.brendangregg.com/dtrace.html
- ZFS Fragmentation: http://wildness.espix.org/index.php?post/2011/06/09/ZFS-Fragmentation-issue-examining-the-ZIL
- AWR: http://www.oracle-base.com/articles/10g/automatic-workload-repository-10g.php

Roman Gächter
Roman.Gaechter
@trivadis.com

Vertrauen in Performance, weniger in Oracle

Insbesondere in den letzten 12 Monaten hat das Interesse der IT-Anwender und Datacenter-Betreiber an sogenannten Appliances – integrierte Server/Storage/Network-Systeme – stark zugenommen. Oracle gehört zu den Trendsettern in diesem Bereich – deren Engineered Systems bauen auf hochintegrierten Oracle-Technologien auf, die deshalb auch gut optimiert sind. Darüber hinaus verfügen sie über einen One-Vendor-Support.

Um Interesse und Erfahrungen mit diesen Systemen zu eruieren, führte die DOAG gemeinsam mit der Experton Group eine Online-Befragung bei den DOAG-Mitgliedern durch. Insgesamt beteiligten sich über 500 Unternehmen an der Befragung, darunter 290 Anwender und 137 Partner, zusätzlich auch Consultants und Wettbewerber. Der Fragebogen umfasste insgesamt 15 Fragen, die nach den einzelnen Zielgruppen (Anwender, Anbieter, Partner, Berater) ausgewertet wurden. Ausgewählte Ergebnisse wurden auf dem DOAG 2013 IMC Summit am 6. Juni in Mainz von der Experton Group präsentiert und diskutiert.

Sehr wichtig ist den Anwendern insbesondere die Senkung der IT-Betriebskosten. Ungefähr die Hälfte der Unternehmen ist der Meinung, dass Appliances hierfür einen Beitrag liefern können. Als Vorteile sehen die Anwender insbesondere die einfache Implementierung und bessere Antwortzeiten der IT-Systeme – wogegen das Argument „Einstieg in die Private Cloud“ kaum Beachtung findet. Als Nachteil wird die Abhängigkeit vom Hersteller gesehen – sowohl Technologie wie Support betreffend, wogegen ein sehr hohes Vertrauen in die Leistungsfähigkeit/Performance der Systeme besteht.

Weitere Ergebnisse der Befragungs-Analyse sind unter http://engsys.doag.org zu finden.

Andreas Zilch
Experton Group AG
info@experton-group.com

Der sichere und effiziente Umgang mit großen und größten Datenmengen gehört heute sicherlich zu den herausragenden Aufgaben eines jeden IT-Infrastruktur-Dienstleisters – egal, ob betriebsintern oder für den freien Markt. Dabei stehen besonders die Datenintegrität, die technischen und organisatorischen Maßnahmen, die diese gewährleisten sollen, und natürlich der Sicherheitsaspekt beim Zugriff auf die gespeicherten Daten im Vordergrund.

ZFS-Verschlüsselung und andere Neuigkeiten in Solaris 11

Thomas Nau, Universität Ulm – kiz

ZFS hat mit seinem Erscheinen in Solaris 10 im Bereich der Datenintegrität nicht nur neue Maßstäbe gesetzt, sondern sich in den vergangenen Jahren zu einem Maßstab für Filesysteme schlechthin entwickelt. Alle anderen seither entstandenen und auch zukünftigen Entwicklungen müssen sich daran messen lassen. Die Dynamik der ZFS-Weiterentwicklung ist jedoch ungebrochen. Mit Solaris 11 stehen neben Performance-Verbesserungen auch neue Fähigkeiten zur Verfügung. So besteht jetzt die Möglichkeit, ZFS-Filesysteme und Volumes durch ZFS-bereitgestellte Block-Devices transparent zu verschlüsseln und diese Datenbereiche dann beispielsweise für virtualisierte Systeme in Zonen zu nutzen. Ebenfalls in Solaris 11 adressiert und vereinfacht wurde die Datenmigration von UFS hin zu ZFS.

Grundlegende ZFS-Design-Kriterien

Für den Betrieb einer universitären, zentralen Infrastruktur sind File- und Betriebssysteme notwendig, die weitreichende Vorkehrungen zum Schutz der Datenintegrität sowie der Datensicherheit bieten und damit vor Datenverlust schützen. Bei der Entwicklung von ZFS wurden von Beginn an insbesondere auch die designbedingten Schwachstellen herkömmlicher Filesysteme korrigiert:

- Gültige Daten werden niemals überschrieben (copy-on-write, COW).
- Starke Prüfsummen wie „fletcher4“ oder „sha256“ ermöglichen es, Fehler auf dem gesamten Datenpfad zu erkennen und diese, bei entsprechender redundanter Auslegung der Platten-Systeme, auch zu korrigieren. Hervorzuheben ist hierbei, dass die Prüfsummen getrennt von den eigentlichen Daten im sogenannten „Pointer-Bereich“ abgelegt sind. Aufgrund dieser Anordnung lassen sich die Auswirkungen durch den Ausfall einzelner Plattensektoren reduzieren.
- Die besonders wichtigen Metadaten des Systems, also diejenigen, die die Strukturen von Filesystemen und Pools beschreiben, werden mehrfach und von der übergeordneten Redundanz unabhängig in sogenannten „Ditto-Blocks“ gespeichert. Auf Wunsch lässt sich dieses Prinzip auch auf die Nutzerdaten erweitern, indem die entsprechenden Properties für das ZFS-Filesystem gesetzt werden.

ZFS bietet darüber hinaus eine Vielzahl betrieblicher Vorteile, etwa Snapshots und Slones. Diese kommen vermehrt als Ergänzung beziehungsweise als Ersatz üblicher Backup-Lösungen zum Einsatz. Nur so sind mit vergleichsweise geringem Aufwand Filesysteme mit mehreren 10 Millionen Dateien ein oder mehrmals pro Tag aus Sicht der Anwendung zeitlich konsistent zu sichern.

Regelmäßiges „scrubbing“, also das Überprüfen aller Prüfsummen, einschließlich gegebenenfalls notwendiger Korrekturen schützt vor unliebsamen Überraschungen, da Probleme bereits sehr frühzeitig erkannt werden können. Das nachfolgende Beispiel verdeutlicht die Erkennung von Fehlern (siehe Listing 1).

Verschlüsselung mit ZFS

Mit der Freigabe von Solaris 11 haben Anwender nun zusätzlich die Möglichkeit, ZFS-Filesysteme und -Volumes mit sicheren Algorithmen wie AES-256 zu verschlüsseln. Voreingestellt ist „AES-128-CCM“. Sofern die eingesetzte Hardware Beschleuniger für kryptographische Operationen bietet, etwa Oracle T4- und T5-Chips oder auch die Intel AES-NI-Erweiterung, werden diese automatisch vom Solaris-Crypto-Framework erkannt und genutzt. Eine Verschlüsselung des Root-Filesystems ist derzeit ausschließlich für nicht globale Zonen möglich. Hierzu später mehr.

Zwei Punkte sind vor dem Einsatz der Verschlüsselung zu beachten. Zum einen kann sie nur für neu anzulegende Filesysteme und Volumes aktiviert werden und zum anderen ist auch eine spätere Deaktivierung nicht möglich. Man bindet sich also für die Lebensdauer der Filesysteme. Alle notwendigen Befehle sind in das „zfs“-Kommandozeilen-Tool integriert. Der Einsatz von Verschlüsselung ist auch bei bereits existierenden Pools möglich, solange diese mindestens die Versionsnummer 30 tragen oder ein dahingehender Upgrade möglich ist (siehe Listing 2). Die aktuellen, auf den ehemaligen OpenSolaris-Quellen basierenden ZFS-Versionen in Illumos und FreeBSD bieten die Möglichkeit zur Verschlüsselung nicht. Die zugehörigen ZFS-Properties lassen sich wie gewohnt auslesen (siehe Listing 3).

Der einmal gewählte Verschlüsselungsalgorithmus ist nicht änderbar. Alle zugehörigen ZFS-Properties wer-

```
# zpool scrub testpool
# sleep 15 ; zpool status -v testpool

  pool: testpool
 state: ONLINE
status: One or more devices has experienced an
        unrecoverable error. An attempt was made to correct
        the error.  Applications are unaffected.
action: Determine if the device needs to be replaced, and
        clear the errors using 'zpool online' or replace the
        device with 'zpool replace'.
        see: http://www.sun.com/msg/ZFS-8000-9P
 scrub: scrub in progress, 6.21% done, 0h4m to go
config:
        NAME          STATE   READ WRITE CKSUM
        testpool      ONLINE     0     0     0
          mirror-0    ONLINE     0     0     0
            c4t0d0s0  ONLINE     0     0     0
            c4t1d0s0  ONLINE     0     0    58  228.5 repaired
```

Listing 1

```
# zfs create -o encryption=aes-256-ccm pool/thomas
Enter passphrase for 'pool/thomas':
Must be at least 8 characters.
Enter passphrase for 'pool/thomas':
Enter again:
```

Listing 2

den vererbt, also alle nachgeordneten Filesysteme, insbesondere auch Snapshots und Clones, sind ebenfalls zwingend verschlüsselt.

Die bei ZFS zum Einsatz kommende Schlüsselverwaltung ist zweistufig. Nach außen sichtbar ist der sogenannte „Wrapping Key". Er wird dem System beispielsweise aus einer „passphrase" generiert. Alternativ kann der Schlüssel auch als Byte-Folge oder Hexadezimal-String bereitgestellt werden. Der Wrapping Key dient ausschließlich dazu, den eigentlichen, vom Kernel verwendeten Schlüssel, den „Encryption Key", zu schützen. Beim Setzen einer neuen „passphrase" verschlüsselt das System nur den vorhandenen Encryption Key neu. Der Datenbestand bleibt davon unberührt, die Daten werden also nicht komplett neu kodiert, sondern bleiben mit den jeweiligen alten Encryption Keys verschlüsselt. Dies gilt übrigens auch für die Änderung des Encryption Key durch den Solaris-Kern. Hier wird lediglich eine ausreichend lange Byte-Folge vom System zufällig ausgewählt. Hier nutzt das System mittels Solaris Crypto Framework gegebenenfalls vorhandene Zufallszahlen-Generatoren. Das Kommando „# zfs key -K pool/thomas" übernimmt diese Aufgabe. Die Änderung spiegelt sich direkt in den Properties wieder (siehe Listing 4).

Der neue Schlüssel kommt dann für alle ab dem Änderungszeitpunkt geschriebenen neuen Daten zum Einsatz. Änderungen dieses internen Schlüssels sollten eher selten und im Einklang mit den lokal vorgegebenen Sicherheitsempfehlungen vorgenommen werden. Das National Institute of Standards and Technology (NIST) empfiehlt, derartige Schlüssel alle zwei Jahre zu wechseln.

Gänzlich anders verhält es sich mit den zu jedem verschlüsselten ZFS-Volume beziehungsweise -Filesystem gehörenden individuellen Schlüsseln und „passphrases". Diese lassen sich nach Belieben ändern. Um sie vom Benutzer abzufragen oder auch aus einem Gerät auszulesen, stehen mehrere Mechanismen und Formate zur Verfügung. Dies sind derzeit:

- *Prompt*
  Interaktive Abfrage der „passphrase", wenn das Filesystem erzeugt oder „gemountet" wird
- *file://filename*
  Der Schlüssel wird aus einer Datei, etwa aus einem USB-Stick, gelesen
- *pkcs11*
  Der Schlüssel wird aus einem PKCS#11-Token ausgelesen
- *https://location*
  Der Schlüssel wird von einem Web-Server über eine HTTPS-Verbindung bereitgestellt

Dabei stehen folgende Formate zur Verfügung:

- *Raw*
  Bytefolge
- *Hex*
  Hexadezimal-String
- *Passphrase*
  Passwort, aus dem der Schlüssel generiert wird

Achtung: Beim Aushängen („umount") eines verschlüsselten Filesystems werden die Schlüssel nicht aus dem Kernel entfernt, daher kann ein Administrator das Filesystem bis zu einem „reboot" wieder „mounten" – und zwar ohne dass der Schlüssel benötigt wird. Um dies zu verhindern, muss der Encryption Key des infrage kommenden Filesystems oder Volumes mit „# zfs key -u pool/thomas" explizit aus dem Kernel entfernt werden. Die folgenden Beispiele verdeutlichen den einfachen Umgang mit den Verschlüsselungsfähigkeiten von ZFS. Wechsel des Schlüssels (siehe Listing 5), Laden des Schlüssels in den Kernel (siehe Listing 6) und Wechsel des Bereitstellungsmechanismus, etwa aus einer Datei (siehe Listing 7). Diese ist im Idealfall auf einem leicht entfernbaren

```
# zfs get \
      encryption,keychangedate,keysource,keystatus,rekeydate \
      pool/thomas
NAME         PROPERTY       VALUE                  SOURCE
pool/thomas  encryption     aes-256-ccm            local
pool/thomas  keychangedate  Sat Sep 15 18:03 2012  local
pool/thomas  keysource      passphrase,prompt      local
pool/thomas  keystatus      available              -
pool/thomas  rekeydate      Sat Sep 15 18:03 2012  local
```

Listing 3

```
# zfs get keychangedate,rekeydate pool/thomas
NAME          PROPERTY       VALUE                  SOURCE
pool/thomas   keychangedate  Sat Sep 15 18:03 2012  local
pool/thomas   rekeydate      Sat Sep 15 18:06 2012  local
```

Listing 4

```
# zfs key -c pool/thomas
Enter new passphrase for 'pool/thomas':
Enter again:
```

Listing 5

```
# zfs key -l pool/thomas
Enter passphrase for 'pool/thomas':
```

Listing 6

```
# echo "My Big Secret" > /root/KEY
# chmod 400 /root/KEY
# zfs key -u pool/thomas
# zfs set keysource=passphrase,file:///root/KEY pool/thomas
# zfs key -l pool/thomas
```

Listing 7

```
# echo "My_Passphrase" > /root/KEY_TN
# zfs create -o encryption=aes-256-ccm \
             -o keysource=passphrase,file:///root/KEY_TN \
             rpool/home/tn_enc
# ls -l /home/tn_encrypt
total 0

obi-wan# zfs set shadow=file:///home/tn rpool/home/tn_encrypt
obi-wan# ls -l /home/tn_enc
total 4
drwx------   2 nau      kizinfra       2 Sep 10 15:08 bin
drwx------   2 nau      kizinfra       2 Jul 10 08:04 doc
drwx------   2 nau      kizinfra       2 Oct 27  2002 src
```

Listing 8

Datenträger wie einem USB-Memory-Stick gespeichert.

Die Verschlüsselung von Home-Directories wird in Solaris 11 durch Pluggable Authentication Modules (PAM) unterstützt, sofern diese in jeweils eigenen ZFS-Filesystemen untergebracht sind. In diesem Fall leitet sich der Wrapping Key aus dem Passwort des Nutzers ab. Weiterführende Informationen hierzu stehen unter „https://blogs.oracle.com/darren/entry/user_user_home_directory_encryption" in Darren Moffats Blog.

Verschlüsselung versus Deduplizierung

Das Zusammenspiel zwischen ZFS-eigener Komprimierung, Verschlüsselung und Deduplizierung stellt sich beim Schreiben von Daten folgendermaßen dar:

1. Komprimierung der Daten, sofern aktiviert
2. Verschlüsselung der komprimierten Daten, sofern aktiviert
3. Bildung der Prüfsumme
4. Deduplizierung, sofern aktiviert

Da die zur Verschlüsselung der Datenblöcke eingesetzten Encryption Keys zwischen unterschiedlichen Filesystemen im Allgemeinen nicht gleich sind, unterscheiden sich auch identische Eingangsdaten nach dem zweiten Schritt der Prozesskette. Dies kann erheblichen Einfluss auf die zu erwartende Deduplizierungsrate des gesamten Pools haben.

Zonen

Ebenfalls neu in Solaris 11 sind die sogenannten „Immutable Zones". Sie bieten im Vergleich zu herkömmlichen Zonen eine noch weiter reichende Sicherheit, da das zugehörige Root-Filesystem vor Manipulationen innerhalb der Zone geschützt wird. Dieser Schutz kann nach Wahl sehr strikt ausfallen, also keinerlei Ausnahmen zulassen, oder auch flexibel gestaltet sein. Im letzteren Fall können beispielsweise Dateien im Home-Directory von „root" oder in „/etc" und „/var" verändert werden. Auch eine Kombination mit verschlüsselten Filesystemen ist leicht zu bewerkstelligen, wie Darren Moffat in einem seiner Blog-Beiträge unter https://blogs.oracle.com/darren/entry/immutable_zones_on_encrypted_zfs zusammengefasst hat.

Die Solaris 11 „Shadow Migration" gibt Administratoren ein Mittel an die Hand, um lokale, aber auch NFS-gemountete UFS- und ZFS-Filesysteme in ein neues ZFS-Filesystem zu migrieren. Dies geschieht weitestgehend transparent für die Nutzer des Systems. Eine Ausnahme bilden die gelegentlich auftretenden Verzögerungen beim Datenzugriff, sofern eine Datei noch nicht migriert wurde. Die zu erfüllenden Voraussetzungen sind einfach:

- Das Quell-Filesystem muss „read-only" gemountet sein und darf im Falle von NFS auch serverseitig nicht verändert werden
- Das Ziel-Filesystem muss leer sein

Mittels „Shadow Migration" lassen sich zum Beispiel bestehende Home-Verzeichnisse transparent und quasi im laufenden Betrieb in verschlüsselte ZFS Filesysteme migrieren, da ja eine nachträgliche Aktivierung der Verschlüsselung für bereits benutzte Filesysteme nicht möglich ist. Lediglich die notwendige Anpassung des Pfades für die Verzeichnisse, etwa in „/etc/passwd", hat gegebenenfalls eine Unterbrechung für die jeweiligen Nutzer zur Folge. Das nachfolgende Beispiel verdeutlicht die notwendigen Schritte (siehe Listing 8).

Thomas Nau
thomas.nau
@uni-ulm.de

Dieser Artikel richtet sich an Datacenter-Architekten und Solaris-Administratoren. Er befasst sich mit dem Oracle-Produkt „Oracle VM for SPARC", auch bekannt als Logical Domains (LDoms). Im ersten Teil wird die Technologie erklärt und im zweiten Teil von praktischen Erfahrungen berichtet.

Was sind Logical Domains (LDoms) und worin liegt ihr Nutzen?

Marcel Hofstetter, JomaSoft GmbH

Voraussetzung für diese Technologie ist ein Oracle-SPARC-Server der T-Serie (CMT System), denn der notwendige Hypervisor ist nur in diese Server-Hardware integriert. Da der Hypervisor in der Hardware/Firmware enthalten ist, wird der Virtualisierungs-Overhead auf ein Minimum reduziert. Die LDom-Manager-Software ist Bestandteil von Solaris 11 und kann für Solaris 10 kostenlos von Oracle bezogen werden.

In jede logische Domäne (LDom) lässt sich eine unabhängige Solaris-Betriebssystem-Instanz installieren. Somit können verschiedene Solaris-Releases gleichzeitig auf derselben Hardware betrieben werden. Dies ist eine ideale Möglichkeit, parallel zu Solaris 10 neue Solaris-11-Umgebungen aufzubauen.

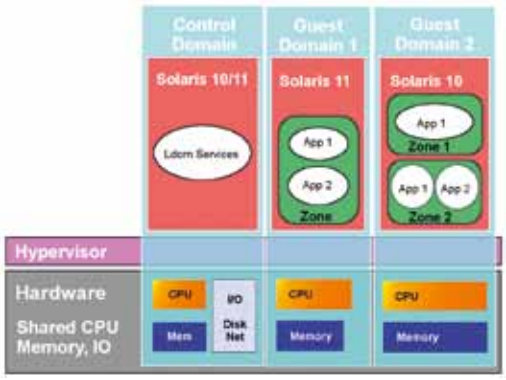

Abbildung 1: LDoms auf einen Blick

Die LDoms (oder Guest Domains) werden via „Control Domain" verwaltet (siehe Abbildung 1). Diese stellt virtuelle Devices und Services bereit, die den LDoms zugeteilt werden können und somit den Zugriff auf Disks und Netzwerk ermöglichen. Ressourcen wie CPU und Memory werden den LDoms fix zugewiesen, können aber später auch zur Laufzeit verändert werden. Eine LDom kann ohne Unterbrechung von einem Server auf einen anderen migriert werden, wenn die Daten auf einem zentralen Storage abgelegt sind (Live Migration).

Vorteile

LDom ist eine kostenlose Technologie, die die Virtualisierung und Konsolidierung im Solaris-Rechenzentrum unterstützt. Mit den von Oracle angebotenen „physical-to-virtual"-Tools (P2V) lassen sich alte, nicht mehr unterstützte Systeme einfach auf neue Hardware migrieren, ohne an der eigentlichen Server-Installation etwas verändern zu müssen.

Dank der Migrations-Funktionen können die LDoms bei Bedarf zwischen Systemen verschoben werden. Es lassen sich Kosteneinsparungen erzielen, da die bestehende Hardware besser ausgelastet ist. Die reduzierte Anzahl physischer Server führt zu weniger Bedarf an Platz, Strom und Kühlung.

Mit LDoms lassen sich neue Applikations-Umgebungen in wenigen Minuten bereitstellen. Aus organisatorischen Gründen empfiehlt es sich, pro Kunde/Mandant mindestens eine LDom zu erstellen und darin mehrere Solaris-Zonen für die einzelnen Applikationen/Umgebungen. Für Oracle-Software sind LDoms als „Partitionen" akzeptiert, wodurch sich Optimierungen beziehungsweise Einsparungen bei den Software-Lizenzen erzielen lassen.

```
-bash-4.1$ gdom -c show cdom=s0024
      Name  cState      rState               CDom        ActiveBuild        CPU    RAM     Comment
     g0050  ACTIVE      ACTIVE (RUNNING)     s0024       5.10sv_u8_req      2      2048    MNGT and PUBL
     g0051  ACTIVE      ACTIVE (RUNNING)     s0024       5.10svz_u10_req    2      1024    brands
     g0054  DETACHED    -                    (s0024)     5.10sv_u8_req      4      1024    Patch Testing
     g0055  DETACHED    -                    (s0024)     5.10sv_u9_user     4      2048    MQ V6
     g0056  DETACHED    -                    (s0024)     5.10sv_u9_all      8      2048    s10 mit iscsi
     g0058  ACTIVE      ACTIVE (RUNNING)     s0024       5.10sv_u9_all      4      1024    T4-2 Level
     g0060  ACTIVE      ACTIVE (RUNNING)     s0024       s11.1-sru3-s       4      2048    Produktiv
     g0064  DETACHED    -                    (s0024)     s11.0-sru8-s       4      1024    upgrade test
     g0065  DETACHED    -                    (s0024)     5.10sv_u9_user     4      1024    MQ V7
     g0066  ACTIVE      ACTIVE (RUNNING)     s0024       s11.0-sru8-s       8      1024    Demo Gdom
     g0070  DETACHED    -                    (s0024)     s11.0-s            4      1536    s11 cluster
     g0072  ACTIVE      ACTIVE (RUNNING)     s0024       s11.1-s            2      2048    U1 Testing
     g0073  ACTIVE      ACTIVE (RUNNING)     s0024       5.10sv_u11_user    4      1024    Veritas 6.0.1
```

Listing 1

```
-bash-4.1$ cdom -c show name=s0024
          Server  Information
             ORCL,SPARC-T4-1  CPU: 1  Threads: 64 x SPARC-T4 2848MHz
          Domain  Information        Ldom Version:  3.0
            Type  State              CPUs        RAM/MB          MAUs        CPU%        RAM%
            NODE  ACTIVE             64          32256           -           100         100
            CDOM  ACTIVE             16          4096            0           25          12
            GDOM  -                  26          10240           0           41          32
            LEFT  -                  22          17920           -           34          56
```

Listing 2

Nachteile

Beim Ausfall eines physischen Servers sind zahlreiche Solaris-Instanzen und -Applikationen betroffen. Mit der Zunahme von Technologien, Komplexität und Flexibilität im Rechenzentrum steigt die Anforderung an die System-Administratoren. Diese Problemfelder können mit einem geeigneten Management-Werkzeug adressiert werden.

Praktische Erfahrungen

Das Unternehmen des Autors betreibt einen Oracle-T4-1-Server, auf dem zahlreiche LDoms mit vielen unterschiedlichen Solaris-Releases laufen. Die LDoms oder Guest Domains (gdoms) werden mit dem eigenen Management-Werkzeug Virtual Datacenter Control Framework (VDCF) verwaltet.

Die LDom-Daten sind auf dem SAN-Storage abgelegt. Somit kann eine LDom vom laufenden System entfernt („detach“) und später bei Bedarf wieder in Betrieb genommen oder auf ein anderes T-System migriert werden. Dies ist ideal, um viele Test-Umgebungen zur Verfügung zu haben, wobei nicht alle gleichzeitig aktiv sein müssen. Innerhalb von Minuten können zudem neue Test-Umgebungen erzeugt und installiert werden (siehe Listing 1).

In der VDCF-Datenbank ist jederzeit ersichtlich, wie viele Ressourcen noch für zusätzliche LDoms zur Verfügung stehen (siehe Listing 2).

Kundenprojekt mit LDoms bei der Notenstein Privatbank

Für Performance-Vergleiche zwischen einem M5000- und einem T4-2-System wurde dieselbe Solaris-Version der M5000 in eine LDom auf dem T4-2 installiert (siehe Abbildung 2). So konnte man die Bank-Applikation mit ZFS in die T4-2-LDom kopieren und einen

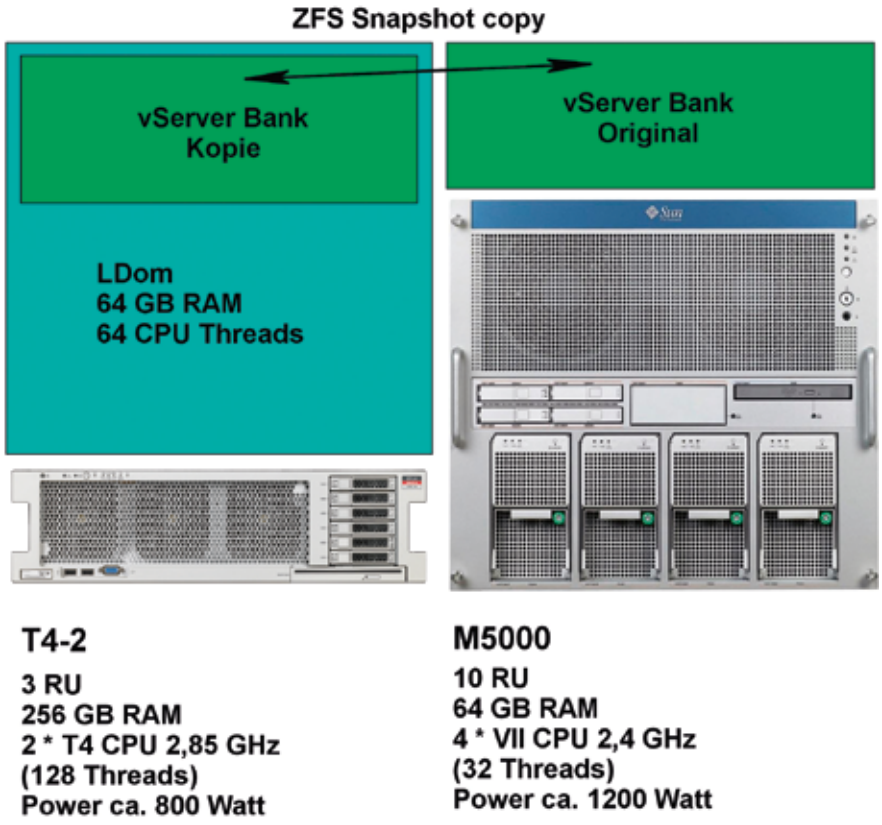

Abbildung 2: T4 und M5000 im Vergleich

1:1-Performance-Vergleich durchführen.

Der Performance-Vergleich ergab, dass sequenzielle Verarbeitungen auf beiden Systemen die gleiche Zeit in Anspruch nahmen. Bei parallelen Verarbeitungen, wie bei der Depot-Bewertung, konnte die Durchlaufzeit auf dem T4-2-System im Vergleich zur M5000 halbiert werden.

Dieser Test zeigt auf, wie einfach ein System mittels Oracle-Technologien von einem Server auf einen anderen übertragen werden kann – einer der großen Vorteile der Virtualisierungs-Technologien. „Das VDCF Management Framework von JomaSoft gibt uns im täglichen Betrieb viel Flexibilität. Es versetzt uns in die Lage, Zonen von einer M5000 in eine LDom auf einer T4-2 zu migrieren. Solche Migrationen dauern nur wenige Minuten und können von uns selbstständig durchgeführt werden. Dank VDCF ist es nicht notwendig, alle Details der Solaris-Virtualisierungs-Technologien zu kennen“, so Michael Bütler, Leiter IT Betrieb, Notenstein Privatbank AG.

Marcel Hofstetter
hofstetter@jomasoft.ch

Der Artikel zeigt anhand eines Beispiels aus der Praxis, wie das Netzwerk in einer virtualisierten Solaris-Umgebung aus Oracle VM Server SPARC in Kombination mit Solaris-Zonen konfiguriert wird. Dabei kommen die Technologien Solaris, LDom, VLAN, Link Aggregation 802.3ad / Trunking, Cisco Trunking ISL/802.1Q, Solaris IPMP und exclusive IP von Solaris-Zonen zum Einsatz.

Eine schwere Netzwerk-Aufgabe mit der Solaris-Virtualisierungslösung Oracle VM Server SPARC

Roman Gächter, Trivadis AG

Vom Namen her kann man die Virtualisierungs-Lösung „Oracle VM Server SPARC“ leicht mit dem Produkt „Oracle VM Server X86“ verwechseln. Es handelt sich dabei um zwei verschiedene Technologien, das letzte Wort macht den Unterschied aus. „Oracle VM Server X86“ ist das Produkt für die Intel-X86-Hardware und baut auf der XEN-Technologie auf. In diesem Artikel wird VM Server SPARC thematisiert, das auch unter dem Namen „Logical Domains“ (LDoms) bekannt und nur auf SPARC-Hardware verfügbar ist.

Ausgangslage

Oft ist man vor vollendete Tatsachen gestellt und muss aus den bestehenden Möglichkeiten das Beste herausholen. Im vorliegenden Fall war die Hardware bereits gekauft und konfiguriert. Im Rahmen eines In-Sourcing-Projekts wurde eine Banken-Applikation auf neuer SPARC-Hardware in den firmeneigenen Rechenzentren aufgesetzt. Um die Hardware-Kosten niedrig zu halten, entschied man sich für Oracle VM Server SPARC. Es war notwendig, mehrere Umgebungen (Produktion, Entwicklung und Abnahme) aufzubauen. Zudem musste die Hardware redundant über zwei Rechenzentren verteilt bereitgestellt werden. So wurden zwei SPARC-T4-2-Boxen gekauft und verteilt auf zwei autonome Rechenzentren installiert.

Pro Umgebung wurde jeweils eine „Guest LDom“ aufgesetzt. Die einzelnen Systeme der Applikation wurden als Solaris-Zonen in der „Guest LDom“ konfiguriert. Die gesamte Installation der Zonen liegt auf dem SAN. Im Desaster-Fall können die Zonen in das andere Rechenzentrum verschoben werden (siehe Abbildung 1).

Knacknuss Netzwerk

Es standen insgesamt acht 1-GB-Ethernet-Ports zur Verfügung. Die Frage war nun: „Wie lege ich das Netzwerk aus, damit eine Primary Domain, drei Guest Domains und fünfzehn Zonen verteilt auf drei Umgebungen redundant und mit optimalem Durchsatz angeschlossen werden können? Es musste ein Netzwerkkonzept ausgearbeitet werden, das die folgenden Vorgaben erfüllt:

- Verwendung der bestehenden T4-2-Hardware, die mit zwei „Quad

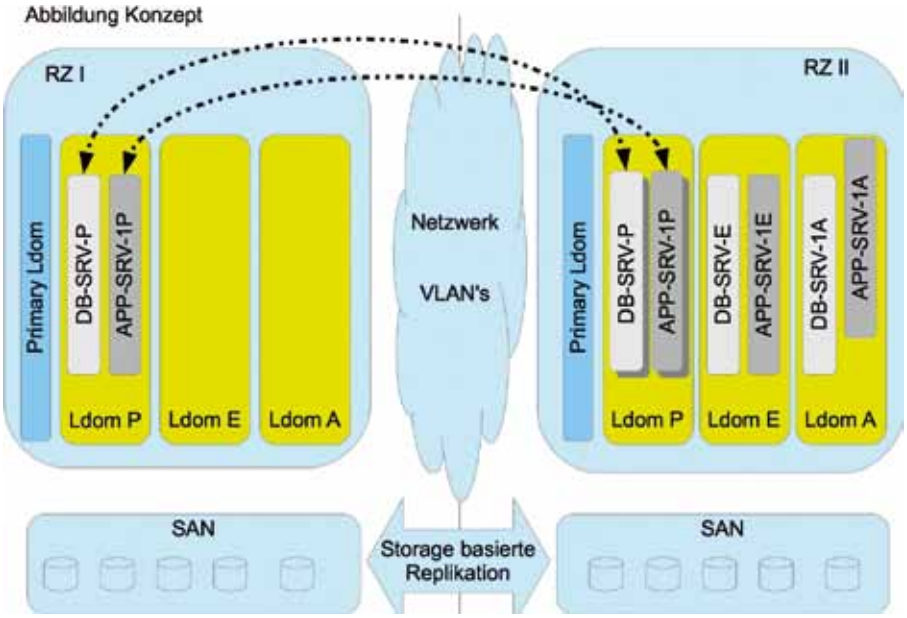

Abbildung 1: Das Konzept

Port“ Network Interface Controllern (NICs), also acht 1-GBPorts, bestückt ist.
- Neben der Primary Domain müssen noch drei weitere Domains betrieben werden.
- Pro Umgebung sind fünf Solaris-Zonen installiert. Die einzelnen Zonen innerhalb der Umgebungen sind durch Firewalls zu separieren, da für mehrere Mandanten ausgelegt
- Die SPARC-Hardware ist direkt an Cisco Core Switches „Trunking Ports“ angeschlossen. Die virtuellen Switches und Vnets der LDoms müssen mit VLANs konfiguriert werden.
- Die Netzwerk-Performance ist für die produktive Umgebung kritisch.
- Das Netzwerk-Konzept muss auf Ausfallsicherheit ausgelegt sein.

Die beiden von Solaris unterstützten Technologien, um Netzwerk-Interfaces zu bündeln, sind „IP Multipath“ (IPMP) und „Link Aggregation“. Beide Technologien bieten Features an, die sich zum Teil überschneiden, jedoch auf unterschiedlichen Netzwerk-Schichten des OSI-Modells implementiert sind: „Link Aggregation“ in der MAC-Schicht, IPMP in der IP-Schicht. Es war schnell klar, dass in dieser virtualisierten Umgebung eine Gruppierung von NICs notwendig war. „Link Aggregation 802.3ad“ bietet folgende Vorteile:

- Implementiert auf dem MAC-Layer
- Erhöhte Bandbreite durch Bündelung mehrere NICs
- Automatisches „Failover/Failback“ von Links des Aggregats
- Load Balancing, Verteilung des „Inbound und Outbound Traffic“ gemäß der gewählten Policy
- Redundanz ist möglich

Das Zusammenspiel dieser Technologie mit Cisco Trunking wurde anhand eines Proof of Concept (POC) überprüft. Bei der Variante 1 (siehe Abbildung 2) hat man ein Aggregat über je einen Port des e1000g- und igb-NIC gebildet und an zwei Switches angeschlossen. In LDom wurden ein virtueller Switch und zwei virtuelle Netzwerke konfiguriert und diese den Solaris-Zonen exclusiv zur Verfügung gestellt. Die Zonen mussten zwingend in unterschiedlichen VLANs betrieben werden (Trennung der Systeme beziehungsweise der logischen Netzwerke).

Die Idee dieser Konfiguration war: Es gibt eine redundante Netzwerk-Infrastruktur. Ob nun ein Switch, ein NIC oder ein einzelner NIC Port ausfällt – das Netz bleibt mit reduzierter Bandbreite verfügbar.

Leider hat diese Konfiguration in einem Fehlerfall (Ausfall eines Core Switch) nicht funktioniert. Durch den simulierten Ausfall eines der Switches hat jeweils eine der Zonen die Netzverbindung verloren. Erst nach dem Gratuitous-ARP-Paket der Zone, das bei Solaris in einem Intervall von fünf Minuten gesendet wird, wurde der Link-Status „down des Vnets“ erkannt.

Das Gratuitous-ARP-Package hat unter anderem den Effekt, ARP Caches im Netzwerk zu aktualisieren. Die Protokolle „Link Aggregation 802.3ad“ und Cisco Trunking sind in dieser Konfiguration nicht kompatibel. Die notwendigen VLAN- und Link-Informationen wurden zwischen den verschiedenen Protokollen nicht korrekt ausgetauscht. Da die notwendigen Konfigurationsänderungen im Netzwerk-Bereich nicht vorgenommen werden konnten, musste man eine andere Lösung suchen.

Bei der Variante 2 (siehe Abbildung 3) kamen zwei Aggregate zum Einsatz. Diese sind nicht mehr Switch-übergreifend. In LDom wurden nun zwei virtuelle Switches und vier virtuelle Netzwerke konfiguriert. Das neue Element ist hier IP Multipath (IPMP) von Solaris. Die Vnets der Zonen sind auf zwei Aggregate beziehungsweise zwei Cisco-Switches verteilt. Die Idee dieser Konfiguration auch hier ist: Wir haben eine redundante Netzwerk-Infrastruktur. Ob nun ein Switch, ein NIC oder ein einzelner NIC Port ausfällt – das Netz bleibt mit reduzierter Bandbreite verfügbar.

Mit dieser Konfiguration wurde dank IPMP eine funktionierende Redundanz erreicht. Auch der Ausfall eines Switch wird korrekt erkannt und IPMP verwendet nur noch das Vnet mit Link-Status „up“. Wichtig ist hier, dass in der LDom-Konfiguration für die Vnets das Property „link_state“ auf „physical“ gesetzt ist, sonst klappt es nicht. Abbildung 4 zeigt schematisch die Übersicht des Netzwerks mit allen Aggregaten und LDom sowie symbolisch jeweils zwei Zonen pro LDom.

Die Aggregate „aggr 1“ und „aggr 3“ sind am ersten Core-Switch, „aggr 2“ und „aggr 4“ am zweiten angeschlossen. Die Vnets sind über IPMP immer so aufgesetzt, dass sie sich über zwei verschiedene Aggregate und Core-Switches erstrecken. Die produktive Umgebung hat die Bandbreite von 4 x 1 Gbps zur Verfügung. Die Umgebungen „Entwicklung“ und „Abnahme“ teilen

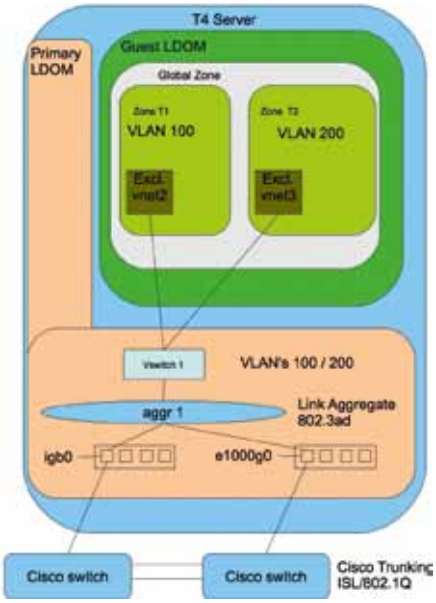

Abbildung 2: POC Variante 1

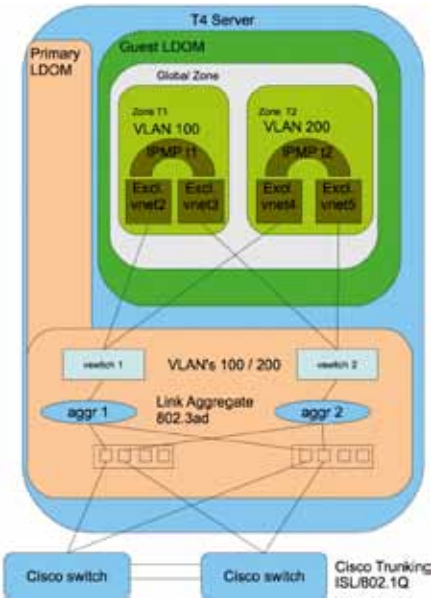

Abbildung 3: POC Variante 2

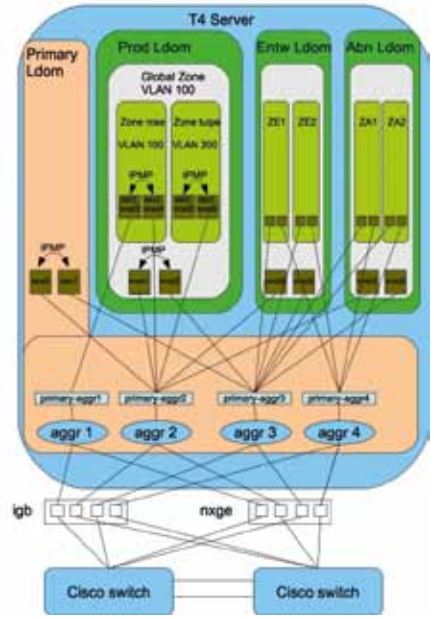

Abbildung 4: Übersicht über das Netzwerk

sich den Rest. In dieser Konfiguration müssen die virtuellen Switches im Hypervisor in der Lage sein, mehrere VLANs zu managen.

Konfigurations-Beispiele

Die folgenden Beispiele zeigen, wie die oben beschriebene Konfiguration erstellt werden kann. Sie gelten für Oracle VM Server SPARC 3.0 und Solaris 10:

- *Konfiguration Link Aggregation 802.3ad*
  Es wurden verschiedene „load balancing policies" getestet und sich am Schluss für „L2" entschieden. Damit wird das „outbound device" gemäß der MAC-Adressen in den Paketen selektiert. Das „link aggregation control protocol" (LACP) wurde nicht konfiguriert („off mode"), was dem Solaris-Default entspricht. Listing 1 zeigt, wie auf der „Primary Domain" ein solches Aggregat mit dem „dladm"-Command erzeugt werden kann.
- *Netzwerk-Konfiguration im Hypervisor*
  Das Beispiel zeigt wie zwei virtuelle Switches auf den vorher erstellten Aggregaten, um die VLANs 100, 200, 300 und 400 zu managen. Mit der „vid"-Property wird bestimmt, dass der Switch das VLAN-Tagging für die entsprechenden VLANs durchführen soll (siehe Listing 2). Es ist von Vorteil, die Mac-Adressen selber eindeutig zu vergeben. Der „Logical Domain Manager" ist zwar in der Lage, diese automatisch zu vergeben, erkennt jedoch nicht die Adressen weiterer LDoms auf anderer Hardware. Anschließend erstellt man für die eben erzeugten Switches Vnets, die dann für die IPMP-Konfiguration verwendet werden können. Die „pvid"-Property bestimmt, zu welchem VLAN das Vnet gehören soll. Wichtig ist das Property „linkprop=phys-state". Es wird gebraucht, um den Link-Status der physischen Netzwerk-Devices an die virtuellen durchzureichen. IPMP kann nur dann funktionieren, wenn ein physischer Link-Fehler an die virtuellen Devices weitergegeben wird (siehe Listing 3).
- *Link based IPMP*
  Dieses Beispiel zeigt, wie IPMP mit „link-based failure detection" und „active active mode" für die Interfaces „vnet3" und „vnet4" konfiguriert werden kann. Der verwendete Hostname ist „tulpe", der Name der IPMP-Gruppe „pz1". Für das erstes NIC-File „/etc/hostname. vnet3" nimmt man „tulpe netmask + broadcast + group pz1 up" und für das zweite NIC-File „/etc/hostname. vnet4" heißt es „group pz1 up".

```
dladm create-aggr -d nxge0 -d igb0 1
dladm modify-aggr -P L2 1
```

Listing 1

```
ldm add-vsw mac-addr=00:14:4f:fc:00:00 vid=100,200,300,400 net-
dev=aggr1 primary-aggr1 primary
ldm add-vsw mac-addr=00:14:4f:fc:00:01 vid=100,200,300,400 net-
dev=aggr2 primary-aggr2 primary
```

Listing 2

```
ldm add-vnet mac-addr=00:14:4f:fc:00:03 linkprop=phys-state pvid=100 excl_rose_1 primary-aggr1 prodldom
ldm add-vnet mac-addr=00:14:4f:fc:00:04 linkprop=phys-state pvid=100 excl_rose_2 primary-aggr2  prodldom
ldm add-vnet mac-addr=00:14:4f:fc:00:05 linkprop=phys-state pvid=200 excl_tulpe_1 primary-aggr1 prodldom
ldm add-vnet mac-addr=00:14:4f:fc:00:06 linkprop=phys-state pvid=200 excl_tulpe_2 primary-aggr2 prodldom
```

Listing 3

Fazit

In einer virtuellen Umgebung kommt man kaum darum herum, Netzwerk-NICs zu bündeln. In diesem Beispiel ist aufgezeigt, wie die Technologien „Link Aggregation“ und IPMP kombiniert werden können. In Zusammenarbeit mit den Kollegen vom Netzwerk-Team wurde mit der oben beschriebenen Konfiguration eine gute Lösung gefunden. Die Systeme sind redundant und mit optimaler Performance am Netzwerk angeschlossen. Auch im Fall von Wartungen im Netzwerk-Bereich können die Systeme ohne Unterbrechung weiterbetrieben werden. Weil „Load Balancing“ implementiert ist, zeigten die Messungen eine optimale Verteilung der Netzwerk-Last über die Netzwerk-Ports. Auch der Durchsatz entsprach den Erwartungen.

Es lohnt sich, im Vorfeld genug Zeit zu investieren und ein gutes Netzwerk-Design auszuarbeiten. Wichtig sind auch ausführliche Tests der möglichen Varianten.

Literatur und Links

- https://blogs.oracle.com/droux/entry/link_aggregation_vs_ip_multipathing
- http://docs.oracle.com/cd/E19253-01/816-4554/
- http://docs.oracle.com/cd/E37707_01/html/E29665/preface.html
- http://www.ieee802.org/3/hssg/public/apr07/frazier_01_0407.pdf

1. http://standards.ieee.org/findstds/standard/802.1Q-2011.html

Roman Gächter
Roman.Gaechter@trivadis.com

Ein knappes Jahr ist seit der Veröffentlichung von Oracle VM 3.1 für x86 vergangen. Dies soll Anlass für ein Review und den Überblick über wesentliche Features der aktuellen Version sein.

OVM 3 (x86) – was sich getan hat

Dirk Läderach, Robotron Datenbank-Software GmbH

Während die ersten Versionen (3.0.1 bis 3.1) bei vielen Anwendern nach Tests oder Upgrades für Ablehnung beziehungsweise Schmunzeln bis Verärgerung sorgte, kann mittlerweile von vielen Seiten eine stete, schrittweise positive Resonanz im Umgang mit der Virtualisierung-Lösung beobachtet werden. Im letzten Jahr hat die DOAG eine Liste der aus Sicht ihrer Mitglieder aktuellen Probleme veröffentlicht und auch Oracle diesbezüglich um Stellung gebeten. Viele dieser Themen sorgten auch bei unseren Kunden für eine schleppende Akzeptanz der Lösung. In der Zwischenzeit ist die Version 3.2.2.520 verfügbar und viele der Mängel sind behoben oder es existiert zumindest ein zufriedenstellender Workaround.

Ein paar positive Beispiele

Upgrade-Probleme gehören seit der Version 3.1.1. nahezu der Vergangenheit an und mittlerweile gibt es auch ein funktionierendes Rollback. Bei fünfzehn vom Autor selbst durchgeführten Migrationen auf 3.2.1 und 3.2.2 hat eine einzige nicht funktioniert; diese wurde sauber wieder zurückgerollt und es erfolgte ein „redeploy“ der Anwendung.

Einbinden von ISO-Dateien funktioniert zwar weiterhin nur über den Umweg des OVM-Hosts, aber die Import-Limitierung (http, ftp) ist inzwischen einem funktionierenden Repository-Import gewichen, der über eine Aktualisierung des Repository gestartet wird. Diese Funktionalität ist übrigens auch in der Lage, per „scp“ in das Repository kopierte Objekte wie virtuelle Disks, Templates, Assemblies bis hin zu den Konfigurationsdateien der VMs zu importieren (wenn die Syntax stimmt) und über die Oberfläche bereitzustellen.

Das oft genutzte Feature der Hard-Partitionierung mittels CPU-Pinning erforderte anfangs noch manuelles Editieren der VM-Konfigurationsdatei („vm.cfg“). Ebenfalls genutzt werden konnten die OVM-Utils. Im aktuellen Release besteht nun mit den CPU-Pools eine sehr gelungene, automatisierte Variante, das CPU-Pinning für die jeweilige Lizenzierung und/oder die Performance-Ansprüche umzusetzen. Für die Oracle Database Appliance ist diese Methode Pflicht.

Auch die Integration in den Enterprise Manager Cloud Control, der Voraussetzung für eine rollenbasierte Nutzerverwaltung ist, und das OPS-Center

```
[root@ovm01 /]# dmidecode|grep UUID
UUID: 20BDCCAA-D378-4C3E-B968-74AB09200A4E

/etc/ovs-agent/agent.ini:
…
[server]
fakeuuid=20BDCCAA-D378-4C3E-B968-74AB09200A4E
…
```

Listing 1

funktioniert in der Version 3.2.2 nahezu reibungslos. Allerdings hätte man sich einen Hinweis im Upgrade-Guide oder in den Release-Notes darüber gewünscht, dass bei Verwendung des Cloud Control 12c R1 einiges zu tun ist, wenn das Plug-in für den OVM Manager nach dem Update auf 3.2.2.520 weiterhin funktionieren soll.

Was noch fehlt oder noch nicht zufriedenstellend umgesetzt ist

Man sollte niemals versuchen, einen Clustered-Server-Pool mit nur einem Host zu betreiben. Fällt dieser wider Erwarten einmal aus und wird komplett (oder nur das Mainboard) ausgetauscht, kann man sich schon mal eine geeignete Strategie zur erneuten Inbetriebnahme des Server-Pools überlegen. Was einfach klingt und beim Wettbewerber VMware relativ unproblematisch verläuft, kann sich auch in der aktuellsten Version zu einem „Trial & Error"-Szenario entwickeln. Folgende Strategie hat sich in der Praxis bewährt.

Bei Verwendung der vorherigen Server-Namen und IP-Adressen sowie in der Hoffnung, dass der FC-HBA wiederverwendet werden kann, ist es erfolgversprechend, den Host zu „discovern" und die Konfigurationsdatei des Agent („/etc/ovs-agent/agent.ini") mit der über „dmidecode" ermittelten „UUID" zu bearbeiten (siehe Listing 1). Wenn das nicht funktioniert, muss die lokale RPM-DB neu erstellt werden (siehe Listing 2).

```
cd /var/lib
rm __db* && rpm -rebuilddb
reboot
```

Listing 2

Darüber hinaus konnten gute Erfahrungen mit dem Storage-Plug-in von Fujitsu gesammelt werden. Dieses funktionierte praktisch sehr gut und konnte die gestellten Aufgaben wie Anlegen von LUNs, Volumes, Host-Affinity-Groups sowie Snapshots mittels der OVM-Manager-Oberfläche fehlerfrei erledigen. Lediglich der Snapshot von virtuellen Disks mit Thin Provisioning funktionierte nachweislich nicht, hinterher war immer das „root"-Dateisystem des Klons defekt.

Wer bereits mit OVM 2.x gearbeitet hat, kennt die Rolle des Utility-Servers, der in Version 3.1.1 wieder eingeführt wurde. Dadurch ist es möglich, einzelnen OVM-Hosts explizit die Rollen „VM Server" und „Utility Server" zuzuweisen. Server mit der Rolle des „Utility Server" werden für I/O-intensive Operationen (wie das Importieren von Templates) bevorzugt verwendet.

Einige Verbesserungen hat auch die Oberfläche erfahren. So ist es nun beispielsweise möglich, durch Mehrfachauswahl viele VMs auf einmal zu starten oder zu stoppen. Leider hat dieses Feature die Live-Migration noch nicht erreicht. Hier kann weiterhin nur sequenziell gearbeitet werden, beim Wettbewerber VMware ist die Migration von bis zu acht VMs gleichzeitig möglich.

Die OVM-Shell

Mit der Version 3.2.1 hielt die finale Version des OVM Command Line Interface (CLI) sowie der Shell Einzug. Damit ist nun endlich eine gut funktionierende Schnittstelle (API) für Skripte etc. vorhanden. Diese sollte aus Sicht des Autors stetig erweitert werden, um in Zukunft vielleicht den einen oder anderen Hersteller von Backup-Software zu animieren, diese Schnittstelle in der eigenen Lösung zu nutzen. Aktuell gibt es noch keine Funktionalität eines „Snapshot mit Redo-Log-Funktionalität", wie es etwa VMware bietet. Dennoch kann mit der CLI die Backup/Restore-Funktionalität mit eigenen Skripten einigermaßen umgesetzt werden.

Snapshots eignen sich nur bei VMs, bei denen die Anwendungen ihre Daten nicht primär im Hauptspeicher vorhalten und bearbeiten. Besonders Application- und File-Server lassen sich damit gut und vor allem konsistent sichern. Bei Oracle-Datenbanken kann durch einen Snapshot allein kein konsistentes Backup erzielt werden, dies ist als alleiniges Sicherungskonzept daher völlig ungeeignet.

Beim Backup mit der OVM-Shell wird bei Verwendung des „OCFS2"-Dateisystems ein „Reflink"-Klon aller Disks der VM erzeugt. Die Verwendung der CLI hat zu den bisher gezeigten Varianten mittels „Reflink" (siehe DOAG

```
OVM> clone Vm name=oel6_wls01 destType=VmTemplate destName=oel6_wls01_backup serverPool=prod_pool01
Command: clone Vm name=oel6_wls01 destType=VmTemplate destName=oel6_wls01_backup serverPool=prod_pool01
Status: Success
```

Listing 3

```
OVM> importVirtualDisk repository name=backup server=ovm01 url='http://nfs-server/ovmbackup/oel_backup01.
img'
Command: importVirtualDisk repository name=ovm_repo1tb server=ovm01 url='http://nfs-server/ovmbackup/oel_
backup01.img'
Status: Success
Time: 2013-04-15 15:51:12.512
OVM> create VM name=oel6_revover repository=Repo01 domainType=XEN_PVM memory=1024 on Server name=ovm01
Command: create VM name=oel6_revover repository=Repo01 domainType=XEN_PVM memory=1024 on Server name=ovm01
Status: Success
Time: 2013-05-15 16:04:06.071
OVM> create vmDiskMapping name=recoverMap1 slot=1 storageDevice=oel_backup01.img on vm name=oel6_recover
Command: create vmDiskMapping name=recoverMap1 slot=1 storageDevice=oel_backup01.img on vm name=oel6_recover
Status: Success
```

Listing 4

News, Ausgabe 03/2012) den entscheidenden Vorteil, dass lediglich der Serverpool und der VM-Name bekannt sein müssen. Die bisherigen Backup-Implementierungen mit „Reflink"-Technologie im Monitoring- und Backup-Tool „robotron*DBAcheck" mussten noch mühsam feststellen, auf welchem Host die VM läuft und ob es sich um Shared Disks handelt. Da die CLI auf dem Manager läuft, kann natürlich auch auf alle Informationen des Manager zugegriffen werden (siehe Listing 3).

Wenn das Repository per NFS zugänglich gemacht wird, kann die Konfigurationsdatei der zu sichernden VM im Template-Verzeichnis lokalisiert werden. Nun gilt es, die zugehörigen Disks zu ermitteln. Das kann sehr einfach mit „grep", „sed", „cut", „awk" etc. erledigt werden. Dabei ist der originale Pfad durch den NFS-Mount-Punkt mittels „/OVS/Repositories/id/VirtualDisks/ -> /nfs/archiv/ovm-backup/VirtualDisks/" zu ersetzen. Das Ergebnis stellt die Quelle für das Sichern der virtuellen Disks auf ein externes Medium dar. Darüber hinaus kann die Konfigurationsdatei sehr leicht im Template-Verzeichnis gesichert werden. Das ist hauptsächlich für die MAC-Adresse interessant.

Beim Restore gibt es verschiedene Varianten. Die virtuellen Disks müssen auf jeden Fall wieder in das Repository und dort muss eine VM entstehen. Eine Möglichkeit zum Instant Recovery (Start der VM aus dem Backup-Verzeichnis) gibt es momentan noch nicht. Dieses Skript kann sehr einfach implementiert werden und erstellt aus der Sicherung eine neue VM (siehe Listing 4).

Die virtuelle Netzwerkkarte kann ebenfalls über die CLI zugewiesen werden. Falls die originale MAC-Adresse notwendig ist, kann diese ebenfalls mithilfe von „find . -type f -exec sed -ie ‚s#alter Wert#neuer Wert#g' {} \;" über das Restore-Skript ausgetauscht werden.

Fazit

Oracle VM ist schrittweise immer besser geworden und kann inzwischen auch als gut benutzbar betrachtet werden. Wer sich an ein paar Regeln hält, wie keinen Stand-Alone-Serverpool zu betreiben oder nicht zu exzessiv mit VLANs und PXE-Boot zu arbeiten, kann durchaus auch über einen Einsatz im Rechenzentrum nachdenken. Allerdings gibt es noch keine Lösung, um eine Windows-Installation zu Oracle VM zu portieren. Hier kann weiterhin nur der VMware-Konverter genutzt werden, der die Installation nach OVF portiert, das dann vom Manager wiederum in Form von Assemblies verstanden wird. Ironischerweise gibt Oracle das mittlerweile auch selbst als Workaround heraus.

Die OVM Shell hat ebenfalls noch Potenzial. Hier würde man sich beispielsweise ein alternatives Zielverzeichnis für VM-Kopien wünschen. Die VM-Messages bieten viele Möglichkeiten bei der Konfiguration und Kommunikation mit VMs ohne direkten Zugriff auf die VMs selbst.

Dirk Läderach
dirk.laederach
@robotron.de

Oracle VM – die Virtualisierungslösung von Oracle – besteht aus zwei Komponenten. Einerseits dem OVM Server, dem Host für die virtuellen Maschinen, und andererseits dem OVM Manager, der Komponente zur Verwaltung des OVM Servers.

Backup einer Oracle-VM3-Umgebung

Martin Bracher, Trivadis AG

Das Thema „Backup und Recovery“ ist in den Oracle-Dokumentationen nur sehr rudimentär beschrieben. Dieser Artikel betrachtet das Thema etwas genauer und zeigt, was man wo und wie sichern muss. Die zu sichernden Komponenten sind zum einen die Infrastruktur, auf der die VMs laufen (OVM Server Installation (Disk), Pool Filesystem, Repository Filesystem und optional die in VMs verwendeten LUNs). Zum anderen der OVM Manager (lokale Disks (OS/Software) und die Repository-Datenbank) sowie die virtuellen Maschinen (VM) selbst (Definition, Disk-Files und die Daten in der VM, siehe Abbildung 1).

Der OVM Server

Hier muss beim Backup der Host selbst gesichert werden, also das Betriebssystem, das sich normalerweise auf lokalen Disks befindet. Es ist jedoch nicht möglich, zusätzliche Software auf einem OVM Server zu installieren. Man kann aber lokal vorhandene Tools („tar“, „gzip“, „cp“, „scp“ etc.) verwenden. Dieses Backup kann man per „scp“ wegkopieren oder auf dem Repository Filesystem ablegen und dort via NFS abholen: „tar --one-filesystem –zcvf /OVS/repository/<uuid>/root_$HOSTNAME.tgz /“.

Ein zeitgesteuertes Backup muss man beispielsweise vom OVM Server aus über „ssh“ starten, da auf dem OVM Server kein „cron“ installiert ist. Man kann aber auch die Strategie verfolgen, kein Backup des Servers zu machen, sondern ihn im Problemfall einfach aus der Konfiguration zu löschen, danach neu zu installieren und wieder in den Pool einzufügen.

Pool Filesystem

Das Pool Filesystem ist das gemeinsame Filesystem eines Cluster-Pools. Es ist somit ein Shared Filesystem mit OCFS2. Es befindet sich auf einem SAN, iSCSI oder auf NFS. Wenn es auf NFS liegt, ist es eine etwa 12 GB große Datei, die ein OCFS2-Filesystem enthält und auf den Servern Loopback „gemountet“ ist.

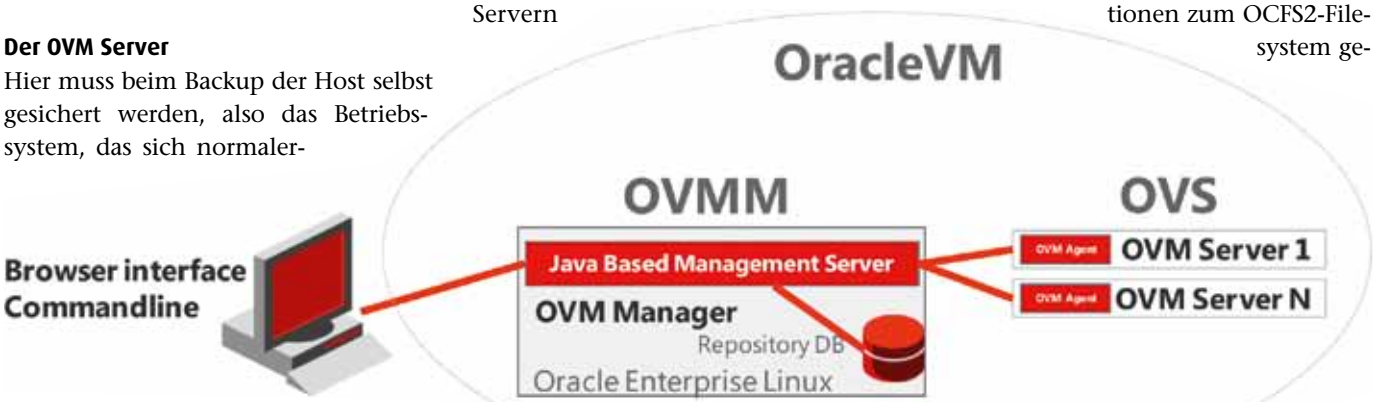

Abbildung 1: Die Oracle VM

Dieses Filesystem beinhaltet die Informationen zum Cluster und wird zum Handling der Node-Membership benötigt. In der Terminologie von RAC hat es die Funktionalität der Cluster-Registry und der Voting Disks.

Dieses Filesystem ist initial nach der Installation bei gestopptem Cluster zu sichern. Bei NFS kopiert und komprimiert man die erwähnten Datei, ansonsten wird mit dem „dd“-Befehl die entsprechende LUN mit „dd if=/dev/sdb bs=1M | gzip >pool_fs.gz“ wegkopiert und ebenfalls komprimiert. Es bleiben erfahrungsgemäß 10 bis 15 MB übrig. Danach kann man im regulären Betrieb ein regelmäßiges Backup des Filesystems etwa mit „tar“ erstellen.

Repository Filesystem

Auf dem Repository Filesystem sind die virtuellen Maschinen abgelegt (Definition und virtuelle Disks). Es ist ebenfalls „shared“ zwischen den Servern und entweder ein NFS Share oder ein OCFS2-Filesystem. Zu Beginn sollte man die Verzeichnisstruktur sowie die Datei „.ovsrepo“ sichern.

Darüber hinaus werden die Informationen zum OCFS2-Filesystem gesichert (siehe Listing 1).

Mit diesen Informationen lässt sich dann das Filesystem notfalls wieder neu erstellen (siehe Listing 2).

Regelmäßig sollten nun die Dateien gesichert werden – sowohl die Definitions-Files in den Verzeichnissen „VirtualMachines“, „Templates“ und „Assemblies“ als auch die Dateien in ISOs. Die Dateien in VirtualDisks können nur gesichert werden, wenn sie nicht in Verwendung sind. Wie man geöffnete Files sichern kann, folgt später.

Bei Verlust des Filesystems muss die-

```
# tunefs.ocfs2 -Q „B=%B, T=%T, N=%N V=%V U=%U\n“ /dev/mapper/1IET_00010001
b=4096, T=131072, N=32 V=OVS56e8973883d87 U=0004FB00000500002DC56E8973883D87
```

Listing 1

ses also neu erzeugt werden, danach ist das Backup der Directory-Struktur sowie der (statischen) Files durchzuführen. Zum Schluss sind die Snapshots (siehe später) wieder an den Platz der Original-Diskdateien zu kopieren.

OVM Manager

Der OVM Manager wird auf einem RedHat- oder Oracle-Enterprise-Linux installiert. Es sollte ein Backup der ganzen Filesysteme erstellt werden, wie für jeden produktiven Server auch. Bei Verlust können diese Dateien vom Backup zurückgespielt werden. Der Manager kann eine existierende Oracle-Standard- oder Enterprise-Edition-Datenbank als Repository-Datenbank verwenden. Es wird nur ein zusätzliches Schema (OVS) installiert. Diese Datenbank kann sich lokal auf dem Manager Host oder auf einem anderen Server befinden.

Bis OVM3.1 wurde für Testzwecke eine Oracle XE Edition mitgeliefert (für produktive Systeme nicht unterstützt, läuft aber perfekt). Ab Version 3.2 wurde diese leider durch MySQL ersetzt. Wieso leider? Die meisten Benutzer verwenden OVM für Oracle-Datenbanken, also was soll man mit so einem Exoten, den kaum ein Oracle-DBA bedienen kann? Für den Autor ein absolutes No-Go.

Die Oracle-Datenbank kann wie gewohnt mit „rman“ oder Export gesichert werden. Falls die Datenbank verloren geht, kann sie mit den dem Oracle-DBA bekannten Verfahren wiederhergestellt werden. Wenn man die letzten Transaktionen verloren hat, ist das nicht schlimm. Der Manager ist tolerant genug, auch mit einem etwas älteren Backup-Stand weiterarbeiten zu können. Er kann die geänderten Daten von den OVM Servern zurücksynchronisieren. Vor dem Restore sollte man den Manager stoppen („/etc/init.d/ovmm stop“) und nach dem Restore wieder starten („/etc/init.d/ovmm start“).

Virtuelle Maschinen

Eine VM besteht aus zwei Teilen: Erstens dem Storage, also (virtuellen) Disks. Dies können eine Datei oder ein Block-Device auf dem OVM Server sein, per Netzwerk ein externer NFS Share oder eine iSCSI LUN. Zweitens aus der virtuellen „Hardware“ ein Konfigurationsfile, in dem die CPU, Netzwerkkarten, Memory etc. definiert sind.

Beim Backup müssen wir zwischen einem Backup der gesamten VM (zwecks „Bare-Metal“-Recovery) und einem Backup des Inhalts der (virtuellen) Disks (zwecks Restore einzelner Filesysteme/Dateien) unterscheiden. Für die virtuelle Hardware ist die Datei „<repo-fs>/VirtualMachines/<vm-uuid>/vm.cfg“ zu sichern. Sie wird zwar aus den Informationen im Repository erzeugt, jedoch zum Restore des Repository Filesystems benötigt. Seit OVM ein vernünftiges Commandline-Interface zur Verfügung stellt, sollten wir die Befehle zur Erstellung der VM ebenfalls speichern (siehe Listing 3). Dieses Script würde man brauchen, wenn man die gesamte Umgebung neu aufbauen müsste.

Falls die Disks eine LUN auf dem OVM Server sind, können sie bei gestoppter VM mit dem „dd“-Befehl gesichert werden. Falls der Storage eine Möglichkeit für Snapshot bietet, könnte dies sogar online erfolgen.

Wenn die virtuellen Disks ein File auf dem Storage-Repository sind, kann dieses gesichert werden. Falls es sich um SAN-/iSCSI-Storage mit einem OCFS2-Filesystem handelt, können diese Files auch durch einen Snapshot gesichert werden, der anschließend auf ein anderes Medium wegkopiert wird. Der Snapshot selbst ist noch kein Backup, denn er befindet sich immer noch auf demselben Storage.

Um einen solchen Snapshot zu erstellen, kann man aus der VM ein Template klonen und danach die (nicht geöffneten) Files dieses Klons wegsichern. OVM bietet die Möglichkeit, das Storage-Repository per NFS zu exportieren. Man erinnere sich: Es ist nicht zertifiziert, Backup-Software auf dem OVM Server zu installieren. Aber wir können die Dateien via NFS sichern.

Der Restore einer VM erfolgt dann, indem man diese stoppt und danach die Diskfiles der VM durch die gesicherten Snapshot-Files des Klons ersetzt. Danach ist die VM neu zu starten und sie enthält wieder den Zustand vom Zeitpunkt des Snapshots.

Zum Sichern einzelner Filesysteme oder einzelner Files sowie von Datenbanken innerhalb der VM unterscheidet sich das Backup-Konzept nicht von dem eines physischen Hosts. Man installiert innerhalb der VM entsprechende Backup-Software und sichert die Dateien auf Storage oder direkt auf ein Tape.

Ein Problem ist, dass Backup-Clients häufig pro Hostname lizenziert sind. Mit etwas Scripting-Aufwand lassen sich diese Kosten umgehen: Man erstellt einen Snapshot der Disk-Datei, hängt diesen dann an eine VM mit lizenziertem Backup-Client und macht den Backup von dort aus. Wenn man, wie oben beschrieben, jeweils die gesamte VM mit Snapshots sichert, kann man sich den Filesystem-Backup auch sparen, bei Bedarf den Snapshot „read-only“ hinzufügen und die Dateien von dort zurückkopieren.

Spezielle Recovery-Situationen

- *Verlust der Repository-Datenbank (ohne Backup)*
  In dem Fall muss eine neue Datenbank ohne „OVS“-Schema erzeugt werden. Ebenso ist der OVM Manager neu zu installieren, und zwar per „--uuid 0004EC000000100001C87-

```
mkfs.ocfs2 -J block64 -b 4096 -L OVS56e8973883d87 \
   -U 0004fb00000500002dc56e8973883d87 -T vmstore \
   -N 32 /dev/mapper/1IET_00010002
```

Listing 2

```
create Vm name=slot011 repository=slotreposan01 domainType=XEN_PVM osType=OL_5 bootOrder=DISK cpuCount=2
highAvailability=yes memory=2048 on ServerPool name=slot

create Vnic name=00:21:f6:01:00:0b network=vlanpublic

add Vnic name=00:21:f6:01:00:0b to Vm name=slot011

create VmDiskMapping slot=0 storageDevice=slot011_xvda name=xvda on Vm name=slot011
```

Listing 3

C2AEE23B7“ mit der bisherigen UUID „./runInstaller“. Diese ist beispielsweise in „/etc/sysconfig/ovmm“ hinterlegt. Nach dieser Neu-Installation können nun die bestehenden OVM Server „rediscovered“ werden. Da wir immer noch mit derselben UUID arbeiten, beginnt der Manager, sein Repository aufgrund der auf dem Server gespeicherten Informationen zu rekonstruieren.

- *Verlust des OVM Manager (ohne Backup), Datenbank noch vorhanden*
  Wenn der Backup fehlen sollte, kann man den Manager neu installieren. Zu diesem Zweck muss man aber zuerst das noch existierende OVS-Schema exportieren, dieses dann löschen und danach den Manager mit der alten UUID neu installieren. Nach der Installation stoppt man den Manager nochmals, löscht das neu erstellte OVS-Schema und importiert den Backup des alten OVS-Schemas wieder. Nach dem Start des Managers steht der bisherige Zustand wieder zur Verfügung.
- *Verlust von OVM Manager und Datenbank (ohne Backup)*
  Wenn beide Komponenten nicht gesichert wurden, installiert man eine neue Datenbank und einen neuen Manager mit der alten UUID. Diese findet man beispielsweise auf dem Repository Filesystem in der „.ovsrepo“-Datei. Nach einem Re-Discover der bestehenden Server sollte alles wieder weitgehend in Ordnung sein.
- *Desaster-Szenario: Verlust der ganzen Umgebung*
  Mit den Backups der Snapshot-Files und den Create-Scripts für die VM lässt sich eine neue Umgebung aufbauen. Dazu eine Neu-Installation/-Konfiguration von OVM Manager und Servern durchführen. Danach Kopieren der Diskfiles vom Backup ins Repository unter VirtualDisks und Durchführen eines Re-Scans des Repository. Diese Disks sind dem Manager danach wieder bekannt. Jetzt müssen nur noch die VMs neu definiert und die Diskfiles zugewiesen werden.

Fazit

Im Prinzip müssen sich die Backup-Konzepte von physischen Servern und OVM praktisch nicht unterscheiden. Der (virtuelle) Server wird mit klassischen Backup-Clients gesichert. Wenn man nun einen physischen Server oder eine VM verliert, besorgt man sich einen neuen physischen/virtuellen Server und installiert diesen neu beziehungsweise macht einen Restore des Backups. OVM bietet uns aber einige zusätzliche Möglichkeiten, um das Backup-Konzept zu optimieren oder zu erweitern, wie beispielsweise die Möglichkeit für Snapshots ganzer Diskfiles.

Der Verlust des OVM Manager ist kein großes Problem. Die Server und VMs laufen unabhängig von diesem weiter und der Manager kann aufgrund der Informationen auf den Servern weitgehend wieder neu aufgebaut werden.

Martin Bracher
martin.bracher@trivadis.com

Neu: ADF Mobile 1.1 ist verfügbar

Mit dem neuen JDeveloper-Release 11.1.2.4 hat Oracle nicht nur zahlreiche Bugs behoben, sondern auch die Version 1.1 des ADF Mobile Frameworks mit neuen Features veröffentlicht. Neben der Unterstützung der aktuellen Phone- und Tablet-Versionen gestaltet sich die Entwicklung mobiler Unternehmensanwendungen einfacher. Die ersten Testdrives weisen zudem eine verbesserte Performance und Stabilität auf.

Das ADF Mobile Framework ist als JDeveloper-Extension verfügbar und unterstützt mittels Single-Source-Ansatz die Entwicklung von iPhone und Android Apps auf Basis von HTML5 und Java. Durch die Verwendung des Cordova-Frameworks (ehemals PhoneGap) wird zudem eine Vielzahl nativer Geräte-Services unterstützt.

Zu den Neuheiten zählen Push-Benachrichtigungen und die sogenannten „Badges“. Dabei handelt es sich um Benachrichtigungssymbole, die die oftmals in Kombination mit Push-Benachrichtigungen zum Einsatz kommen und die Anzeige der Anzahl neuer Anwendungsevents ermöglicht. Zudem lassen sich nun Dateien über eine Dateivorschau mit der nativen Vorschau-App anzeigen.

Eine neue Funktionalität der Version 1.1 ist die Bereitstellung vom Mobile Application Archive (MAA). damit ist es möglich, eine Basis-Applikation in Form einer Vorlage anderen ADF-Mobile-Projekten zur Verfügung zu stellen. Das konsumierende Projekt kann damit bestimmte Elemente anpassen und als App-Rahmen für neue Apps mit den eigenen Zertifikaten als Herausgeber verwenden.

Zudem hat Oracle die technische Infrastruktur geändert. Neben einer verbesserten Performance ist nun auch eine iPhone5- und iPad-Mini-Unterstützung möglich.

Weitere Informationen unter http://www.doag.org/home/aktuelle-news/article/oracle-adf-mobile-11-ist-verfuegbar.html

„ETL vs. ELT" ist nach wie vor eines der Reizthemen bei der Diskussion von BI-Architekturen. Die Diskussionen beziehen sich dabei in der Regel auf den Schritt der Beladung des DWH (Staging) und der Weiterverarbeitung in das DWH-Modell (Integration Layer). Etwa bei der Erstellung der Data Marts für die BI-Applikationen erhält das Daten-Management dagegen weniger Aufmerksamkeit. Nicht selten sind Performance-Probleme die Folge, die mit viel Aufwand (und Hardware) wieder beseitigt werden müssen. Betrachtet man BI-Applikationen nur als ETL-Prozesse, öffnen sich durch den veränderten Blickwickel auf das Daten-Management unentdeckte Möglichkeiten zur Lösung von Problemen.

Applikationsanbindung an das Data Warehouse: ETL vs. ELT

Dr. Gernot Schreib, b.telligent GmbH & Co. KG

Die Planung von Datenflüssen in DWH-Landschaften setzt ein Architektur-Prinzip von Schichten und Ebenen voraus. Aus den Erfahrungen der letzten zwanzig Jahre hat sich eine Standard-Architektur als praxistauglich erwiesen. Die Layer „Source Abstraction (Stage)", „Integration (Core)" und „Presentation (KPI)" dürfen in einer modernen DWH-Architektur nicht fehlen (siehe Abbildung 1: BI-DWH-Architektur).

Der Source-Abstraction-Layer hat die Aufgabe, die Quelldaten aufzunehmen und in eine für die weiteren Verarbeitungsschritte einheitliche Schnittstelle zu überführen. Der Integration-Layer bildet den fachlichen Kern des Datenmodells ab und modelliert idealerweise unabhängig von den Datenquellen (und Anwendungen) die fachlichen Entitäten auf granularer Ebene. Der Presentation-Layer schließlich stellt für die Applikationen (gegebenenfalls unter Berücksichtigung von Benutzer- und Rollen-Konzepten) eine einfach handhabbare, performante Zugriffsschicht auf das DWH zur Verfügung. In diesem Artikel steht die Umsetzung der Datenflüsse zwischen diesen Layern im Mittelpunkt (siehe Abbildung 1, Nr. 1 bis 4).

ETL vs. ELT

Die Google-Suche mit „ETL vs. ELT" ergibt ca. 270.000 Treffer, davon eine ganze Reihe von Blog-Einträgen. Das Thema wurde in der Vergangenheit und wird aktuell nach wie vor intensiv diskutiert. Interessanterweise ist bei den meisten der Beiträge eine klare Befürwortung beziehungsweise Ablehnung einer der beiden Architekturen zu erkennen. Dies ist insofern erstaunlich, als nahezu jeder der Artikel eine mehr oder weniger vollständige Betrachtung der Vor- und Nachteile enthält, die es erlauben würde, sich fallweise für die eine oder andere Architektur zu entscheiden. Es ist möglicherweise die Zubeziehungsweise Abneigung gegenüber dem Einsatz von Tools, gegebenenfalls auch einem bestimmten Tool, durch die sich die Autoren dann doch zu einer Generalisierung hinreißen lassen und einer der beiden Architekturprinzipien grundsätzlich den Vorzug geben.

Zur Unterscheidung der Reihenfolge Transform/Load beziehungsweise Load/Transform wird in diesem Artikel das Kriterium verwendet, ob die Daten die Datenbank verlassen müssen beziehungsweise ob die Datenbank während der Transformation noch die Kontrolle über das Daten-Management (wie Möglichkeit der Parallelisierung) innehat oder nicht. Während eines ETL-Prozesses dient die Datenbank ausschließlich als Datenspeicher. Das Daten-Management der Transform-

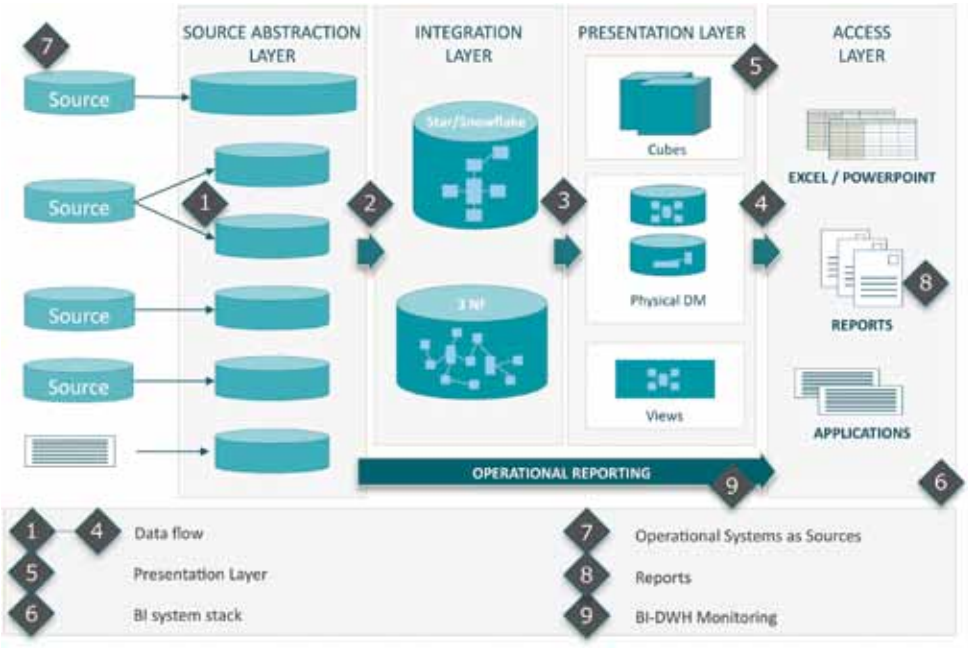

Abbildung 1: BI-DWH-Architektur

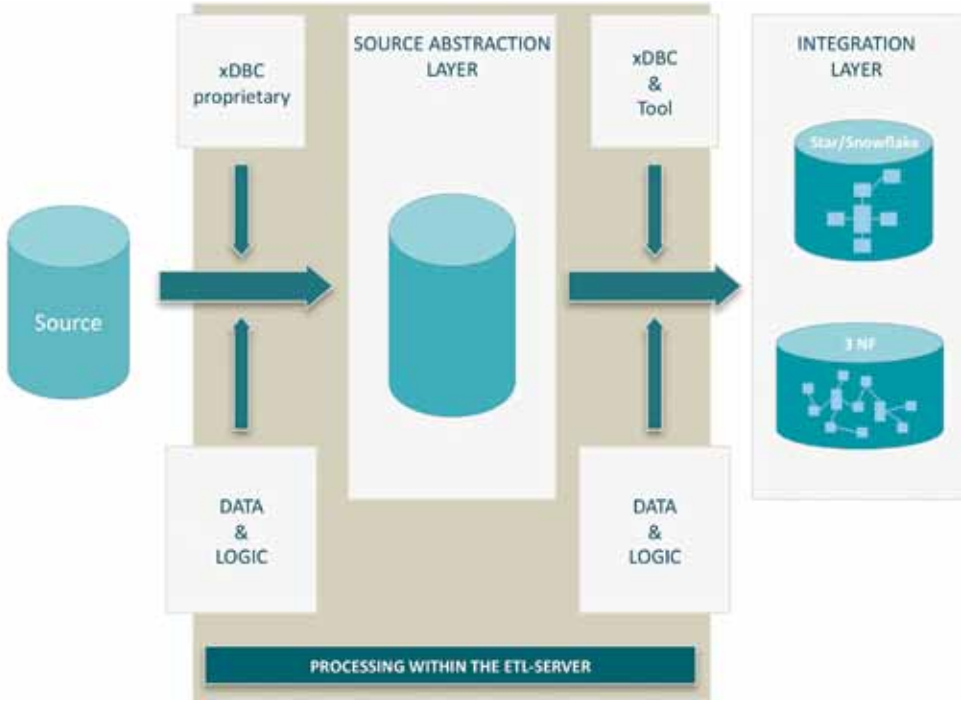

Abbildung 2: ETL-Datenfluss

Operationen wird außerhalb, etwa durch ein externes Tool durchgeführt (siehe Abbildung 2). Die grau hinterlegten Operationen, insbesondere Daten-Management und Logik, liegen außerhalb der Kontrolle der Datenbank.

Im ELT-Datenfluss ist hingegen die Datenbank das treibende System. Die Transform-Operation wird durch eine DML-/SQL-Operation innerhalb der Datenbank durchgeführt. Die notwendige (in der Regel zeilenbasierte) Logik muss durch die Datenbank aufrufbar zur Verfügung stehen (siehe Abbildung 3: ELT-Datenfluss). Die grau hinterlegten Operationen finden unter der Kontrolle der Datenbank statt. Die notwendigen logischen Operationen sind durch Datenbank-interne Prozeduren (gegebenenfalls nach der Nutzung von External-C-/Java-Funktionsaufrufen) realisiert.

Sowohl ETL- als auch ELT-Architekturen haben ihre Berechtigung, je nachdem, welche der Vorteile jeweils für die konkrete Aufgabenstellung günstiger sind. Während oft für die Datenbewirtschaftung der Schnittstelle „Source-Systeme“ zum DWH (Source Abstraction Layer) eine Grundsatzentscheidung getroffen wird (ETL-Tool vs. ELT-Datenbank-Ladung), ist die Situation für die Herstellung des Presentation-Layers differenziert. Eine Gegenüberstellung der Vor- und Nachteile speziell bei diesem Übergang folgt im nächsten Abschnitt.

Applikationen als ETL-Prozesse

Die Begriffe ETL beziehungsweise ELT werden in der Regel für den Prozess des Ladens der Sourcen in das DWH verwendet. Seltener wird der Prozess der Informationsbereitstellung für BI-Applikationen – also der Übergang in den Presentation-Layer – auch als ETL-Prozess bezeichnet, obwohl hier meist genauso viele Daten unter Einsatz von noch mehr Logik bewegt werden. Gerade hier sollte die BI-Architektur aus Performance-, Prozess- und Kosten-Aspekten sehr individuell den Erfordernissen angepasst sein. Beispiele für Anwendungen sind vor allem KPI-Berechnungen als Basis für Reporting, analytisches CRM (wie Segmentierung, Entscheidungsbäume, Regression, neuronale Netze etc.) und operatives CRM (wie Regelverarbeitungen von CRM-Logiken etc.).

Während die Bereitstellung von Basis-KPIs meist selbstverständlich direkt in SQL (und damit in einem ELT-Prozess) realisiert wird, ist es bei komplexen Logiken durchaus üblich, diese in einem externen System (wie SAS-, SPSS-Server, Visual Rules, Talend etc.) abzubilden. Tabelle 1 zeigt die Vor- und Nachteile des ELT-Prozesses.

Der Einsatz von externen Tools (wie SPSS Modeler oder SAS Enterprise Miner) bei komplexen Logiken ist insofern gerechtfertigt, als die komplexen Logiken dort auch entwickelt und selten direkt als SQL implementiert werden. Dies führt dann in der Nomenklatur dieses Vortrags zu einem ETL-Prozess, da die Datenbank als reiner Datenspeicher dient und das Datenmanagement im externen Tool liegt. Tabelle 2 zeigt die Vor- und Nachteile dieses Ansatzes.

Applikationen als ELT-Prozesse

Aufgrund der großen Bandbreite von Applikationen eines BI-DWH ist eine generelle Entscheidung für eine ETL- beziehungsweise ELT-Architektur beim Übergang vom Integration- zum Pre-

VORTEILE	NACHTEILE
• Daten verlassen die DB nicht, d.h. die Verarbeitung der großen Datenmengen (Cursor) findet in der Datenbank statt • Speicherverwaltung der DB wird benutzt • Parallelisierungsoptionen der DB wird benutzt • Transaktionssystem der DB wird benutzt • Reine Rechenoperationen können in (externen) Funktionen / Prozeduren sehr schnell und effizient ausgeführt werden • Ideal für kontext-/nebenläufigfreie Rechenoperationen auf Spalten einer einzelnen Zeile (z. B. Segmentierung, Scoring, decision tree, regression, ...) • Kein zweites Systemumfeld (Komplexitätsreduktion, Kosten)	• Logik muss in SQL / PLSQL codiert werden • DB muss externe Funktionen unterstützen • Bei externen Funktionen i. d. R. nur Operationen innerhalb einer Zeile möglich; keine zeilenübergreifenden Operationen • DB-Server wird belastet (CPU-Last) • keine Redundanz, da nur 1 System

Tabelle 1: Vor- und Nachteile von ELT

sentation-Layer (Nr. 3 in Abbildung 1: BI-DWH-Architektur) noch schwieriger zu treffen als bei der Datenladung des Source-Abstraction- (Stage) beziehungsweise Integration-Layers (Nr. 1 und 2 in Abbildung 1: BI-DWH-Architektur). Möglicherweise ist diese generelle Entscheidung gar nicht sinnvoll und sollte von Applikation zu Applikation separat getroffen werden, um die jeweiligen Vorteile zu nutzen beziehungsweise Nachteile zu vermeiden. Leider wird gerade beim Einsatz eines externen Tools beziehungsweise Servers oft die Möglichkeit, den (operativen) Prozess über einen ELT-Ansatz abzuwickeln, gar nicht berücksichtigt. Warum eigentlich nicht?

Damit eine Überführung der Logik in die Datenbank möglich ist, muss sie zum einen durch das externe Tool unterstützt werden. Dies wird dadurch realisiert, dass die Applikationslogik entweder als SQL oder in einer gängigen Programmiersprache wie Java oder C exportiert werden kann. Zum anderen muss dann innerhalb der Datenbank die Programmlogik des Exports performant zur Verfügung gestellt werden können. Für Oracle ist letztere Bedingung durchweg erfüllt. Ein generiertes SQL ist (meist) kein Problem, aber auch Java-Code kann direkt in Oracle geladen und durch die integrierte Java-Umgebung ausgeführt werden. Schließlich lassen sich dynamische Libraries (C-Code) über PL/SQL-Wrapper mit wenigen Handgriffen einbinden. Als einfachstes Beispiel dient hier eine PL/SQL-Funktion für eine C-Funktion (siehe Listing 1).

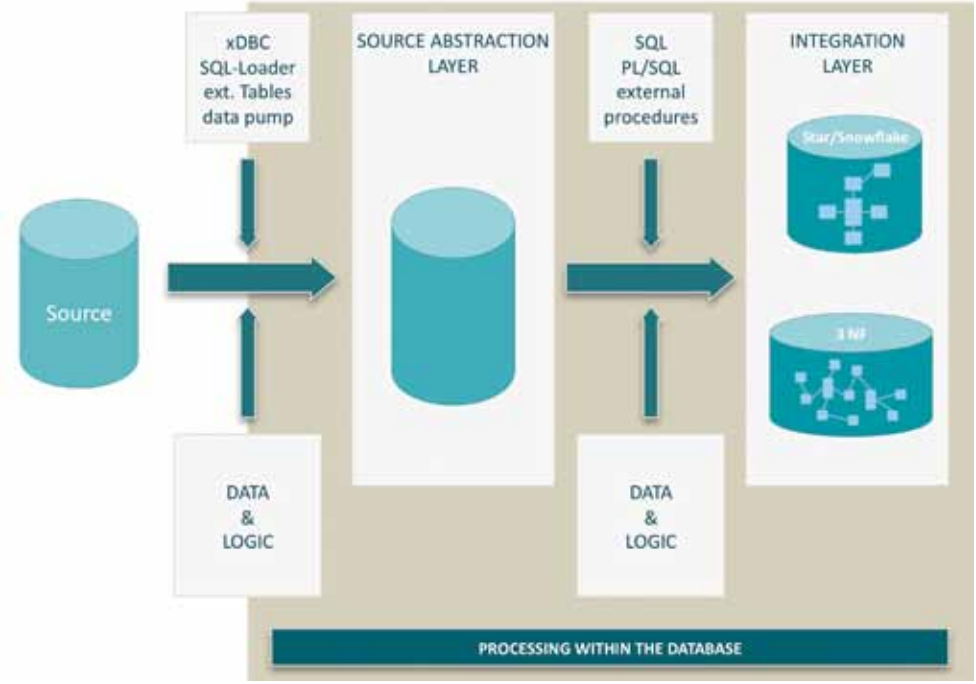

Abbildung 3: ELT-Datenfluss

Dabei ist „<libname>“ ein mit „create library“ angelegtes Oracle-Library-Objekt, das den Pfad und den Namen der Library auf Betriebssystem-Ebene enthält, und „<c-fct-name>“ der Name der C-Funktion innerhalb dieser Library. Die Syntax für das Wrapping von C-Funktionen innerhalb von Oracle ist sehr mächtig und kann deutlich mehr. Dieses Beispiel hier ist der einfachste Fall, zeigt aber auch, dass das Einbinden von C-Funktionen innerhalb von Oracle (mit den Standard-Datentypen „number“ und „varchar2“) sehr einfach umgesetzt werden könnte.

Die Überführung eines ETL-Applikationsprozesses in einen ELT-Prozess scheitert daher meist an der fehlenden Unterstützung durch die Applikations-Tools. Dies ist nicht weiter verwunderlich, da alle Hersteller aus nachvollziehbaren Gründen daran interessiert sind, mit ihren Tools einen möglichst großen Teil der Prozesskette abzudecken. Deshalb werden Funktionalitäten zum Export der Logiken entweder gar nicht oder nur rudimentär und mit Einschränkungen implementiert. Vor allem bei datenintensiven Anwendungen steht dieses Vorgehen klar im Konflikt zum Interesse des BI-Architekten. Aus Architektur-Sicht wäre hier ein ELT-Ansatz viel sinnvoller.

VORTEILE	NACHTEILE

• Unabhängig von Datenbank (nur definierte Schnittstellen wie z. B. ODBC, JDBC Connection notwendig), wenn kein embedded SQL
• Stärken der Programmiersprache im externen Tool voll ausnutzbar
• kein tiefes Datenbank know how notwendig
• Durch Systemtrennung Entlastung von DB-Server möglich
• Redundanz
	• Verbindung zur DB (User / Passwort)
	• größere Datenmengen müssen aus der Datenbank extrahiert werden, extern verarbeitet werden und ggf. wieder geschrieben werden
	• i.d.R. sequentielle Verarbeitung der Daten
	• ggf. embedded SQL notwendig (proprietär, wartungsunfreundlich, fehlende Versionssicherheit)
	• hohe Komplexität und ggf. Kosten aufgrund Einsatz zweier Systeme (DB + 3rd party)
	• Performanceprobleme, falls
	  • 3rd party System und/oder DB nicht parallelisieren können
	  • Datenmenge den verfügbaren Hauptspeicher übersteigt

Tabelle 1: Vor- und Nachteile von ETL

Anwendungsbeispiele

BI-Prozesse finden sich in jedem Unternehmen; die Haupt-Anwendungsgebiete sind Management-Informationssysteme, Analytik sowie Kampagnen-Management und Planung. Erfahrungsgemäß entwickelt sich ein BI-Umfeld auch entlang dieser Themen. Sind diese Haupt-Anwendungen etabliert, benötigen sie dann in etwa die gleichen Ressourcen des BI-DWH.

Unternehmungen mit ausgeprägter B2C-Geschäftstätigkeit betreiben

```
FUNCTION c_fct_wrapper (<var-list>)
  RETURN BINARY_INTEGER
AS LANGUAGE C LIBRARY <libname> NAME "<c-fct-name>";
```

Listing 1

(neben dem Klassiker „Reporting“) intensiv Anwendungen zur Pflege des Kundenstamms. Ohne Anspruch auf Vollständigkeit sind im Folgenden drei wichtige Beispiele beschrieben.

- *Segmentierung*
  Zur Bewertung des Kundenstamms, der Darstellung der Entwicklung der Zusammensetzung sowie der Allokation von Angeboten und Produkten erfolgt eine Segmentierung des Kundenbestands. Dabei wird jedem Kunden gemäß der Segmentierungsregel genau eine Klasse zugewiesen. Dieses Attribut bleibt nun über einen vorgegebenen Zeitraum – etwa Quartal, Halbjahr oder Jahr – für alle Kunden konstant, sodass mithilfe dieser Klassifikation Bewegung zwischen den Segmenten beobachtet, im günstigen Fall aktiv beeinflusst werden kann. Die Anzahl und Beschreibung der Klassen ist – mit Ausnahme der Klasse der „Neukunden“ – sehr individuell und wird den Zielen und Bedürfnissen des jeweiligen Unternehmens angepasst.
- *Analytik und Prognose*
  Der Begriff „Scoring“ ist in der Öffentlichkeit vor allem aus dem Bankensektor bekannt und dort (leider) negativ besetzt. Ganz anders sieht die Situation im BI-Umfeld aus. Scoring bezeichnet konkret die Anwendung statistischer Verfahren zur Prognose des Kundenverhaltens aus bekannten Daten, etwa darüber, ob ein Kunde ein Angebot aus der Werbung (nicht) annehmen wird, oder – in einigen Branchen sogar gesetzlich vorgeschrieben – die Vorhersage des Risikos eines Zahlungsausfalls. Mit Scoring-Verfahren werden also die Eintrittswahrscheinlichkeiten für das betrachtete Ereignis pro Kunde auf Basis von bereits bekannten Kundendaten ermittelt.
- *Kampagnen und Selektion*
  Ein weiteres wichtiges Anwendungsfeld ist das operative CRM. Basierend auf Prognosen über das Kauf- und Reaktionsverhalten von Kunden auf eine werbliche Ansprache werden die Kunden selektiert, die die höchste Affinität zur Kampagne aufweisen. Bei vorher festgelegter Größe der benötigten Kundengruppe kann so ein optimales Ergebnis erzielt werden.

Die genannten und viele weitere BI-Anwendungen erlangen im Laufe ihres Reifungsprozesses nahezu den Status operativer Anwendungen. Umso wichtiger ist es, diese Prozesse entsprechend zu automatisieren und in einem geordneten Regelbetrieb abzubilden. Besonderes Augenmerk ist dabei auf den Übergang von der Entwicklung (wie analytisches CRM) in den operativen Einsatz (wie operatives CRM) zu legen, da gerade das Operationalisieren komplexer analytischer Algorithmen aufwändig werden kann.

ETL- und ELT-Anbindung eines „SAS Enterprise Miner“-Modells

Obwohl die Verzahnung der operativen System- und BI-Landschaft in den letzten Jahren zugenommen hat, stehen zwischen beiden Welten sehr häufig, meist aus Sicherheits- und Stabilitätsgründen, hohe Mauern. In der täglichen Praxis bedeutet dies, dass die BI-Abteilung zwar eine agile und schnelle Analyse von geschäftskritischen Prozessen zu leisten vermag, die entsprechend zeitnahe Umsetzung im operativen System dann jedoch an der Ressourcen-Knappheit der IT-Abteilung scheitert. Dies liegt vor allem daran, dass die zur Analyse verwendeten Tools nicht gleichzeitig für die operative Umsetzung verwendet werden oder werden können. Wie diese Hürde dennoch überwunden werden kann, soll nun am Beispiel des SAS Enterprise Miner (EM) gezeigt werden.

Mit diesem oft genutzten und mächtigen Analyse-Tool können in einer entsprechenden Umgebung Analysen und Modellentwicklungen auf Basis eines Analyse-Datensatzes leicht aufgebaut werden (siehe Abbildung 4). Die Architektur der operativen Umgebung – für jeden Kundendatensatz muss regelmäßig die Modellberechnung durchgeführt werden – ist alles andere als trivial. Eine dedizierte Instanz des EM als Teil der operativen System-Landschaft (mit entsprechenden Release-Zyklen) ist aus Architektursicht denkbar. Die Anbindung erfolgt klassisch als ETL-Prozess.

Allein aus Kostengründen fällt diese Variante jedoch häufig aus. Neben den reinen Lizenzkosten sind auch Prozesskosten zu betrachten. So muss der IT-Betrieb die Wartung und den Betrieb zur Verfügung stellen und das durchaus komplexe Deployment der BI-Modelle in der Produktivumgebung durchführen. Letzteres erfordert spezielles SAS-EM-Wissen, das selten zum Portfolio eines IT-Betriebs gehört. Nun ist ein komplexer Deployment-Prozess in ein operatives System schlichtweg inkompatibel mit der agilen Arbeitsweise einer BI-Analytik-Abteilung. Andere Lösungen müssen gefunden werden.

Dafür wird die Fähigkeit des EM genutzt, die entwickelten Modelle entweder als Java- oder als C-Code (zusätzlich entsteht ein XML-File, das die Metadatenbeschreibung enthält) zu exportieren. Java- und C-Code sind für die IT leicht zu verarbeiten. Der vom SAS-EM generierte Code kann in ein C-/Java-/Embedded-SQL-Framework mit einer Oracle-Datenbank als Datenquelle eingefügt werden. Für die IT ist es nun ein Leichtes, einen schnellen und schlanken Deployment-Prozess zu etablieren, sodass der Agilität der BI-Analytik Rechnung getragen werden kann. Die Verwendung von Programmen mit Embedded SQL führt zu ETL-Prozessen.

Vom ETL zum ELT: Lösungsvariante „external function“

Falls die Berechnung der im BI-Modell enthaltenen Logik vor allem aus Zeilentransformationen besteht, lässt sich der vom SAS-EM generierte Code auch gut als „external function“ innerhalb der Oracle-Datenbank zur Verfügung

stellen. Da das Daten-Management dann vollständig innerhalb von Oracle stattfindet, sind alle gängigen Performance-Maßnahmen (wie Parallelisierung) innerhalb der Datenbank anwendbar. Darüber hinaus lässt sich die Automatisierung der Codeerstellung sogar so weiterentwickeln, dass etwa über ein in der Oracle-Datenbank (PL/SQL) hinterlegtes Framework die XML-Metadaten geladen und ausgewertet werden.

Es ist kein Problem für die BI-Abteilung, in der ihr gut bekannten PL/SQL-Umgebung einen Generator zu implementieren, der die notwendigen Header-/Wrapper-Funktionen in PL/SQL und gegebenenfalls C beziehungsweise Java erzeugt. Dieser Code kann in einer Test- oder Integrations-Umgebung auf syntaktische und semantische Fehler hin durch die BI-Abteilung selbst geprüft werden. Der durch die IT durchzuführende Deployment-Vorgang in der Produktionsumgebung kann auf ein absolut notwendiges Minimum an Tätigkeiten – hier den Austausch einer Library auf Betriebssystemebene – reduziert werden. Damit ist die Agilität auch in der Produktionsumgebung angekommen. Durch Verwendung der „external function" innerhalb der DB ist nun der Wechsel vom ETL- zum ELT-Prozess vollzogen. Während ETL-Prozesse alle Basisdaten extrahieren und dann das Ergebnis für den gesamten Datenbestand zurückliefern müssen, kann etwa durch selektive Auswertung von Views, die die Logik mithilfe der „external function" abbilden, die notwendige Materialisierung auf ein Minimum – beispielsweise pro Tag oder einen Teil der Kundengruppe – reduziert werden. Nur der Teil, der tatsächlich benötigt wird, muss materialisiert werden.

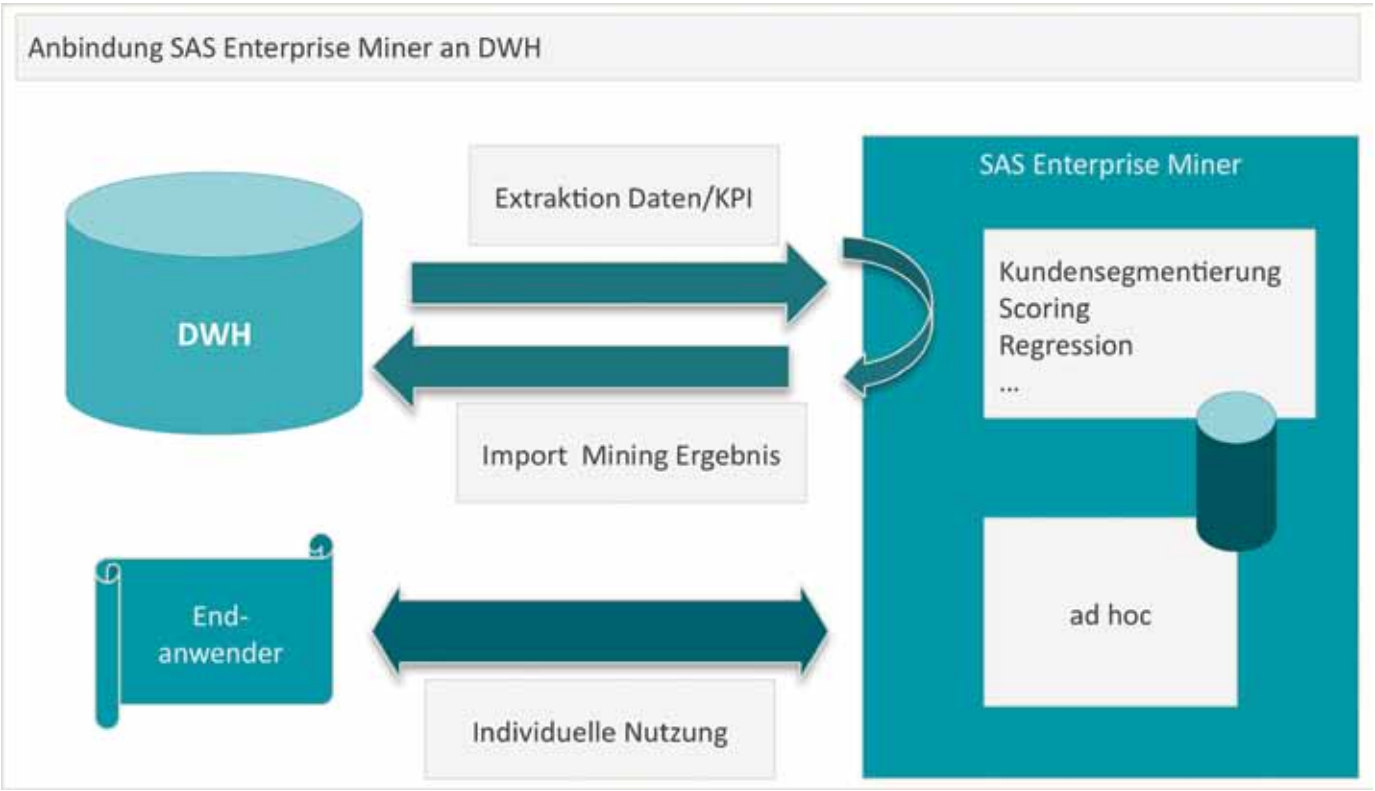

Abbildung 4: Beispiel ETL mit SAS-EM

Fazit

Bei der Erstellung der BI-Architektur von DWH-Landschaften wird oft ein erbitterter Streit zwischen ETL- und ELT-Anhängern geführt. Dies bezieht sich aber meist auf den Schritt der Beladung des DWH (Source-Abstraction-Layer), gegebenenfalls noch auf das der Mappings in den Integration-Layer. Bei der Erstellung des Presentation-Layers wird dem Daten-Management weniger Aufmerksamkeit geschenkt. Da funktionale Anforderungen im Vordergrund stehen, erfolgt die Implementierung dieses Prozesses vielmehr so nebenbei während der Umsetzung. Nicht selten sind Performance-Probleme die Folge, die mit viel Aufwand (oft auch Hardware) wieder beseitigt werden müssen. Eine mögliche Lösung des Problems besteht darin, eine Umstellung der Applikation von einem ETL-Prozess zu einem ELT-Prozess durchzuführen. Aus dem Blickwinkel des Datenmanagements in der DWH-Landschaft eröffnen sich hier oft ungeahnte Möglichkeiten.

Dr. Gernot Schreib
gernot.schreib@btelligent.com

Wie oft wünscht man sich, bestimmte wiederkehrende Routineaufgaben nicht mehr von Hand oder am liebsten gar nicht mehr selbst erledigen zu müssen? Warum sollte dies bei der Entwicklung von ETL-Prozessen in einem Data Warehouse anders sein?

Automatische Generierung der ETL-Prozesse: die Möglichkeiten von OWB und ODI

Irina Gotlibovych, MT AG

Anhand eines von der MT AG entwickelten Frameworks werden in diesem Artikel Möglichkeiten der automatischen Generierung von ETL-Prozessen im Oracle Warehouse Builder einerseits und im Oracle Data Integrator andererseits gegenübergestellt und verglichen.

Bei der Entwicklung der ETL-Prozesse in einem Data Warehouse sieht man sich wiederholt vor die Aufgabe gestellt, Prozess-Schritte aufbauen zu müssen, die einer gleichartigen Logik folgen. So werden in jedem Projekt viele Daten-Objekte auf die gleiche Weise aus Quellsystemen in den Arbeitsbereich geladen. Beim Transformationsschritt werden Daten in das einheitliche Format der Ziel-Datenbank überführt; gängige Verfahren dabei sind beispielsweise Datentyp-Konvertierung und Daten-Bereinigung. Anschließend werden Daten nach dem gleichen Prinzip – etwa mit Delta Load oder SCD – in das Data Warehouse eingebracht.

Wenn man mit dem OracleWarehouse Builder arbeitet, bedeutet dies in der Praxis oft, logisch identische Mappings in manueller Kleinarbeit anlegen zu müssen. In jedem dieser Mappings sind von Hand Operatoren anzulegen und zu verbinden. Für jedes Attribut eines Expression Operators muss der Ausdruck eingetragen werden. Eigen-

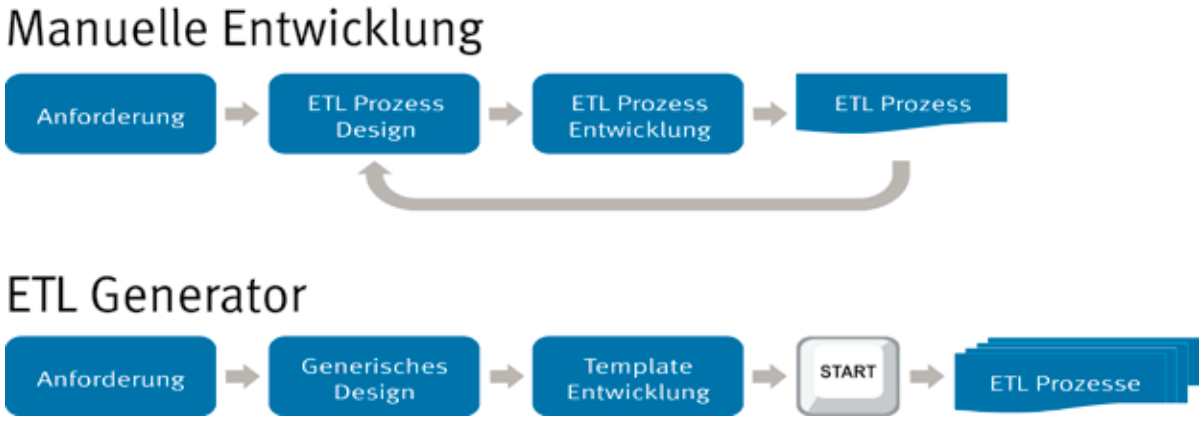

Abbildung 1: Prozesskette bei der Erstellung von Prozessen mit dem ETL-Generator im Vergleich zur manuellen Entwicklung

schaften von Operatoren und Attributen sind immer wieder neu zu setzen. Kommt in einer Quell-Tabelle später eine neue Spalte hinzu, muss sie in den meisten Fällen identisch zu den anderen Spalten geladen und verarbeitet werden. Um das zu erreichen, ist aber im entsprechenden Mapping jeder betroffene Operator manuell zu ändern. Besonders aufwändig wird es, wenn sich die grundlegende Logik ändert; im Falle von späteren Änderungsanforderungen ist zumeist jedes Mapping wieder anzupassen. Obwohl sie der gleichen Logik folgen, muss trotzdem jedes dieser Mappings einzeln getestet werden: Da sie unabhängig voneinander entwickelt wurden, können in jedem auch unterschiedliche Fehler auftreten. Bei der manuellen Entwicklung spielt der menschliche Faktor eine enorme Rolle. Repetitive Entwicklungsarbeiten wie das fünfzigfache Anfertigen eines Delta-Load-Mappings sind monoton und führen dadurch zu Fehlern.

Ganz ähnlich verhält es sich bei der Entwicklung mit dem Oracle Data Integrator. Interfaces müssen für jede Ziel-Tabelle einzeln angelegt werden. Trotz der gleichen Logik sind in jedem Interface Zuordnungen von Attributen unabhängig voneinander zu definieren und bei späteren Anforderungen auch einzeln zu ändern. Der Oracle Data Integrator beinhaltet durch die mitgelieferten Knowledge- Module bereits die Möglichkeit, generische Funktionalitäten zu nutzen beziehungsweise aufzubauen. Ähnlich wie Makros oder Templates beinhalten diese Knowledge-Module wiederverwendbare Logiken und nehmen dadurch dem Entwickler einen Teil der manuellen Arbeit ab. Sie können in mehrere Interfaces eingebunden werden, ersetzen aber nicht die manuelle Anlage jedes einzelnen.

Generische ETL-Entwicklung mit OWB und ODI

Die Frage stellt sich: „Warum entsteht so ein Mehraufwand bei der manuellen Entwicklung und wie kann man diesen vermeiden?“ Wäre es nicht schöner, die Logik nur einmal zu entwickeln und diese dann in weiteren Prozessen beziehungsweise Projekten mehrmals zu verwenden? Das Problem bei der Entwicklung sowohl im OWB als auch im ODI ist, dass es keine Möglichkeit gibt, Mappings beziehungsweise Interfaces ohne Bindung an konkrete Objekte (Tabellen, Spalten etc.) anzulegen. Die fachliche Logik eines Prozesses ist immer fest mit den Umgebungs-Informationen verbunden. Um die gleiche Logik nicht mehrfach neu erzeugen zu müssen, bräuchte man einen Weg, Prozesse generisch, also ohne Bezug zu den eigentlichen Objekten definieren zu können. Die Erzeugung der Mappings beziehungsweise Interfaces kann dann automatisch erfolgen, wobei Objekt-Namen als Parameter dem Generierungsprozess mitgegeben werden.

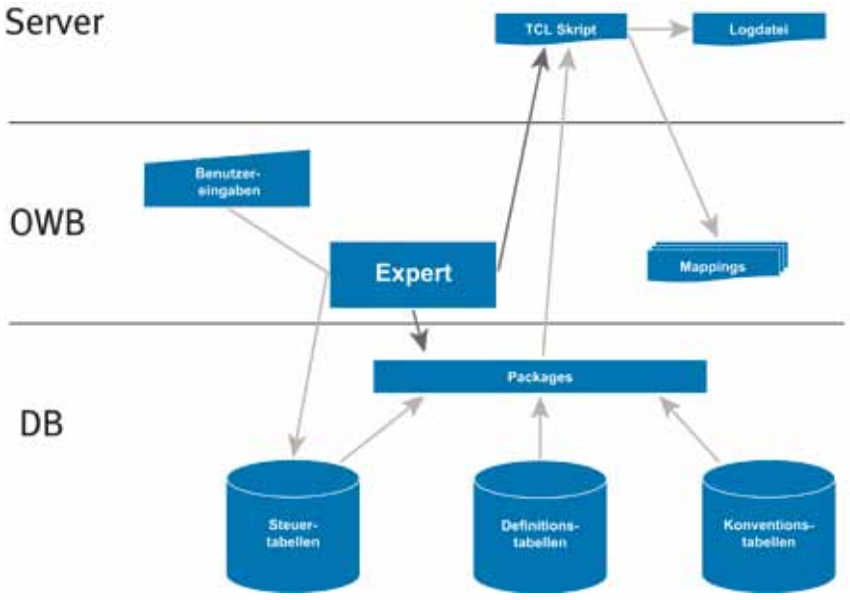

Abbildung 2: Architektur des OWB Mapping Generators

Dieser Ansatz wurde bei der Entwicklung des ETL-Generators zugrunde gelegt (siehe Abbildung 1).

Welche Vorteile bringt so ein generischer Ansatz? Angenommen, man möchte Slowly Changing Dimensions Typ 2 in seinem Data Warehouse implementieren. Bei der manuellen Entwicklung würde man für jede Ziel-Tabelle die komplexen Join- und Splitter-Bedingungen in Abhängigkeit von den jeweiligen Primärschlüsseln und Spalten einzeln implementieren. Wählt man den generischen Ansatz, wird die Logik unabhängig von Tabellen und Spalten einmal in Form eines Templates definiert. Der Generierungsprozess arbeitet nun nach allgemeinen Regeln, was identischen Code und gleiche Qualität für alle Prozesse garantiert. Und wenn später eine neue Spalte hinzukommt? Da das Template allgemein definiert wurde und die konkreten Primärschlüssel beziehungsweise Spalten erst bei der Verarbeitung aus der Datenbank ausgelesen werden, wird die neue Spalte bei einer erneuten Generierung des Mappings beziehungsweise Interface automatisch berücksichtigt. Es ist keine manuelle Anpassung des Prozesses im OWB beziehungsweise ODI notwendig. Doch wie sieht es mit Fehlerbehebung und Testen aus? Da man die Logik an einer Stelle entwickelt, müssen die Fehler auch nur an einer Stelle behoben werden – nämlich in dem zugrunde liegenden Template und nicht in jedem Prozess einzeln. Ist man einmal sicher, dass die Logik in dem Template richtig definiert ist, kann man auch sicher sein, dass jedes damit generierte Mapping beziehungsweise Interface korrekt laufen wird.

ETL-Generator

Das besagte Framework besteht aus zwei Komponenten – OWB Mapping Generator und ODI Interface Generator – und ermöglicht es, Prozesse auf Basis von mitgelieferten oder selbst entwickelten Templates automatisch zu generieren. Im OWB Mapping Generator werden Mappings mithilfe von OMB Plus beziehungsweise TCL erzeugt. ODI Interface Generator verwendet das bei ODI zur Verfügung stehende Java-API (siehe Tabelle 1). Im Gegensatz

Objekt	OWB	ODI
Prozess	LOOP insert_line (OMBCREATE MAPPING <name>); END LOOP;	LOOP odiInterface = new OdiInterface (…,<name>,…); END LOOP;
Operator	LOOP insert_line (OMBALTER MAPPING <name> ADD <operator type> OPERATOR <operator name>); END LOOP;	LOOP Source table sourceDatastore = …findByName (<name>,<model>); interfaceHelper.setSourceDataS- tore (sourceDatastore); Filter interfaceHelper.setSourceFilter (<filter condition>); END LOOP;

Tabelle 1: Automatische Generierung: OWB vs. ODI

Objekt	Eigenschaft	Wert
OWB Table Operator	Operator type	TABLE
OWB Set Operator	SET_OPERATION	MINUS
OWB Expression Operator	EXPRESSION	INGRP1.$attr_name
ODI Source Datastore	Model	STAGE
ODI DataSet	Operator	UNION
ODI Target Column	Mapping	SYSDATE
ODI Filter	Filter condition	$func_get_my_filter_cond

Tabelle 2: Definition von Templates

Schema	SOURCE	STAGE	CORE
Tabelle	SRC_PRODUCT	STG_PRODUCT	PRODUCT

Tabelle 3: Tabellenstamm „PRODUCT“ in den Schemata „Source“, „Stage“ und „Core“

zur manuellen Entwicklung setzt man beim Gebrauch des Frameworks nicht mehr jedes Mapping im OWB Design Center beziehungsweise jedes Interface im ODI Designer einzeln um, sondern (siehe Abbildung 1) definiert ein allgemeingültiges Template für eine Klasse von Prozessen. Anschließend generiert man die dazugehörigen Prozesse unter Einbeziehung seiner Projektvorgaben automatisch. An dieser Stelle ist anzumerken, dass es sich bei solch einem Template nicht um ein programmiertes Skript zur Generierung der Prozesse handelt, sondern genauso wie im OWB und ODI um eine deklarative Definition auf Basis von Metadaten.

Im OWB Mapping Generator wird die Generierung der Mappings über einen „OWB Expert“ aus der Oracle Warehouse Builder GUI angestoßen. Dieser leitet dialoggestützt durch die einzelnen Schritte. Nachdem man seine Auswahl bezüglich des zu generierenden Templates und der zu verwendenden Objekte getroffen hat, legt der OWB Mapping Generator die Eingaben in den Steuer-Tabellen auf der Datenbank ab. Danach werden PL/SQL-Prozesse gestartet (siehe Abbildung 2), die aus den Definitions-Tabellen das gewünschte Template auslesen und die darin gespeicherte Logik mit den Umgebungs-Informationen aus den Konventions-Tabellen anreichern. Damit wird nun ein TCL-(OMB-Plus)-Skript generiert, mit dem die Mappings anschließend automatisch erzeugt werden. Das Ergebnis ist sofort im Oracle Warehouse Builder sichtbar und kann weiterverwendet beziehungsweise eingesetzt werden. Während der Generierung der Mappings ist jeder Schritt in einer Logdatei auf dem Server protokolliert. Man hat so stets den vollen Überblick über die generierten Objekte im Data Warehouse.

Die Architektur des ODI Interface Generators ist der des OWB Mapping Generators sehr ähnlich (siehe Abbildung 3). Die Generierung der Prozesse funktioniert allerdings auf eine andere Weise. Statt der dynamischen PL/SQL-Prozesse kommt hier das statische ODI-Java-API zum Einsatz. Es stellt vordefinierte Methoden zur Verfügung, mit denen ODI-Interfaces direkt erzeugt werden können. Im Gegensatz dazu wird im OWB Mapping Generator das ausführbare Skript erst zur Laufzeit erstellt. Beim Ausführen eines OMB-Plus-Skripts wird das Logfile immer automatisch auf dem Server generiert. Im ODI Interface Generator ist das Logging ein Teil der implementierten Java-Schnittstelle.

Einer für alle

Das Kernstück der Architektur des ETL-Generators bilden die bereits mehrfach erwähnten generischen Templates. Der ETL-Generator stellt in der Datenbank einen Satz von Definitions-Tabellen bereit, in denen die Templates mithilfe der Metadaten beschrieben sind. In den Definitions-Tabellen findet man keine Tabellen- oder Attribut-Namen, die konkreten Objekte werden erst während der Generierung automatisch an die Templates gebunden. Um den Einstieg in das Framework zu erleichtern und die Anlage der Templates zu vereinfachen, besitzen OWB Mapping Generator und ODI Interface Generator jeweils eigene Definitions-Tabellen. Die Begrifflichkeiten beim OWB Mapping Generator sind die gleichen wie im Oracle Warehouse Builder und beim ODI Interface Generator wie im Oracle Data Integrator – man findet sich demnach schnell zurecht.

Im Datenmodell des OWB Mapping Generators sind Informationen über Operatoren, Attribute, Properties und Connections enthalten. Das Datenmodell des ODI Interface Generators beinhaltet unter anderem Data-Sets, Operatoren (wie Tabellen, Joiner oder Filter), Properties (wie Mappings oder Knowledge-Module) und Optionen. Bei Namen für Tabellen-Operatoren, die sich von Prozess zu Prozess unterscheiden

und von der zugehörigen Tabelle abhängen, werden Platzhalter verwendet. Die Property-Tabellen können je nach Anforderung oder Komplexität der umzusetzenden Logik sowohl statische als auch dynamische Werte enthalten (siehe Tabelle 2). Eine Eigenschaft lässt sich mithilfe vordefinierter dynamischer Parameter festlegen. Somit beschreibt man beispielsweise alle Attribute eines Operators zusammen, und nicht jedes Attribut einzeln. Erfordert die fachliche Logik eine umfassendere Berechnung der Werte, etwa abhängig vom Primärschlüssel der Tabelle oder von Datentypen der Spalten, hat man im ETL-Generator die Möglichkeit, eine benutzerdefinierte Funktion anzulegen, die dann in der Property-Tabelle verwendet werden kann.

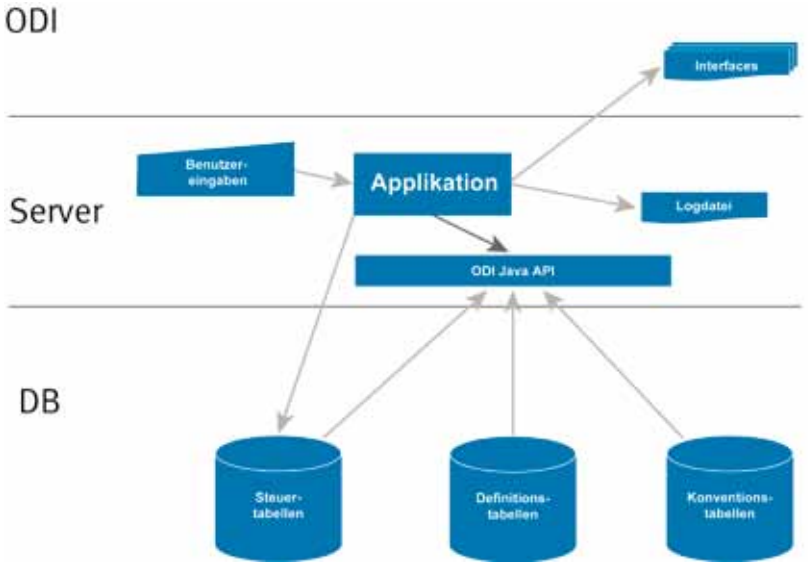

Abbildung 3: Architektur des ODI Interface Generators

Ohne Namenskonventionen läuft nichts

Da die Definition der Templates generisch erfolgt, braucht man nun einen Weg, diese mit den erforderlichen Umgebungsobjekten (Schemata, Tabellen etc.) zu verbinden. Um die Generierung einzelner Prozesse entsprechend den jeweiligen Anforderungen zu ermöglichen, sollte man im ETL-Generator Namenskonventionen und Umgebungsinformationen ablegen. Dabei spielt der Begriff „Tabellenstamm" (table radical) eine zentrale Rolle. Damit ist der gemeinsame Teil der Tabellen-Namen (siehe Tabelle 3) über alle im ETL-Prozess verwendeten Schemata hinweg gemeint.

Der Tabellenstamm wird bei der Generierung von ETL-Prozessen verwendet, um zusammengehörende Objekte in einem Prozess zu verbinden. Die Funktionsweise des Frameworks basiert auf der Annahme, dass alle verwendeten Datenbank-Objekte einer allgemeinen Namenskonvention folgen. In den bereitgestellten Konventions-Tabellen beschreibt man mithilfe der regulären Ausdrücke die Namenskonventionen der Datenbank-Objekte innerhalb der OWB-Module beziehungsweise ODI-Module und legt die Namenskonvention für die zu erzeugenden Mappings beziehungsweise Interfaces fest. Durch die einfache Erweiterbarkeit und Individualisierung des Frameworks können im ETL-Generator beliebige Namenskonventionen abgebildet werden.

Fazit

Der ETL-Generator ist ein kleines Framework, mit dem man die Entwicklung in OWB- und ODI-Projekten enorm beschleunigen kann. Wie in Abbildung 4 leicht zu erkennen ist, lohnt sich der Einsatz des ETL-Generators bei steigender Anzahl der Prozesse. Man profitiert dabei in allen Projektphasen:

- Die Entwicklung wird beschleunigt, da statt einer großen Menge von Prozessen nur noch ein Template entwickelt werden muss
- Vereinheitlichung und damit auch Qualitätsverbesserung des Codes ist ein unstrittiger Gewinn, den man in großen Projekten kaum noch erreichen kann
- Der Testaufwand wird dank des generischen Ansatzes ebenfalls deutlich reduziert
- Auf Neuanforderungen kann schnell reagiert und ein Data Warehouse in kurzer Zeit neu aufgebaut werden
- Die Wartungsaufwände werden bei der Verwendung des ETL-Generators deutlich minimiert, wodurch man mehr Zeit für konzeptionelle Aufgaben und neue Projekte gewinnt

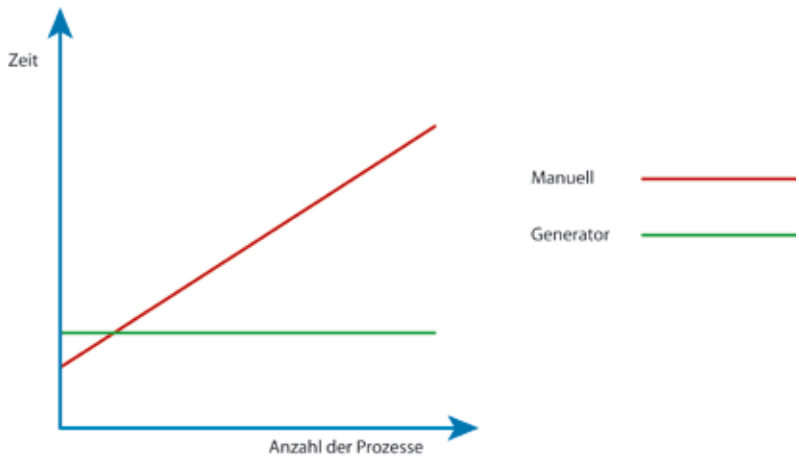

Abbildung 4: Zeitgewinn innerhalb aller Projekt-Phasen beim Einsatz des ETL-Generators

Irina Gotlibovych
irina.gotlibovych@mt-ag.com

Effizienz und Laufzeiten von ETL-Prozessen werden immer wieder heftig diskutiert. Die Auswirkungen von ungeschickt entworfenen Ladestrecken zeigen sich schließlich im gesamten System und sind nicht zuletzt teuer. Mehr und mehr setzt sich die Erkenntnis durch, dass kritische ETL-Prozesse am besten in der Data-Warehouse-Datenbank selbst durchgeführt werden sollten. Deshalb hier einige Hinweise zur Planung und Umsetzung von ETL-Prozessen in der Oracle-Datenbank.

ETL-Prozesse in der Oracle-Datenbank

Alfred Schlaucher, ORACLE Deutschland B.V. & Co. KG

Ein Data Warehouse ist im Gegensatz zu den meisten OLTP-Anwendungen ein datengetriebenes System. Es geht um die Organisation von statischen Informationen sowie die Modellierung von Informations-Zusammenhängen und nicht um die Modellierung von Prozessen, bei denen Datenflüsse zu regeln sind. „Funktionen zu den Daten bringen" kann daher ein wegweisendes Paradigma sein.

Die spezifischen Eigenschaften des Data Warehouse nicht unterschätzen

Ein Data Warehouse ist kein OLTP-System. Bei der Planung von ETL-Prozessen in der Datenbank ist diese Unterscheidung besonders wichtig. Tabelle 1 zeigt zusammengefasst die wichtigsten Unterschiede, die für die Planung von ETL-Prozessen bedeutsam sind.

Diese Liste lässt sich weiter fortsetzen. Doch schon jetzt kann man sagen, dass die Mehrzahl aller Änderungsoperationen in einem Data Warehouse durch „INSERTS" durchgeführt wird, die zusammenhängende Bereiche in den Datafiles beschreiben. Die erste Betrachtung gilt also der „INSERT"-Operation.

Logging/Nologging

Zunächst ist es eine bewusste Entscheidung, in einem Data Warehouse mit „Nologging" zu arbeiten. Das bedeutet, die DWH-Datenbank wird im „NOARCHIVELOG"-Modus betrieben, was zur Folge hat, dass alle Log-auslesenden Programme nicht mehr funktionieren. Entsprechende alternative Konzepte sind zu überlegen:

- *Data Guard*
  Ein DWH muss nicht repliziert werden, es ist bereits ein logisches Replikat
- *Flashback-Feature*
  Die Funktion hört sich gut an, die Wirkung ist jedoch durch einen geschickten ETL-Prozess einfacher und Ressourcen-günstiger zu haben
- *RMAN*
  Da meist nur kontrollierte Änderungen stattfinden, kann man ein Sicherungskonzept auch ohne RMAN realisieren

Bleibt der „ARCHIVELOG"-Modus dennoch aktiv, weil man beispielsweise Änderungen an den vielen kleineren DWH-Tabellen doch mit RMAN sichern will, so ist der „NOLOGGING"-Hint (/*+ NOLOGGING */) eine wichtige Einstellung in den „ETL-INSERT"- Statements. Ansonsten stört der „ARCHIVLOG"-Modus wenig, weil abfragende Benutzer keine Änderungen in den Daten hinterlassen – nur der ETL-Prozess ist betroffen.

Insert, CTAS, Direct-Path-Load

Auf dem Weg der Daten in die Datenbank-Blöcke unternimmt Oracle einige Schritte, die bei einfachen „INSERT"-Operationen in einem OLTP-System sinnvoll sind, aber bei einem schnellen Laden im Data Warehouse stören, wie das Space-Management und das Zwischenpuffern in der SGA. Diese Aktivitäten lassen sich durch den Direct-Path-Load-Hint (/*+ APPEND */) umgehen. Darüber hinaus ist die schnellste Schreib-Operation das Schreiben in eine neue und damit leere Tabelle (CREATE

OLTP	DWH
Viele einzelne, nicht kontrollierbare „INSERT"-, „UPDATE"- und „DELETE"-Operationen	Alle Änderungsvorgänge sind in der Regel durch den ETL-Prozess kontrolliert. „Man weiß, was man tut", und man kann gezielt Einfluss nehmen.
„UPDATE"- im Vergleich zu „INSERT"-Operationen relativ häufig	Bei sinnvollem Konzept kaum „UPDATE"-Operationen
Ressourcen-Auslastung relativ homogen verteilt	Ressourcen-Auslastung gliedert sich in Online- und ETL-Phasen, Datenbestände werden unterschiedlich intensiv bearbeitet
Permanent unterschiedliche Blöcke von Änderungen und Lese-Operationen betroffen	Eher zusammenhängende Blockbereiche von Lese- und Änderungs-Operationen betroffen
Die Anzahl großer und kleiner Tabellen lässt sich nicht eindeutig in Gruppen einteilen. Daher müssen gleiche Verarbeitungsregeln für alle Tabellen aufgestellt werden	Tabellen lassen sich gruppieren in eine Gruppe mit vielen kleinen Tabellen und eine Gruppe mit wenigen großen Tabellen. Davon kann man gesonderte Verarbeitungsregeln für die unterschiedlichen Gruppen ableiten.
Backup der gesamten Datenbank	Nur einzelne Bereiche von Backup betroffen

Tabelle 1

	Statement	Zeit in Min:Sek	Redo Size
1	Create Table T20 as select * from T10;	01:00	keine
2	Insert in leere Tabelle	01:46	1519380548
3	Insert in gefüllte Tabelle (Wiederholung)	01:58	1518890292
4	Insert in gefüllte Tabelle (Wiederholung)	01:59	1518890295
5	insert /*+ APPEND */ into t20 select * from t10;	01:00	777596
6	insert /*+ NOLOGGING */ into t20 select * from t10;	01:48	1519391944
7	insert /*+ APPEND NOLOGGING */ into t20 select * from t10;	01:03	777504
8	Einzel-INSERTs in Schleife einer PL/SQL-Prozedur	08:31	Nicht messbar
9	insert into t20 select * from t10 (ARCHIVLOGMODUS)	02:56	1518884743
10	(Setting „parallel_degree_policy=AUTO") insert /*+ APPEND */ into t20 select * from t10;	00:23	779124
11	(Setting „parallel_degree_policy=AUTO") Create Table T20 as select * from T10;	00:20	Nicht messbar
12	(Tabelle T20 auf SSD-Platte) insert /*+ APPEND */ into t20 select * from t10;	00:10	776998
13	(Anlegen Tabelle T20 auf SSD-Platte) Create Table T20 as select * from T10;	00:10	777476
14	(Mehrmaliges Schreiben, ohne zuvor die Buffer Caches zu leeren) insert /*+ APPEND */ into t20 select * from t10;	00:29	777596

Tabelle 2

TABLE name AS SELECT feld1, feld 2.... FROM quelle), auch „CTAS" genannt. In einer Umgebung mit „ARCHIVELOG" und bei bereits bestehenden Tabellen sollte ein „INSERT" immer mit den Hints „/*+ NOLOGGING APPEND */" gesteuert werden.

Folgende kleine Testreihe verdeutlicht anschaulich die Unterschiede (siehe Tabelle 2). Man beachte dabei auch die Verringerung der „Redo Size", mit der zusätzlich Performance-Effekte erzielbar sind. Der Test eignet sich lediglich, um grundsätzliche Trends zu erkennen, da es sich nur um eine kleine Testmaschine mit unzureichendem Storage-System handelt. Geschrieben werden 10 Millionen Sätze (ca. 1,7 GB), bei denen alle Felder zu 100 Prozent gefüllt sind. Gelesen wird aus einer Tabelle „T10" und geschrieben in eine Tabelle „T20".

Die Direct-Path-Load-Variante (APPEND, Fälle 5 und 7) ist die schnellste. Auch hinter „Create Table as select" (CTAS, Fall 1) steckt ein Direct-Path-Load. Zu erkennen ist auch dessen geringe Redo Size. Das Schreiben in eine bereits gefüllte Tabelle (Fälle 3 und 4) ist immer etwas langsamer als das Schreiben in eine komplett leere Tabelle (Fall 2).

„NOLOGGING (Fall 6) ist wesentlich schneller als das Schreiben in einer „ARCHIVELOG-MODUS"-Datenbank (Fall 9). Der Fall 8 sollte noch erwähnt werden: Er simuliert das Schreiben einzelner Sätze, also das Gegenteil einer mengenbasierten Verarbeitung mit reinem SQL.

Auch heute noch wird diese Variante in einzelnen Data-Warehouse-Systemen praktiziert. Hier wird innerhalb einer PL/SQL-Schleife ein zu schreibender Satz zunächst über Variablen zusammengesetzt und dann über einen einzelnen „INSERT" geschrieben. Dahinter steckt der Glaube, dass man zum Beispiel komplexe Prüfungen nur in dieser Weise lösen kann (Zur Lösung dieser Anforderung weiter unten mehr). Dieser Fall entspricht auch der Vorgehensweise einiger ETL-Tools, wenn diese nicht in einem Bulk-Modus arbeiten.

Parallelisierung

Das Parallelisierungs-Feature der Datenbank kann den Schreibvorgang erheblich beschleunigen. Dies setzt allerdings voraus, dass das Hardware-System genügend Platten und CPUs hat. Im Verlauf des ETL-Prozesses kann man die Parallelisierung zusätzlich gezielt und manuell steuern sowie einen höheren Parallelisierungsgrad einstellen, wenn die ETL-Jobs allein auf der Maschine laufen, also nicht gleichzeitig im Online-Betrieb gelesen wird.

Ab dem Datenbank-Release 11.2.0.2 sollte man neben der manuellen Steuerung unbedingt den Parameter „parallel_degree_policy" mit dem Setting „AUTO" testen, da dieser Parameter auch den jeweiligen ETL-Jobs die optimale Parallelisierung in Abhängigkeit von Hardware-Ressourcen und Auslastung zuweist. Mit diesem Parameter wurden in dem Test oben auch beim Direct-Path-Load die Ladezeiten noch einmal optimiert (Fälle 10 und 11).

SSD und andere schnelle Datenträger

Weiter unten im Artikel wird die Verwendung temporärer Tabellen vorgestellt. In der Praxis stellen diese ein sehr wichtiges Hilfsmittel bei der Strukturierung des Gesamtprozesses dar. Einige Informationen können im Verlauf des Ladeprozesses mehrfach gelesen und geschrieben werden, bis sie an ihren endgültigen Bestimmungsort gelangen. Wenn solche Tabellen nicht permanent im Hauptspeicher gehalten werden können, kann man dennoch versuchen, sie auf schnelle Datenträger wie SSDs oder Flashspeicher zu legen, auch wenn die Masse der Data-Warehouse-Daten noch auf klassischen Spindel-Datenträgern liegt. Im Testfall oben hat die Verwendung von nur einer SSD-Platte zu einer Reduzierung der Ladezeit auf 10 Sekunden geführt (Fälle 12 und 13).

DWH-Prüfungen und Umgang mit Constraints in der Datenbank

Alle notwendigen Plausibilitätsprüfungen sollten im Integration-Layer (Stage) des Data Warehouse erfolgen. Diese Prüfungen gehören zu den aufwändigsten Schritten in jedem ETL-Prozess. Datenbank-Constraints bieten sich für einige Prüfungen an. Doch diese verlangsamen ETL-Läufe erheblich. Anders als in OLTP-Systemen laufen Änderungsvorgänge im Data Warehouse kontrolliert ab. Im Rahmen des ETL-Prozesses kennt man jede Änderung. Daraus folgt, dass Constraints, die gegen unkontrolliertes Ändern schützen, nicht notwendig sind.

Die Konsistenz der neu zu ladenden Daten sollte dagegen mengenbasiert mit SQL in der Datenbank gesichert werden. Für die Planung solcher Prüfungen muss man die Art der Prüfung feststellen. Tabelle 3 zeigt die Kategorisierung aller durchzuführenden Prüf-Aktivitäten.

Für jede dieser Kategorien lassen sich standardisierte Wege mit SQL entwickeln. Anmerkung: Im Seminar „ETL in der Datenbank" des Autors werden Lösungen für die jeweiligen Kategorien vorgestellt (siehe Oracle-DWH-Community-Webseite www.oracledwh.de). Alles, was es in einem Data Warehouse zu prüfen gibt, lässt sich ohne Constraints und ohne prozedurale Programmierung mit schnellen, mengenorientierten „INSERT"-Operationen lösen.

Hier gibt es nur den einen Ausnahmefall: die Komplexität. Wird ein SQL-Statement zu komplex, kann man es in prozedurale PL/SQL-Logik umwandeln. Hierzu bieten sich dann parallelisierbare Table-Functions an, die hier nicht weiter betrachtet werden (siehe auch hierzu das Seminar „ETL in der Datenbank").

Kategorie	Beschreibung
Attribut-/Column-bezogene Regeln	Feldformate, Not Null, Maskenformate, Wertebereiche, Ober-, Untergrenzen, Listen
Satzbezogene Regeln	Abhängigkeiten zwischen Werten von Columns desselben Satzes, funktionale Abhängigkeiten
Satzübergreifende Regeln	Eindeutigkeit einer Tabelle (Primary Key), Intervalle, Satzgruppen, Gruppenwechsel, rekursive Strukturen, Aggregatprüfungen
Tabellenübergreifende Regeln	Child-/Parent (Orphan) / Parent-Child (Childless), Kardinalitäten, Aggregatbildungen
Zeit-/Zusammenhang-bezogene Regeln	Zeitintervalle, geographische, politische, soziale, organisatorische Fakten
Verteilungs-/Mengenbezogene Regeln	Durchschnittsbildung, Varianzen, Verteilungen

Tabelle 3

Das Arbeiten mit Zwischen-Tabellen

Um der potenziellen Komplexität von prüfenden SQL-Statements zu begegnen, sollte man temporäre Tabellen nutzen. Hier kann man Zwischen-Ergebnisse von Prüfungen ablegen. Die Tests oben haben gezeigt, dass das Anlegen und Beschreiben von neuen beziehungsweise leeren Tabellen zu den schnellen Vorgängen gehört. Um die Zeit der Einzelsatzverarbeitung in Fall 8 zu erreichen, kann man bequem die gleiche Datenmenge schon in bis zu acht temporären Zwischen-Tabellen speichern. In der Regel reichen schon eine bis drei Zwischen-Tabellen.

Aktivitäten bündeln und die Caching-Funktion der Datenbank nutzen

Ein großer Teil des Aufwands bei den oben gezeigten Lade-Operationen stellt der „SELECT"-Teil der Befehle dar. Über den Befehl „SELECT COUNT(*) FROM T10" (Full Table Scan) misst man 27,47 Sekunden, die in allen Testfällen gleichermaßen stecken. Über eine Gesamtplanung aller ETL-Schritte sind genau solche lang laufenden „SELECT"-Phasen zu identifizieren und so zusammenzufassen, dass sie nach Möglichkeit nur einmal ausgeführt werden müssen oder ein Nachladen bei mehrmaligem Lesen durch das automatische Cachen der Datenbank nicht nötig ist.

In dem oben gezeigten Test wurde vor jedem Aufruf der Buffer-Cache geleert (ALTER SYSTEM FLUSH BUFFER_CACHE). Macht man das nicht, so halbieren sich zumindest für „DIRECT PATH"-Läufe die Zeiten. Das bedeutet, dass man aufwändige Lese-Operationen aus den gleichen Tabellen in zeitlich unmittelbarer Abfolge bündeln sollte, weil damit eine große Chance besteht, dass unmittelbar zuvor gelesene Tabellen sich noch im Hauptspeicher der Datenbank befinden (Fall 14).

Das zusätzliche Cachen bestimmter Tabellen (ALTER TABLE name CACHE) kann nützlich sein. Es behindert jedoch die Automatik, mit der Oracle häufig genutzte Tabellen sowieso im Hauptspeicher vorhält, weil es die für diese Dynamik bereitstehende Speichergröße verkleinert.

Umgang mit aufwändigen Join-Operationen

Müssen mehrmals im Verlauf des gesamten ETL-Prozesses Daten aus einem aufwändigen Join gelesen werden, dann kann eine einmal erstellte temporäre Join-Tabelle mit den wichtigsten Join- und Abfragekriterien helfen. Eine solche Tabelle besteht oft nur aus wenigen Spalten und umfasst nur einen Bruchteil des Datenvolumens des kompletten Joins. Das System kann eine solche kleine Tabelle besser im Hauptspeicher aufbewahren, also automatisch cachen. Kleine Join-Tabellen helfen bei immer wiederkehrenden Lookup-, Childless- und Orphan-Prüfungen, also immer dann, wenn Beziehungen im Spiel sind.

Die Reihenfolge der Prüfungen

Die oben aufgelisteten Prüf-Kategorien sollten in einer bestimmten Reihenfolge abgearbeitet werden, denn über eine geschickte Abfolge kann man die Gesamtlaufzeit ebenfalls minimieren:

1. Zunächst sind Formatprüfungen auf ihre Notwendigkeit hin zu überprüfen: Ist die Datenquelle eine Datenbank, in der Daten bereits formatgerecht abgelegt sind, dann muss man solche Format-Prüfungen in der Data-Warehouse-Datenbank nicht mehr wiederholen.
2. Daten aus Textdateien müssen dagegen geprüft werden. Dabei sollte man die Prüflogik der External Tables beziehungsweise des SQL-Loader nutzen. Man prüft also bereits, bevor die Daten in die Datenbank gelangen. Prüfungen des SQL-Loader beziehungsweise der External Tables sind schneller als Formatprüfungen von Sätzen, die bereits in der Datenbank gespeichert sind, zumal die Datenbank keine „is_Date"- oder „is_Numeric"-Funktion kennt. Solche Funktionen muss man selbst schreiben und sie als „Einzelsatz-Select" aufrufen, sodass die Prüfung in der Datenbank aufwändig werden kann.
3. Innerhalb der Datenbank sind als Erstes immer Beziehungsabhängigkeiten, also Orphan, Childless bezie-

hungsweise satzübergreifende Zusammenhänge wie Eindeutigkeiten zu prüfen. Dies ist meist mengenbasiert durchführbar. Sätze, die diesen als schnell eingestuften Prüfungsschritt nicht passieren, belasten auch nicht mehr die nachfolgenden langsameren Einzelfeld-Prüfungen.

4. Als Letztes sind Einzelfeld-Prüfungen, Format-Prüfungen etc. durchzuführen. Solche Prüfungen sind oft nur als Einzelsatzverarbeitung mit Funktionsaufrufen oder „CASE"-Strukturen möglich. Von solchen Einzelfeld-Prüfungen sollte man möglichst viele in einem einzigen Verarbeitungsschritt bündeln. Denn man ist bereits in der Einzelsatzverarbeitung und in dieser sollte man Sätze nur einmal anfassen.

Prüfen mit Error-Log-Tabellen

Seit den jüngsten Releases der Oracle-Datenbank kann man sogenannte „Error-Log-Tabellen" zusätzlich zu den Ziel-Tabellen definieren. Darin sammelt man fehlerhafte Sätze, die einen aktivierten Constraint verletzt haben. Error-Log-Tabellen können bei kleinen Tabellen hilfreich sein. Sie bremsen jedoch bei Massendaten, weil bei jeder Constraint-Verletzung ein Insert in die Fehler-Tabelle erfolgen muss. Die Constraint-Prüfung an sich verhindert darüber hinaus den Direct-Path-Load (außer „Not Null" und „Unique Key Constraints").

Umgang mit sehr großen Tabellen

In den meisten Fällen finden wir in einem Data Warehouse wenige Tabellen, die sehr groß sind (Bewegungsdaten- und Fakten-Tabellen), während die Masse der Tabellen klein ist (Stammdaten- und Referenz-Tabellen). Große Tabellen sind auch oft diejenigen mit dem größten Ladeanteil während des ETL-Prozesses. Für diese lohnt sich daher auch eine besondere Behandlung. Will man die oben beschriebenen Performance-Effekte nutzen, so sollte man neue Daten für große Tabellen zunächst in neu erstellte temporäre Tabellen schreiben. „Create Table As Select" (CTAS) beziehungsweise das Schreiben in leere Tabellen mit dem Direct-Path-Load sind, wie gesagt, die schnellsten Schreibvorgänge. Diese temporären Tabellen können dann an die eigentliche Ziel-Tabelle über das „Partition Exchange and Load Feature" (PEL) als weitere Partition angeschlossen werden. Dies zeigt, dass die Partitionierung großer Tabellen im Data Warehouse nicht nur für Performance-Effekte während des Lesens wichtig ist, sondern auch für die Beschleunigung des ETL-Prozesses.

Physische Struktur und Eigenschaften der DWH-Tabellen

Data-Warehouse-Tabellen erfahren in der Regel keine „UPDATE"-Operationen. Daher sollten sie grundsätzlich mit „PCTFREE=0" angelegt sein. Es werden also nach Möglichkeit alle Blöcke zu 100 Prozent vollgeschrieben. Das minimiert die Schreibzugriffe und bei nachfolgenden Lesevorgängen die physikalischen Lesezugriffe auf die Speicherplatte um bis zu 10 Prozent. Die Tabellen im Beispiel weiter oben haben mit „PCTFREE=0" anstatt 190.000 nur noch 170.000 Blöcke und anstatt 1,7 GB nur noch 1,46 GB Volumen.

Index-free

Man sollte das Schreiben in eine Tabelle mit aktiviertem Index vermeiden, weil der Index immer aktuell gehalten wird. Man löscht also den Index vor einem Massen-Load beziehungsweise setzt ihn auf „UNUSABLE", um ihn nach dem Laden entweder neu anzulegen oder mit „REBUILD" zu aktualisieren (ALTER INDEX name REBUILD). Generell sollte man die Notwendigkeit von Indizes im Data Warehouse überprüfen.

Temporäre Tabellen im Integration-Layer (Stage) und die meisten Tabellen im Enterprise-Layer (Data-Warehouse-Kern-Schicht) haben keine Index-Definitionen. Im Enterprise-Layer verfügen nur Stammdaten- und Referenzdaten-Tabellen über einen „Btree"-Index, der nach Abschluss des Ladevorgangs zu aktualisieren ist. Große Bewegungsdaten-Tabellen brauchen normalerweise keinen Index. Im User-View-Layer (Data Marts) gibt es auf den Primary-Key-Feldern der Dimensionen „Btree"-Indizes, auf den übrigen Feldern „Bitmap"-Indizes und auf den „FK"-Feldern der Fakten-Tabellen „Bitmap"-Indizes, die allerdings im Rahmen des Partition Exchange Load (PEL) sehr schnell aktualisiert werden können.

Kennzahlen und Aggregat-Tabellen

Neben Fakten-Tabellen gibt es häufig auch noch fest strukturierte Kennzahlen- oder Aggregat-Tabellen im User-View-Layer beziehungsweise in den Data Marts. Diese Tabellen enthalten über bereits bekannte Algorithmen erstellte Berichtsdaten. Solche Aggregat-Tabellen sollte man nicht mit ETL-Prozeduren oder mit ETL-Tool-Mappings herstellen. Dafür gibt es die Materialized Views in der Datenbank. Diese können sich bei Bedarf selbst aktualisieren und nutzen gegebenenfalls auch das Partition-Change-Tracking (PCT), mit dem nur Deltadaten aktualisiert werden müssen. Das ist meist schneller und spart ETL-Verwaltungsaufwand.

Das Schichtenmodell hilft bei der Planung von ETL-Prozessen

Die vorgenannten Schritte zeigen bereits auf, dass eine Gesamtübersicht über alle Lade- und Transformations-Prozesse im Data Warehouse sehr wichtig ist. Lade- und Prüfschritte an irgendwelchen beliebigen Stellen durchzuführen, kann die Gesamtladezeit um ein Vielfaches verlängern.

Das Schichtenmodell strukturiert den Informationsbeschaffungsprozess in einem Data Warehouse in folgende Phasen:

- Prüfen/harmonisieren (Stage- oder Integration-Layer)
- Übergreifend integrieren (Warehouse-Schicht oder Enterprise-Layer)
- Sachgebietsbezogen zusammenfassen (Data Mart oder User-View-Layer)

Zusammen mit den Datenmodellen der Enterprise- und User-View-Layer sollte deutlich sein, an welcher Stelle eine entsprechende Information benötigt wird, und von wo sie zu beziehen ist. Das nutzt man und konzentriert alle prüfenden Vorgänge im Integration-Layer und noch davor. Man sucht also für die Platzierung von Transformationen und Prüfungen die frühestmögliche Stelle im Gesamtprozess, um mög-

lichst viele nachfolgende Stellen davon profitieren zu lassen. Wer erst beim Wechsel in den User-View-Layer prüft, der provoziert potenziell doppelte und inkonsistente Prüfungen. Auf dem Weg in den User-View-Layer sollte es nur noch zusammenführende (Joins und) Lookup-Operationen geben.

Günstige Rahmenbedingungen für den ETL-Prozess schaffen

Die genannte Strukturierung gelingt nur, wenn sich alle Schichten des Data Warehouse in einer Datenbank befinden. Das Auslagern von Data Marts auf separate Maschinen verhindert die gemeinsame Nutzung von sehr großen Tabellen, die man am geschicktesten im Enterprise-Layer belässt und aus den Data Marts heraus referenziert. Auch das spart aufwändige Ladeprozesse.

Die Verwendung von separaten ETL-Servern (oft bei einem Einsatz von ETL-Tools üblich) zersplittert den Ladeprozess. Das Zusammenfassen temporärer Tabellen wird erschwert und die Speicherausnutzung der Datenbank verhindert; mengenbasierte Prüfungen in der Datenbank sind unmöglich.

ETL-Tools

Der Einsatz von ETL-Tools, die außerhalb der Datenbank arbeiten, führt fast immer zu langsameren Ladeprozessen. ETL-Tools haben gegenüber dem Laden in der Datenbank nur einen Vorteil: Sie bieten eine übersichtlichere Dokumentation und damit unter Umständen eine schnellere Implementierung und Wartung. Als Kompromiss wird man aufwändige Lade-Aktivitäten in die Datenbank legen und diese von außen über das jeweilige ETL-Tool steuern. Dieser Weg wird bei fast allen größeren Data-Warehouse-Umgebungen mit performancekritischen Ladeprozessen beschritten.

Alfred Schlaucher
alfred.schlaucher@oracle.com

Seit Februar 2013 steht MySQL 5.6 als Produktions-Release zur Verfügung. Diese Version der populärsten Open-Source-Datenbank der Welt gilt als erstes vollständig unter der Führung von Oracle entwickeltes MySQL-Release. MySQL 5.6 bietet bessere Leistung, Skalierbarkeit, Zuverlässigkeit und Verwaltbarkeit und unterstützt Anwender dabei, die anspruchsvollsten Anforderungen an Web-, Cloud- und integrierte Anwendungen zu erfüllen. Ein großer Schwerpunkt liegt dabei auf der Verbesserung und Erweiterung der MySQL-Replikation.

Neu: MySQL 5.6 GA

Jürgen Giesel und Mario Beck, Oracle B.V. & Co. KG

Dank der Möglichkeit zum Einsatz auf heterogenen Plattformen und Anwendungs-Stacks sowie aufgrund der hohen Leistung, Zuverlässigkeit und Benutzerfreundlichkeit basieren neun von zehn der beliebtesten und am stärksten frequentierten Websites der Welt auf MySQL. MySQL 5.6 setzt diesen Trend fort, indem umfassende Verbesserungen eingeführt werden, die es innovativen DBAs und Entwicklern ermöglichen, die nächste Generation von Web-, Embedded- und Cloud-/SaaS-/DaaS-Anwendungen basierend auf den neuesten Frameworks und Hardware-Plattformen zu entwickeln und bereitzustellen. Kurz gefasst handelt es sich bei MySQL 5.6 um eine noch bessere MySQL-Version mit Neuerungen, die jeden Funktionsbereich des Datenbankkernels erweitern, darunter:

- Bessere Leistung und Skalierbarkeit
  - Verbesserte InnoDB-Speicher-Engine für besseren Transaktionsdurchsatz
  - Verbesserter Optimizer für eine bessere Abfrage-Ausführung in Bezug auf Ausführungszeiten und Diagnose
- Höhere Anwendungsverfügbarkeit durch DDL-/Schema-Änderungen im laufenden Betrieb
- Gesteigerte Entwickler-Flexibilität dank NoSQL-Zugriff mit Memcached-API für InnoDB
- Verbesserte Replikation für verteilte Bereitstellungen mit hoher Leistung und Selbst-Reparaturfunktion
- Verbessertes Performance-Schema für bessere Analysen
- Erhöhte Sicherheit für sorgenfreie Anwendungsbereitstellungen
- Weitere wichtige Verbesserungen

Verbesserte InnoDB-Speicher-Engine

Aus operativer Sicht bietet MySQL 5.6 eine bessere, fortlaufend lineare Leistung und Skalierbarkeit auf Systemen, die mehrere Prozessoren und eine hohe Anzahl von gleichzeitig ausgeführten CPU-Threads unterstützen. Der Grund für diese Verbesserungen liegt in der Effizienz und Leistung der InnoDB-Speicher-Engine von Oracle, die Thread-Konflikte und Mutex-Sperren im InnoDB-Kernel vermeidet. Diese Verbesserungen ermöglichen es MySQL, die Vorteile der Multi-Threading-Prozessoren von modernen, x86-basierten Standard-Hardwarekomponenten vollständig auszuschöpfen.

Interne SysBench-Benchmark-Tests in Bezug auf Lese-/Schreib-Operationen und reine LeseOperationen zeigen eine deutliche und anhaltende Verbesserung gegenüber der aktuellen Version von MySQL 5.5. Nachfolgend wird aufgezeigt, dass MySQL 5.6 eine deutlich gesteigerte und fortlaufend höhere Anzahl an Lese-/Schreibtransaktionen pro Sekunde (Transactions per Second, TPS) auf Systemen bietet, die mehr als 60 gleichzeitige CPU-Threads unterstützen.

Besserer Transaktionsdurchsatz

MySQL 5.6 bietet dank verbesserter InnoDB-Speicher-Engine auch höhere Leistung und Skalierbarkeit für viele gleichzeitige Operationen, transaktionale und leseintensive Arbeitslasten.

In diesen Fällen können die Leistungsverbesserungen am besten an Anwendungsverhalten und -skalierbarkeit bei steigenden Benutzer-Arbeitslasten gemessen werden. Zur Unterstützung dieser Anwendungsfälle verfügt InnoDB über eine neu gestaltete Architektur, die Mutex-Konflikte sowie Engpässe minimiert und einen konsistenteren Zugriffspfad für die zugrunde liegenden Daten bietet.

Bessere Leistung dank SSD-Speicher

Festplatten gehören zu den häufigsten Verursachern von Engpässen auf jedem System, da sie über mechanische Teile verfügen, durch die die Fähigkeit zur Skalierung bei steigender Anzahl gleichzeitiger Vorgänge physisch begrenzt wird. Viele MySQL-Anwendungen werden deshalb auf SSD-aktivierten Systemen bereitgestellt, die genau die arbeitsspeicherbasierte Geschwindigkeit und Zuverlässigkeit bieten, die zur Unterstützung der hohen Ansprüche moderner, webbasierter Systeme erforderlich sind. MySQL 5.6 bietet verschiedene wichtige Verbesserungen, die speziell für den Einsatz mit SSD entworfen sind, darunter:

- Unterstützung für kleinere 4-KB- und 8-KB-Seitengrößen
- Portable „.ibd“-Dateien (InnoDB-Daten), die es ermöglichen, häufig genutzte InnoDB-Tabellen vom Standard-Datenverzeichnis auf SSD- oder Netzwerk-Speichergeräte zu verschieben.
- Getrennte Tablespaces für das InnoDB-Undo-Log

DDL-/Schema-Änderungen im laufenden Betrieb

Die modernen, webbasierten Anwendungen sind so entworfen, dass sie eine schnelle Änderung und Anpassung an Geschäftsanforderungen zur Umsatzsteigerung ermöglichen. Aus diesem Grund werden Entwicklungs-SLAs häufig nicht mehr in Tagen oder Wochen, sondern in Minuten gemessen. Wenn für eine Anwendung eine schnelle Unterstützung für neue Produktlinien oder neue Produkte innerhalb vorhandener Produktlinien gefordert ist, muss das Back-End-Datenbankschema entsprechend angepasst werden – und die Anwendung muss während der Anpassung für den normalen Geschäftsbetrieb verfügbar bleiben. MySQL 5.6 unterstützt diese Flexibilität in Bezug auf Schema-Änderungen im laufenden Betrieb bei verschiedenen „ALTER TABLE“-Operationen.

NoSQL-Zugriff auf InnoDB

Viele der Web-, Cloud-, Social-Media- und mobilen Anwendungen der neuesten Generation erfordern schnelle Schlüssel-/Wert-Operationen. Gleichzeitig müssen sie weiterhin die Fähigkeit bieten, für dieselben Daten auch komplexe Abfragen auszuführen, und sicherstellen, dass die Daten mithilfe von ACID-Garantien geschützt werden. Mit dem neuen NoSQL-API für InnoDB können Entwickler von sämtlichen Vorteilen eines transaktionalen RDBMS und den Leistungsmerkmalen des Schlüssel-/Wert-Speichers profitieren.

MySQL 5.6 bietet über das bekannte Memcached-API eine einfache Schlüssel-/Wert-Interaktion mit InnoDB. Die Implementierung erfolgt über ein neues Memcached-Daemon-Plug-in zu „mysqld“ und das neue Memcached-Protokoll wird direkt in das native InnoDB-API umgesetzt. So können Entwickler vorhandene Memcached-Clients zur Umgehung des Abfrage-Parsing nutzen und für simple Lesezugriffe und transaktionale Updates direkt auf InnoDB-Daten zugreifen. Das API ermöglicht es, standardmäßige Memcached-Bibliotheken und -Clients wiederzuverwenden, während die Memcached-Funktionalität gleichzeitig durch die Integration eines persistenten, absturzsicheren transaktionalen Datenbank-Backends erweitert wird.

Verbesserte Replikation und Hochverfügbarkeit

Die Replikation ist die am häufigsten verwendete MySQL-Funktion für die horizontale Skalierung und zur Erzielung von Hochverfügbarkeit. MySQL 5.6 umfasst neue Funktionen, die es Entwicklern ermöglichen, Web-, Cloud-, Social-Media- und mobile Anwendungen und Dienste der nächsten Generation zu entwickeln, die Replikationstopologien mit Selbstreparaturfunktionen sowie Master und Slaves mit hoher Leistung bieten. Die Neuheiten im Bereich der MySQL-Replikation umfassen:

- *Selbstheilende Replikations-Cluster*
  Die Aufnahme von Global Transaction Identifiers und Utilities bietet das einfache, automatische Erkennen von Ausfällen und deren Behebung. Ausfallsichere Replikation ermöglicht es Slave-Systemen, im Falle eines Absturzes automatisch ihre korrekte Position im Replikations-Stream wiederzufinden und mit der Replikation fortzufahren, ohne dass ein Administrator-Eingriff notwendig wäre.
- *Hochleistungsfähige Replikations-Cluster*
  Eine bis zu fünfmal schnellere Replikation dank Multi-Threaded Slaves, Binlog Group Commit und einer optimierten, zeilenbasierten Repli-

„... eine wichtige Neuerung“

Die Meinung von Matthias Jung, Leiter der DOAG SIG MySQL, zur neuen Version: „Aus meiner Sicht ist die Version 5.6 ein Schritt in die richtige Richtung. Gerade die Erweiterung des „performance_schema“ und die damit verbundenen Analyse-Möglichkeiten sind ein wichtige Neuerung. Die Verbesserungen der Storage-Engine InnoDB lesen sich zunächst einmal sehr gut auf dem Papier. Ob sich dieser Performance-Boost auch beim Kunden einstellen wird, bleibt abzuwarten. Auch einige Neuerungen im Replikationsbereich (vor allem die Global Transaktion Identifier und die dazugehörigen Tools und das Group Commit) sind eine echte Bereicherung. Dem Feature der Multi-Threaded-Slaves stehe ich persönlich skeptisch gegenüber. Hier gibt es aus meiner Sicht bessere Lösungen (Galera Cluster; Galera Plug-in).“

kation ermöglicht es Nutzern, die Leistung und Effizienz der Replikation während der Skalierung ihrer Arbeitslasten zu maximieren.
- *Zeitverzögerte Replikation*
  Diese bietet Schutz gegen operationale Fehler auf Master-Ebene wie ein versehentliches DROP TABLE.

Jürgen Giesel
juergen.giesel
@oracle.com

Mario Beck
mario.beck
@oracle.com

Oracle hat Mitte 2010 die Business Intelligence Suite Version 11*g* veröffentlicht. Neben neuen Funktionalitäten aus Anwender- und Entwicklersicht ergaben sich die meisten Änderungen auf der administrativen Seite. Durch die neue Plattform Fusion Middleware und den WebLogic Server sind moderne Architekturansätze etabliert worden. Dieser Artikel richtet sich an Entscheider und Administratoren, die noch die Vorgängerversion 10*g* einsetzen und eine Migration in absehbarer Zeit planen.

Migration Oracle BI Suite 10*g* auf 11*g* – Vorgehen und Fallstricke

Matthias Kietzke, OPITZ CONSULTING Deutschland GmbH

Die Oracle Business Intelligence Suite (OBI) ist eine Softwarelösung zur Realisierung eines unternehmensweiten Berichtswesens. Neben Standardberichten können Ad-hoc-Analysen und komplexe Dashboards für Endanwender erstellt werden. Sie besteht aus mehreren Komponenten, die es im Rahmen einer Migration zu beachten gilt (siehe Tabelle 1).

Eine vorhandene OBI 10*g* kann nicht in die Nachfolge-Version 11*g* umgewandelt werden – stattdessen müssen OBI 11*g* neu installiert und die Komponenten nacheinander in die neue Version übertragen werden. Dafür bietet Oracle ein grafisches Tool namens „Upgrade Assistant". Damit können die Komponenten „Berichtskatalog", „Metadaten-Repository", „Delivers" und „BI Publisher" migriert werden. Hierzu selektiert der Administrator die Quelldaten der Vorgängerversion und gibt das Zielsystem (11*g*) an, um im anschließenden Migrationsprozess die Inhalte automatisiert zu überführen.

Für die Konfigurations-Dateien, die Usage-Tracking-Daten sowie das Oberflächen-Layout gibt es kein Hilfstool, sie müssen manuell übertragen werden. Nachfolgend eine Übersicht über die Besonderheiten der einzelnen Komponenten.

Berichtskatalog

Die Definition der Berichte und Dashboards erfolgt im XML-Stil, wobei jedes Objekt durch eine separate Datei abgebildet ist. Zu jedem Objekt gehört eine Attribut-Datei, in der die Zugriffsberechtigungen hinterlegt sind. Die XML-Definitionen und die Attribut-Dateien werden mithilfe des Upgrade Assistant migriert.

Die gute Nachricht ist, dass ein Großteil der Berichte und Dashboards anschließend problemlos funktioniert. Strukturen, Prompts (Filter), Diagramme und viele Layout-Einstellungen werden übernommen. Die schlechte Nachricht ist, dass einige Berichte wahrscheinlich nicht funktionieren werden, wobei die Ursachen sehr vielfältig sind. Beispielsweise werden Werte für Abfragegrenzen überschritten, die in OBI 10*g* noch nicht existierten. Ein Bericht, der 10.000 Datensätze zurückliefert und eine komplexe Diagramm-Form beinhaltet, kann unter Umständen in OBI 11*g* eine Fehlermeldung bringen. Für die Konfiguration dieser Grenzwerte stehen vielfältige Parameter zur Verfügung, was die Komplexität der Handhabung erschwert.

Eine weitere Fehlerquelle stellt die dynamische Berechnung von Zeiträumen dar. In OBI 10*g* kann vom aktuellen Systemtag („current_date") direkt ein Wert abgezogen werden (etwa „current_date-7"). Diese Funktion wird häufig in Filtern genutzt, um die Daten der letzten Woche anzuzeigen. In OBI 11*g* muss zwingend die Funktion „timestampadd" verwendet werden. Diese steht bereits in OBI 10*g* zur Verfügung, wird jedoch selten angewendet.

Neben diesen technischen Fehlern existieren zahlreiche Änderungen, die das Berichts-Layout betreffen. So hat Oracle in OBI 11*g* Standards abweichend zu OBI 10*g* umgesetzt. Numerische Werte werden grundsätzlich zunächst mit einer Nachkomma-Stelle dargestellt. Darüber hinaus wird in jedem Eingabe-Prompt automatisch eine Schaltfläche angezeigt, mit der die Auswahl gelöscht werden kann („reset filter"). Zudem werden Diagramme, die auf Pivot-Tabellen basieren, in OBI 11*g* anhand der ersten beiden Kennzahlen-

Komponente	Beschreibung
Berichtskatalog (Answers)	Enthält alle Berichte, Dashboards und weitere Objekte, die für das Berichtswesens verwendet werden
Metadaten-Repository (RPD)	Enthält die logischen und physikalischen Strukturen der Datenmodelle, die als Basis für Abfragen verwendet werden
Delivers	Stellt Funktionen zum automatisierten Versand von Berichten zur Verfügung
BI Publisher	Ist eine separate Anwendung zur Erstellung von druckreifen Berichten, die in die Oracle BI-Suite integriert ist
Konfigurations-Dateien	Enthalten Parameter der verschiedenen Komponenten
Usage Tracking	Sammelt Informationen über die Nutzung der Berichte (Anzahl der Berichtsaufrufe, Dauer der Antwortzeiten etc.)
Oberflächen-Layout	Definiert mithilfe von Stylesheet-Dateien (CSS) und Grafiken das Aussehen der Bedienoberfläche
Client-Tools	Bezeichnen unterstützende Programme, mit denen der Anwender arbeitet, wie BI-Administration-Tool, Katalog-Manager, Microsoft-Office-Plug-in oder BI Publisher Template Builder

Tabelle 1

Spalten sortiert. Dies kann dazu führen, dass die Werte auf den Achsen eine andere Reihenfolge aufweisen. Zusammengefasst sind diese Punkte lösbar, je nach Häufigkeit, Komplexität und Gewohnheit der Anwender bedarf es jedoch entsprechender Nacharbeit.

Metadaten-Repository (RPD)

Das Metadaten-Repository ist eine binäre Datei, die mit dem BI-Administration-Tool gelesen und bearbeitet werden kann. Das Repository umfasst drei Ebenen:

- Die Präsentations-Ebene enthält Objekte, auf deren Basis die Berichte und Dashboards entwickelt werden
- Die Geschäftsmodell-Ebene repräsentiert die Geschäftsdomäne und enthält die logischen Objekte (wie Kunde, Zeit, Kennzahlen)
- Die physikalische Ebene enthält die Strukturen der verschiedenen Datenquellen (Data Marts)

OBI bietet die Möglichkeit, das Metadaten-Repository auf Konsistenz zu prüfen. Wenn das Ausgangs-Repository (10*g*) konsistent und somit strukturell fehlerfrei ist, verläuft die Überführung in das 11*g*-Repository mittels Upgrade Assistant ebenfalls reibungslos.

Die Initialisierungsblöcke sowie die Session- und Repository-Variablen bleiben erhalten; ebenso die im 10g-Repository definierten Benutzer und Gruppen, die dort jedoch nicht mehr relevant sind. Auf diesen Aspekt wird im Abschnitt „Sicherheitskonzept" noch einmal näher eingegangen.

Delivers

Über Delivers werden Berichte oder ganze Dashboards an bestimmte Empfänger automatisiert zugestellt, häufig per E-Mail. Die Konfiguration der Zustellung, etwa mit Angaben zu Inhalt und Empfänger, wird in sogenannten „iBots" hinterlegt. Diese sind Teil des Berichtskatalogs und werden auch problemlos mit diesem migriert. Die zeitliche Konfiguration, also die Angabe darüber, wann welcher Bericht ausgeführt werden soll, liegt nicht im Berichtskatalog vor, sondern ist in Datenbank-Tabellen abgelegt. Diese Einträge überträgt der Administrator mithilfe von Upgrade Assistant in das Oracle-BI-Metadaten-Schema, das im Rahmen der Installation angelegt wird. Es ist zu beachten, dass die Einträge (Jobs) direkt nach der Migration aktiv sind. Es kann also sein, dass bei gültiger SMTP-Konfiguration bereits Mails versendet werden.

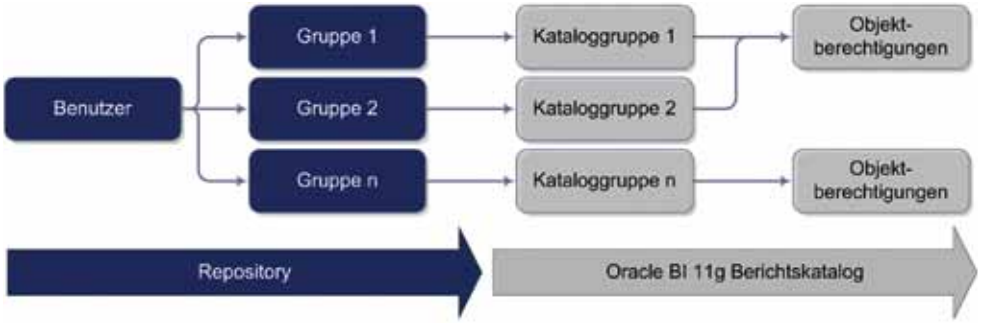

Abbildung 1: Sicherheitsmodell in OBI 10g

BI Publisher

Der BI Publisher ist eine separate, jedoch integrierte Software, die in OBI 11*g* noch stärker eingebunden wurde. Publisher-Objekte wie Datenmodelle und zugehörige Layouts werden mit Upgrade Assistant migriert. Die Integration des BI Publisher in OBI funktioniert jedoch nur fehlerfrei, wenn der Publisher-Katalog (Verzeichnis mit den Publisher-Objekten) ein Teil des OBI-Berichtskatalogs wird. Hierzu muss der Administrator die Publisher-Objekte nach der Migration in den OBI-Berichtskatalog laden. Dies wird über die Web-Oberfläche der Publisher-Administration durchgeführt.

BI-Publisher-Objekte bestehen aus zwei Teilen, dem Berichtslayout und dem Datenmodell. Das Datenmodell hat unter OBI 10*g* die Datei-Endung „xdm". Diese entfällt in OBI 11*g*, aus „datenmodell.xdm" wird die Datei „datenmodell". Nach Erfahrung des Autors verweisen die Berichtslayouts nach dem Hochladen in den OBI-Berichtskatalog noch auf die (nicht mehr gültigen) XDM-Dateien. Die Verweise sind durch die Zuweisung des gültigen Datenmodells manuell zu korrigieren.

Konfigurations-Dateien

Individuelle Einstellungen werden in OBI 10*g* in den Konfigurations-Dateien des BI Servers („NQSConfig.ini"), des Presentation Servers („instanceconfig.xml"), des BI Publisher („xmlpserver-config.xml" und „datasources.xml") sowie des BI Scheduler (eigene „instanceconfig.xml") vorgenommen. Möglicherweise kommen je nach Umgebung noch weitere Konfigurationsdateien hinzu. Diese sind unter OBI 11*g* weiterhin vorhanden. Inhaltlich haben sich die Dateien jedoch teilweise geändert, sodass ein Administrator individuelle Einstellungen manuell prüfen und übernehmen muss. Auch sind einige Einstellungen weggefallen und haben kein Äquivalent unter OBI 11*g*.

Wichtig: Unter OBI 11*g* gibt es eine zentrale Konfigurationsdatei, in der einige Einstellungen führend gehalten werden. Die Konfiguration dieser Werte sollte der Administrator ausschließlich über Fusion Middleware Enterprise Manager vornehmen, um Systemfehler zu vermeiden. Eine alternative Möglichkeit bietet das WebLogic Scripting Tool (WLST), eine Skriptsprache zum Durchführen administrativer Aufgaben. Zu erkennen sind die betroffenen Werte an dem Hinweis „<this configuration is centrally managed by oracle business intelligence>" in den Konfigurationsdateien.

Usage Tracking

Das Usage Tracking sammelt Informationen über die Nutzung der Berichte und Dashboards. Es protokolliert unter anderem, welcher Benutzer zu welcher Uhrzeit welchen Bericht ausführt, wie lange die Ausführung gedauert hat und wie viele Datensätze zurückgeliefert wurden.

Die Datenbank-Tabelle „S_NQ_ACCT", in der die Informationen gespeichert sind, hat sich strukturell minimal geändert. Diese Änderungen müssen im OBI-11*g*-Metadaten-Repository nachgezogen werden. Bei Bedarf sind die unter OBI 10*g* gesammelten Informationen manuell zu überführen. Das ist mit einem von Oracle gestellten SQL-Skript schnell erledigt.

Oberflächen-Layout

Um das Aussehen der Web-Oberfläche zu individualisieren und beispielsweise ein Firmenlogo einzublenden oder das Farbschema dem Corporate Design anzupassen, müssen in OBI 10*g* diverse Stylesheet-Dateien (CSS) modifiziert werden. Auch das Aussehen in OBI 11*g* lässt sich mithilfe von Stylesheet-Dateien anpassen. Es haben sich jedoch die Struktur der Dateien und deren Inhalte geändert, sodass die Dateien der OBI 10*g* nicht übernommen werden können. CSS-Klassen wurden umbenannt, gelöscht sowie neue hinzugefügt. Layout-Änderungen sind daher manuell zu übertragen. Hinzu kommt, dass die Strukturen der Stylesheet-Dateien nur rudimentär dokumentiert sind. Aus diesen Gründen schätzt der Autor den Aufwand für komplexe Design-Anpassungen hoch ein.

Sicherheitskonzept

Durch die Fusion-Middleware-Plattform und den Einzug des WebLogic Servers ergeben sich bedeutende Unterschiede bei den Sicherheitskonzepten. Aus 10g-Sicht bedeutet dies: Benutzer und Gruppen werden üblicherweise im Metadaten-Repository (RPD) verwaltet. In Answers (dem webbasierten Reporting-Frontend) existieren Katalog-Gruppen, denen Benutzer und Gruppen zugeordnet sind. Die Objekt-Berechtigungen im Berichtskatalog, die festlegen, wer welche Berichte öffnen oder bearbeiten, wer Dashboards anlegen oder wer administrative Aufgaben ausführen darf, sind an die Kataloggruppen gekoppelt (siehe Abbildung 1). Soll ein externer Authentifizierungsdienst wie beispielsweise Microsofts Active Directory verwendet werden, wird dieser über Initialisierungsblöcke direkt im Metadaten-Repository angebunden.

Soweit die vereinfachte Erläuterung der Sicherheits-Architektur in OBI 10*g*. In OBI 11*g* ist diese Darstellung nicht mehr ausreichend. Benutzer und Gruppen werden in OBI 11*g* über Authentifizierungsdienste des WebLogic Servers eingebunden. Der Administrator hat die Möglichkeit, neben dem WebLogic-internen LDAP-Server weitere Authentifizierungsdienste (sogenannte „Identity Stores") wie ein unternehmensweites Active Directory anzubinden und parallel zu verwenden. Weiterhin wurden Applikationsrollen eingeführt, an die standardmäßig Berechtigungen vergeben werden sollten. Eine Applikationsrolle kann Benutzer, Gruppen oder andere Applikationsrollen enthalten. Sie stellt eine Gruppierungsart oberhalb von Gruppen dar. Auf diese Weise lassen sich hierarchische Berechtigungskonzepte umsetzen. Zudem werden Applikationsrollen in Fusion Middleware definiert und applikationsübergreifend (etwa in der Oracle SOA-Suite) verwendet.

Bei der Migration des Berichtskatalogs ergibt sich eine Besonderheit: Die Kopplung der Objekt-Berechtigungen an Kataloggruppen, wie in OBI 10*g* realisiert, wird nach 11*g* übernommen. Das bedeutet, dass die Applikationsrollen für den Berichtskatalog vorerst keine Rolle spielen. Bei der Migration finden in dieser Hinsicht keine Änderungen statt. Da Benutzer und Gruppen jedoch Applikationsrollen zugeordnet sind, ergibt sich ein Bruch zwischen Kataloggruppen (relevant für Objekt-Berechtigungen im Berichtskatalog) und Applikationsrollen (enthal-

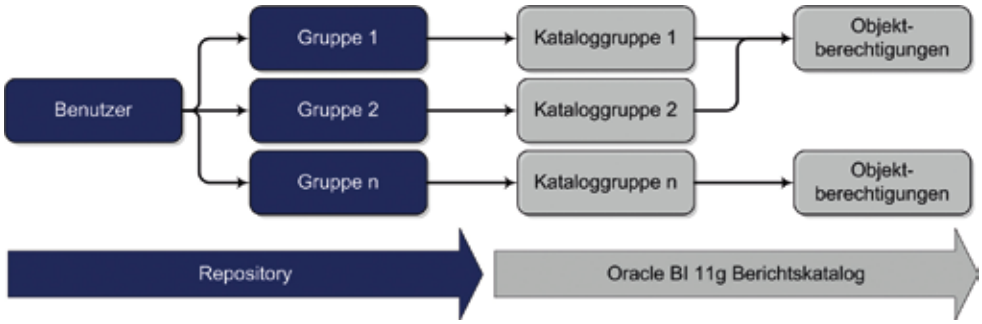

Abbildung 2: Sicherheitsmodell nach der Migration

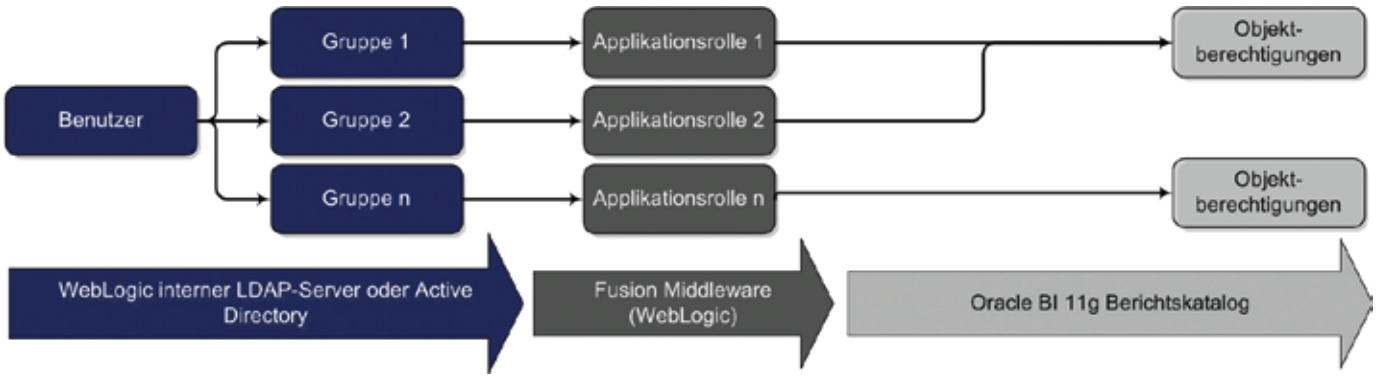

Abbildung 3: OBI-11g-Standard-Sicherheitsmodell

ten Benutzer und Gruppen). Um diesen Bruch zu schließen, können die Applikationsrollen den Kataloggruppen zugewiesen werden. Somit ergibt sich eine Kette, wie in Abbildung 2 dargestellt.

Da die Kataloggruppen nicht dem 11*g*-Standard entsprechen und nur aus Migrationsgründen erhalten geblieben sind, empfiehlt der Autor, die Objekt-Berechtigungen im Berichtskatalog direkt an die Applikationsrollen zu vergeben und die Kataloggruppen somit überflüssig zu machen (siehe Abbildung 3). Dieser Schritt ist vom Administrator manuell im Berichtskatalog durchzuführen und kann im Nachgang einer Migration erfolgen.

Administrative Skripte und Deployment

OBI 10*g* verwendet Oracle Containers for J2EE (OC4J) als Application Server. OBI 11*g* hingegen nutzt den WebLogic Server. Somit haben sich die Befehle zum Starten und Stoppen der BI-Software geändert. Auch das Deployment eines neuen Metadaten-Repository durch einfachen Austausch der Datei, wie in OBI 10*g* üblich, empfiehlt Oracle in OBI 11*g* nicht. Stattdessen sollte ein neues Metadaten-Repository entweder manuell über den Enterprise Manager oder automatisiert mithilfe des WebLogic Scripting Tool (WLST), einer Skriptsprache zur Bedienung des WebLogic Servers, eingespielt werden.

Um einzelne Katalogobjekte zwischen zwei OBI-11*g*-Systemen zu übertragen (beispielsweise von „Entwicklung“ auf „Test“), kann die Funktion „Archive & Unarchive“ verwendet werden. Dabei werden die Quellobjekte in eine binäre Archivdatei gepackt und am Ziel wieder entpackt. Diese Funktion stand in OBI 10*g* bereits im Katalogmanager zur Verfügung. In OBI 11*g* steht sie auch über die Web-Oberfläche bereit. Um einen gesamten Berichtskatalog zu übertragen, empfiehlt Oracle allerdings das Kopieren auf Dateiebene.

Client-Tools

Es gibt Client-Tools, die für administrative und Entwicklungsaufgaben verwendet werden. Hierunter fallen das Administration Tool, der Katalogmanager und der Jobmanager. Diese Tools werden von Oracle in einem Paket bereitgestellt und können direkt von der OBI-Startseite oder dem Oracle Technology Network (OTN) heruntergeladen werden. Der Administrator beziehungsweise Entwickler muss diese Programme auf seinem PC installieren.

Darüber hinaus existieren zwei weitere Tools, die separat aktualisiert werden müssen. Das Microsoft-Office-Plug-in dient dazu, Analysen von OBI nach Excel oder PowerPoint zu übernehmen und dort weiterzubearbeiten. Dieses Tool kann der Anwender von der OBI-Startseite herunterladen und auf seinem PC installieren. Anschließend ist über das Plug-in noch eine Verbindung zum OBI-Server einzurichten.

Das zweite Tool ist der Template Builder. Er dient dazu, druckreife Layouts in Word zu erstellen, die der BI Publisher mit Daten befüllt. Auch dieses Tool kann der Anwender von der OBI-Startseite herunterladen und lokal installieren. Es ist zu beachten, dass Oracle in Version OBI 11.1.1.6.x offiziell ausschließlich Microsoft Office 2010 unterstützt. In Version OBI 11.1.1.7.0 werden offiziell Microsoft Office 2003 und 2010 unterstützt.

Fazit

Ein Business-Intelligence-System von Oracle besteht aus einer Vielzahl von Komponenten. Diese Bestandteile sind vom Migrationsverantwortlichen einzeln zu beachten und zu bearbeiten. Durch die neue Architektur mit Fusion Middleware und WebLogic Server als zentralem Application Server, steigt die Komplexität des Gesamtsystems und somit der Umfang der Administration. Für die Migration des Berichtskatalogs, des Metadaten-Repository, der iBots und der BI-Publisher-Objekte bietet Oracle einen Assistenten. Dieser liefert größtenteils brauchbare Ergebnisse. Je nach Komplexität der Ausgangsobjekte sind im Anschluss jedoch entsprechende Nacharbeiten notwendig, um Mängel oder Fehler zu beseitigen. Die übrigen Komponenten wie beispielsweise Konfigurationseinstellungen oder administrative Skripte müssen manuell übertragen werden. Eine Testmigration im Rahmen eines Workshops kann helfen, die Herausforderungen im Vorfeld zu erkennen und die Aufwände zu beziffern.

Matthias Kietzke
matthias.kietzke@opitz-consulting.com

Tipps und Tricks aus Gerds Fundgrube

Heute: Record-Group-Spalten mit 4.000 Zeichen

Gerd Volberg, OPITZ CONSULTING GmbH

Abbildung 1: Select-Statement der Record Group

Record Groups, in denen man Package-Funktionen (siehe Abbildung 1) nutzt, haben die Eigenart, Spalten des Datentyps „Character 4000“ zu erzeugen (siehe Abbildung 2). Eine solche Record Group wird beim Kompilieren den Fehler „FRM-30187“ erzeugen (siehe Abbildung 3), da eine CHAR-Spalte nur maximal 2.000 Zeichen groß sein darf.

Der einfachste Workaround ist die manuelle Änderung der Länge von 4.000 auf 2.000. Danach ist die Maske wieder kompilierbar. Jede Änderung am Select-Statement der Record Group wird aber sofort wieder den Wert 4.000 erzeugen, was erneut manuell korrigiert werden muss. Der einfachste Weg, dieses Problem dauerhaft zu lösen, besteht darin, die Funktion „Substring“ zu benutzen: „select substr (adre.get_ort (adre_id), 1, 100) adre_ort, adre_id from adressen“. Änderungen am Select-Statement werden in der Folge die Länge der Spalte nicht mehr verändern (siehe Abbildung 4).

Wenn man diesen Substring in einer View versteckt, kann man noch eleganter auf diese Spalte zugreifen (siehe Listing 1).

Das Select-Statement, das man bei dieser View benutzen würde, käme dann ohne Substring aus, da diese Funktion in der View gekapselt ist: „SELECT adre_ort, adre_id FROM Adressen_View“.

Abbildung 3: Fehlermeldung beim Kompilieren

Abbildung 2: Record-Group-Details

Drei Workarounds helfen also bei der Lösung dieses Problems. Der Autor würde immer die Variante bevorzugen, bei der der Substring in der View ver-

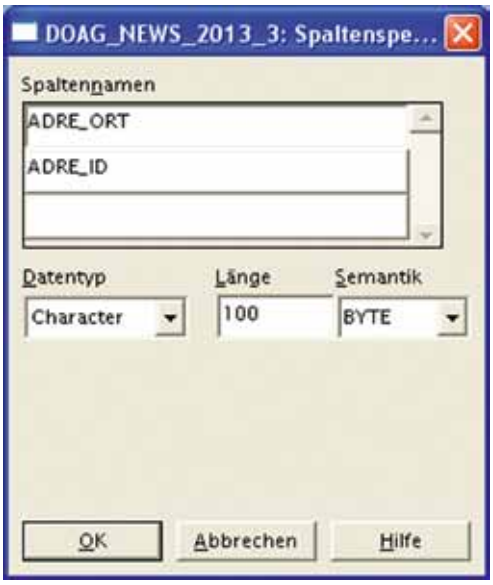

Abbildung 4: Record Group mit korrekter Länge

steckt wird. Diese benötigt zwar den größten Vorbereitungsaufwand, der sich aber auf Dauer auszahlt.

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

```
CREATE OR REPLACE FORCE VIEW Adressen_View (adre_id, adre_ort) AS
SELECT adre.adre_id adre_id,
       substr (adre.get_ort (adre_id), 1, 100) adre_ort
  FROM Adressen adre;
```

Listing 1

In der Rubrik „Frauen in der IT" stellt die DOAG *News* Frauen vor, die erfolgreich im IT-Bereich arbeiten. Ziel ist es, mehr Frauen für die IT-Berufe zu interessieren und ihnen dort auch eine Arbeitsumgebung anzubieten, die Familie und Berufe besser vereinbaren lässt.

„Frauen haben es nicht nötig, erfolgreiche Männer „1:1" zu kopieren …"

Welchen Beruf üben Sie aus?
Hayek: Ich bin freiberufliche Oracle Apex-Applikations-Entwicklerin und arbeite hauptsächlich für Institute der Universität Innsbruck.

Auf welchem Weg sind Sie dorthin gekommen?
Hayek: Nach einem abgeschlossenen Pharmazie-Studium und anschließendem Doktorats-Studium in Chemie arbeitete ich zunächst als Universitäts-Assistentin am Institut für Medizinische Chemie. Um Familie und Beruf unter einen Hut zu bringen, begann ich nach der Geburt meiner beiden Kinder von zu Hause aus für das Beilstein-Institut für Organische Chemie in Frankfurt beim inhaltlichen Aufbau der Crossfire-Datenbank zu arbeiten. Wenige Jahre später nahm ich eine Anstellung am Zentralen Informatikdienst der Universität Innsbruck als Web-Applikations-Entwicklerin mit Oracle-Datenbanken an.

Was hat Sie motiviert, diesen Beruf zu ergreifen?
Hayek: Zunächst einmal: meine Bereitschaft etwas Neues zu probieren, das Leben nicht in vorgegebenen Bahnen zu sehen und Herausforderungen positiv aufzugreifen. Dann aber auch: die gute Vereinbarkeit von Beruf und Familie; die kreative Tätigkeit bei der selbstständigen Entwicklung von Applikationen; die überaus positive Kooperation mit den verschiedenen Universitäts-Instituten; die ständige Horizont-Erweiterung bei der Auseinandersetzung mit verschiedensten Forschungsgebieten und beim Einarbeiten in neue Sachthemen.

Wie sehen Sie generell die Rolle der Frau in der IT?
Hayek: Frauen werden als technisch weniger interessiert und versiert angesehen. Das dürfte zu einem gewissen Maß auch stimmen. Ich habe jedoch in meiner langjährigen Berufserfahrung festgestellt, dass technikverliebte Männer oft den Wald vor lauter Bäumen nicht mehr sehen, also im Grunde für einfache Probleme einen Lösungsansatz mit enormem technischen Aufwand suchen.

Bietet die IT-Branche für Frauen die Möglichkeit, ihre Stärken einzusetzen?
Hayek: Frauen könnten dazu beitragen, mit gewissen Vorurteilen, die ganz allgemein der IT gegenüber herrschen, aufzuräumen: Frauen könnten oft einen pragmatischeren Ansatz bei Problemlösungen finden. Unkonventionelle, kreative Denkansätze, mehr Sozial- und mehr Sprachkompetenz könnten Barrieren zwischen der IT-Welt und den sogenannten „Dummies" abbauen. Das bedeutet auch: Weniger abgehobene verbale Kommunikation mit technisch nicht so versierten Kunden kann

Zur Person: Ingrid Hayek
Ingrid Hayek stürzte sich 1978 in die damals noch von Männern dominierte Welt der Chemie. Dort gefiel es ihr, sie lernte ihren Mann kennen, gebar zwei Kinder und verbrachte mit ihm mehrere Forschungsaufenthalte in Berkeley, Kalifornien. Über die Mitarbeit an der Datenbank für Organische Chemie am Beilstein Institut in Frankfurt und über den Zentralen Informatikdienst der Universität Innsbruck glitt sie allmählich in die, ebenfalls von Männern beherrschte, IT-Welt. In dieser Welt ist sie immer noch, allerdings nicht ausschließlich: Nebenbei liebt sie Sprachen, Wissenschaft, Kultur, Sport und ihr persönliches Hilfsprojekt in Ecuador. Als inzwischen freiberufliche Apex-Applikations-Entwicklerin kann sie mühelos alle ihre Interessen verfolgen.

die Attraktivität einer Firma durchaus erhöhen.

Was könnte Frauen motivieren, einen Beruf in der IT zu ergreifen?
Hayek: Die Wichtigkeit der oben genannten Eigenschaften für die IT sollte besser herausgestrichen werden. Die komplizierteste Technik hilft nicht viel, wenn der Endanwender mit dem Produkt nicht problemlos umgehen kann. Technisch aufwändige und oft undurchschaubare Anwendungen bleiben meist als „Leichen" zurück, wenn der Entwickler nicht mehr verfügbar ist. Frauen könnten der IT ein „menschlicheres" Image geben, die Benutzerfreundlichkeit generell verbessern und IT-Anwendungen auf das reduzieren, was sie sein sollen: Hilfsmittel, die die Bewältigung von Arbeitsprozessen vereinfachen, beschleunigen und erleichtern.

Welche Eigenschaften sollte eine Frau mitbringen, um sich in der IT-Branche durchzusetzen?
Hayek: Eine gewisse technische Begabung sowie ein Interesse an IT sind natürlich Voraussetzungen. Um sich aber wirklich durchzusetzen, braucht es meiner Meinung nach ein gesundes Selbstbewusstsein, gepaart mit einer guten Portion Humor. Mit der Kombination dieser Voraussetzungen haben wir Frauen es nicht nötig, erfolgreiche Männer „1:1" zu kopieren. Es ist meiner eigenen Erfahrung nach sehr wohl möglich, neue Denkansätze und eine unkompliziertere („weiblichere"?) Sicht der Dinge einzubringen und dabei auch noch akzeptiert zu werden.

Was kann eine Anwendervereinigung wie die DOAG tun, damit mehr Frauen in die IT kommen?
Hayek: Gezielt weibliche Mitglieder anwerben und DOAG-Mitglieder dazu animieren, Kolleginnen zu Veranstaltungen mitzubringen. Frauen interviewen und Frauen explizit als Vortragende einladen. Sofern es nicht dem Gleichheitsgrundsatz widerspricht: Mitgliedsbeiträge und Veranstaltungskosten für Frauen reduzieren, bis eine von der DOAG gewünschte Frauenquote erreicht ist.

Was erwarten Sie von einem IT-Unternehmen wie Oracle?
Hayek: Ich erwarte mir Offenheit und Aufgeschlossenheit gegenüber den oben erwähnten „weiblichen" Eigenschaften – nicht aus Toleranz, nicht dem reinen Wunsch folgend „die Frauenquote zu erhöhen", nicht aus „Mitleid", sondern aus der Erkenntnis heraus, dass eben diese Eigenschaften der IT neuen Schwung und ein neues Image verleihen würden und dass vermutlich viele IT-Prozesse radikal vereinfacht und beschleunigt werden könnten.

Was wünschen Sie sich für die Zukunft?
Hayek: Dass ich sie im Voraus nicht kenne. Dass sie mir weiterhin Überraschungen bietet und dass mein Leben so spannend bleibt, wie es bisher war. Dass ich nie die Bereitschaft verliere, ständig Neues zu lernen und mich geänderten Bedingungen anzupassen. Dass wir Menschen, männlich und weiblich, die IT vernünftig beherrschen. Dass die IT niemals uns Menschen beherrscht.

Wir begrüßen unsere neuen Mitglieder

Persönliche Mitglieder

Ero Ossel
Christian Piasecki
Alexander Galesky
Christophe De Greve
Alain Lacour
Anke Taplick
Lutz Platen
Mirko Schmidt

Firmenmitglieder

Heiko Kropp, BIM GmbH
Thimo Bastuck, Freudenberg IT Information Services SE & Co. KG
Jörg Brast, Prosystems IT GmbH

Christian Trieb
Leiter Database Community

Die Datenbank-Webinare

Seit eineinhalb Jahren führt die Datenbank Community für DOAG-Mitglieder regelmäßig deutschsprachige Webinare durch. Diese finden an jedem zweiten Freitag im Monat um 11 Uhr statt. Die DOAG hatte mit einer Lizenz für 25 Teilnehmer begonnen, wobei die Lizenzgrenze schnell erreicht war. Sie wurde im Jahr 2012 auf 100 Teilnehmer erweitert. Zur Teilnahme reichen die DOAG-Mitgliedschaft, ein Internet-Browser und ein Telefon aus. Die Dauer liegt in der Regel bei 45 Minuten mit anschließender Beantwortung von Fragen.

Die Webinare ergänzen die anderen DOAG-Veranstaltungen dahingehend, dass Themen aus den SIGs aufgegriffen und weiter besprochen werden können. Aber auch die andere Richtung ist möglich, indem ein SIG-Thema im Webinar eingeführt und vorgestellt wird, um dann während der SIG-Veranstaltung vertieft zu werden. Inhaltlich stehen Datenbank-Themen im Vordergrund. Es gab bereits Webinare zu den Themen „Cost Based Optimizer“, „Performance Tuning“ oder „Oracle-Datenbank für Einsteiger“. In diesem Jahr sind „Replikationslösungen im Vergleich“ (Juli), „Applikationen mit RAC hochverfügbar machen (12c-Version)“ (September) und „Proxy Authentication und Remote Log-in ohne sichtbare Passwörter (Wallet)“ (Oktober) bereits fest geplant. Weitere Themenwünsche und Referentenvorschläge nehme ich gerne unter dbc@doag.org entgegen.Weitere Informationen unter www.doag.org/de/events/webinar.html

Tilo Metzger
Leiter SIG Security

„To be or not to be“ – sicher oder nicht sicher

Am Dienstag, 23. April 2013, traf sich die SIG Security in München in einem Hotel im Raum „Shakespeare“. Die Veranstaltung hatte den Schwerpunkt „Sicherheit für Entwickler und Datenbank-Administratoren“. So erschienen nicht nur Interessierte aus der Region, sondern aus dem gesamten Land. Auf dem Programm standen insgesamt fünf Vorträge.

Als Erstes referierte Dr. Bruce Sams (OPTIMAbit GmbH) zum Thema „Sicherheitslücken aufdecken mittels Code Review“. Er schilderte eindrucksvoll, dass rund drei Viertel der Sicherheitslücken durch die Anwendung entstehen und wie man solche Schwachstellen durch Code Analyse und Code Review aufdecken kann. Die Teilnehmer erhielten einen Überblick über Konzepte, Strategien und Werkzeuge für das erfolgreiche Aufdecken von Sicherheitslücken im Programmcode. An Beispielen wurde erklärt, wie falsche Programmierung Türen für SQL Injections oder Cross-Site-Scripting durch potenzielle Angreifer öffnen.

Eine gelungene Überleitung von Shakespeare zur Datenbank-Sicherheit mittels Verschlüsselung schaffte Heinz-Wilhelm Fabry (ORACLE Deutschland B.V. & Co. KG) in seinem Vortrag „Transparent Data Encryption, ORACLE Database Vault und wie man durch Kombination beider Produkte eine neue Qualität bei der Datensicherheit erreichen kann“. Diese Lösung ist nicht nur interessant für Cloud-Anbieter und -Kunden, sondern für alle mit sicherheitssensiblen Daten, die eine völlige physikalische Absicherung benötigen.

Alle Apex-Anwender kamen beim Vortrag „Apex? Aber sicher! Tipps und Tricks für eine sichere Apex-Umgebung“ voll auf ihre Kosten. In seinem Vortrag informierte Carsten Czar-

ski (ORACLE Deutschland B.V. & Co. KG) die Teilnehmer über Architektur, Security-Attribute sowie den Aufbau von Apex-Anwendungen. Er zeigte, wie man seinen Code grundsätzlich organisieren sollte, was beim Thema „Autorisierung" zu beachten ist, wie man sich vor SQL Injections schützt und welche Features Apex dafür mitbringt. Abschließend wurde erläutert, wie man das Apex-Dictionary für „Security Audits" nutzen kann.

Unter der Frage „Wie kritisch ist es wirklich?" erklärte Katja Werner (Opitz Consulting GmbH), welche Patch-Arten es bei Oracle gibt und wie man die Risk-Matrix interpretiert. Da Patches oft mit hohem Aufwand verbunden sind, muss im Vorfeld ermittelt werden: Sind meine Systeme betroffen, wie hoch ist das Risiko eines Angriffs und welche Folgen kann ein Angriff über diese Schwachstellen haben? Zur besseren Argumentation gegenüber dem Management ist der Base Score effektiv einsetzbar.

Zum Abschluss berichtete Stefan Oehrli (Trivadis AG) in seinem Vortrag „A sneak preview on Security with the latest Generation of Database Technology" über seine Erfahrung mit der Beta-Version der neuesten Datenbank-Generation. Die Teilnehmer erhielten einen kurzen Überblick über neueste Trends und Sicherheits-Features.

An dieser Stelle noch einmal herzlichen Dank an alle Referenten, die zum Gelingen dieser Veranstaltung beigetragen haben. Die nächste SIG Security findet voraussichtlich am 11. September 2013 in Frankfurt/Main statt. Falls Sie Anregungen und Vorschläge zu Themen für diese Veranstaltung haben oder selbst mal einen Vortrag halten möchten, senden Sie bitte eine Nachricht an sigsecurity@doag.org.

Stefan Kinnen
Leiter Development Community

Neues aus der Development Community

„Neue Formate müssen her!", das hat die Development Community bei ihrem Frühjahrstreffen in Berlin in einem Fünf-Punkte-Plan deutlich herausgearbeitet. Künftig soll die Zusammenarbeit mit der Oracle ADF Community in Deutschland durch weitere Unterstützung intensiviert werden. Ziel ist es, dass auch in der DOAG das ADF-Know-how zugänglich ist.

Mit noch mehr Schwung kommt das Thema „Apex" in die Praxis. Mit der Ausrichtung auf Mobile Computing in der Version 4.2 und als bewährte Plattform für abgrenzbare Datenbank-Applikationen ist Apex weit verbreitet. Das Expertenseminar im April war ausgebucht, jetzt sollen regionale und auch überregionale Angebote in einer eigenen Apex-Community gebündelt werden. Mit Niels de Bruijn konnte die DOAG einen Themenverantwortlichen gewinnen, der als Apex-Experte bekannt ist. Wir freuen uns über seine Unterstützung.

Punkt drei ist die Gewinnung des Nachwuchses. Gerade im Bereich der Software-Entwicklung ist die Gruppe der Studierenden und Auszubildenden wichtig. Für deren Ansprache werden neue Veranstaltungsformate in Form von „BarCamps" getestet und eingeführt. Ähnlich wie bei der „Unconference" während der Jahreskonferenz sollen hier zwanglos und mit aktiver Beteiligung gerade die jüngeren Talente für die DOAG gewonnen werden.

Im Themenbereich „Java" sind die üblichen Formate nicht immer passend. In Kooperation mit dem Interessenverbund der Java User Groups e.V. (iJUG) soll nun ein kleines Team ein neues Event-Format ausarbeiten, mit dem die DOAG das Thema „Java" im deutschsprachigen Raum besser adressieren kann.

Last but not least möchte die Development Community bereits begonnene fachliche Themen auch ohne klaren Fokus auf eine Technologie weiterführen. Hier wird konkret mit dem Thema „Mobile Computing" gestartet. Auf Basis der bisherigen Events ist ein Arbeitskreis entstanden, der nun eine kontinuierliche Weiterführung der Themen sicherstellt.

Insgesamt sind wir zuversichtlich, dass der Erfahrungsaustausch und Wissenstransfer in der Development Community mit diesen Maßnahmen deutlich steigen wird. Vielen Dank an alle Aktive.

Impressum

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

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

Chefredakteur (ViSdP):
Wolfgang Taschner, redaktion@doag.org

Redaktion:
Fried Saacke, Carmen Al-Youssef, Mylène Diacquenod, Dr. Dietmar Neugebauer, Stefan Kinnen, Tilo Metzger, Christian Trieb

Titel, Gestaltung und Satz:
Alexander Kermas
DOAG Dienstleistungen GmbH

Foto Titel: © burak çakmak / Fotolia.com
Foto S. 18: © Sashkin / Fotolia.com
Foto S. 32: © S.John / Fotolia.com
Foto S. 46: © Julien Eichinger / Fotolia.com
Foto S. 55: © marigold_88 / Fotolia.com

Anzeigen:
Simone Fischer, anzeigen@doag.org
DOAG Dienstleistungen GmbH
Mediadaten und Preise finden Sie unter:
www.doag.org/go/mediadaten

Druck:
Druckerei Rindt GmbH & Co. KG,
www.rindt-druck.de

11.06.2013
Regionaltreffen Hamburg/Nord
Stefan Thielebein
regio-nord@doag.org

12.06.2013
Regionaltreffen Berlin/Brandenburg
Michel Keemers
regio-bb@doag.org

13.06.2013
Regionaltreffen Karlsruhe
Reiner Bünger
regio-karlsruhe@doag.org

13.06.2013
JD Edwards Community Day
Kasi Färcher-Haag
bsc-jde@doag.org

14.06.2013
DOAG Webinar
office@doag.org

18.06.2013
Regionaltreffen NRW (Vorabend DOAG 2013 Development)
Stefan Kinnen, Andreas Stephan
regio-nrw@doag.org

19.06.2013
DOAG 2013 Development
„Die effektive Durchführung von Softwareprojekten" ist das Motto der zweiten Auflage der Community Konferenz für Entwickler, Softwarearchitekten und Projektleiter.
Stefan Kinnen
office@doag.org

20.06.2013
Regionaltreffen Nürnberg/Franken
André Sept, Martin Klier
regio-franken@doag.org

25.06.2013
Regionaltreffen Hannover
Andreas Ellerhoff
regio-hannover@doag.org

26.06.2013
Regionaltreffen München/Südbayern
Andreas Ströbel
regio-muenchen@doag.org

27.06.2013
Regionaltreffen Dresden
Helmut Marten
regio-sachsen@doag.org

02.07.2013
SIG Oracle & SAP
Jörg Hildebrandt
sig-sap@doag.org

09.07.2013
Regionaltreffen Jena/Thüringen
Jörg Hildebrandt
regio-thueringen@doag.org

12.07.2013
DOAG Webinar
office@doag.org

16.07.2013
Regionaltreffen Freiburg
Volker Deringer
regio-freiburg@doag.org

18.07.2013
Regionaltreffen Nürnberg/Franken
André Sept, Martin Klier
regio-franken@doag.org

18.07.2013
Regionaltreffen Stuttgart
Jens-Uwe Petersen
regio-stuttgart@dag.org

23.07.2013
Regionaltreffen München/Südbayern
Andreas Ströbel
regio-muenchen@doag.org

03./04.09.2013
Berliner Expertenseminar:
„Oracle EM12c Monitoring"
mit Bernhard Wesely
Cornel Albert
expertenseminare@doag.org

04.09.2013
Regionaltreffen NRW
Stefan Kinnen, Andreas Stephan
regio-nrw@doag.org

04.09.2013
Regionaltreffen Berlin/Brandenburg
Michel Keemers
regio-bb@doag.org

06./07.09.2013
DOAG Leitungssitzung
office@doag.org

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

Unsere Inserenten

Runs Oracle 10x Faster*

The World's Fastest Database Machine

- Hardware by Sun
- Software by Oracle

*But you have to be willing to spend 50% less on hardware.

ORACLE®

10x faster based on comparing Oracle data warehouses on customer systems vs. Oracle Exadata Database Machines. Potential savings based on total hardware costs. Oracle Database and options licenses not included. Actual results and savings may vary.

oracle.com/exalogic or call 0800 1 81 01 11

BI ist nur so intelligent wie die Köpfe dahinter.

Trivadis ist das führende Unternehmen für IT-Beratung, Systemintegration, Solution-Engineering und IT-Services mit Fokussierung auf Oracle- und Microsoft-Technologien im D-A-CH-Raum. Wir entwickeln Ihre Bussiness-Intelligence-Lösung von der Anforderungsanalyse Ihrer Wertschöpfungskette über die individuelle BI-Strategie bis hin zum Datawarehouse. Das Resultat: effizientere Geschäftsprozesse, schnellere Reaktion auf neue Anforderungen. Sprechen Sie mit uns über Ihre BI-Lösung. www.trivadis.com | info@trivadis.com

BASEL ▪ BERN ▪ LAUSANNE ▪ ZÜRICH ▪ DÜSSELDORF ▪ FRANKFURT A.M. ▪ FREIBURG I.BR.
HAMBURG ▪ MÜNCHEN ▪ STUTTGART ▪ WIEN

trivadis
makes IT easier.