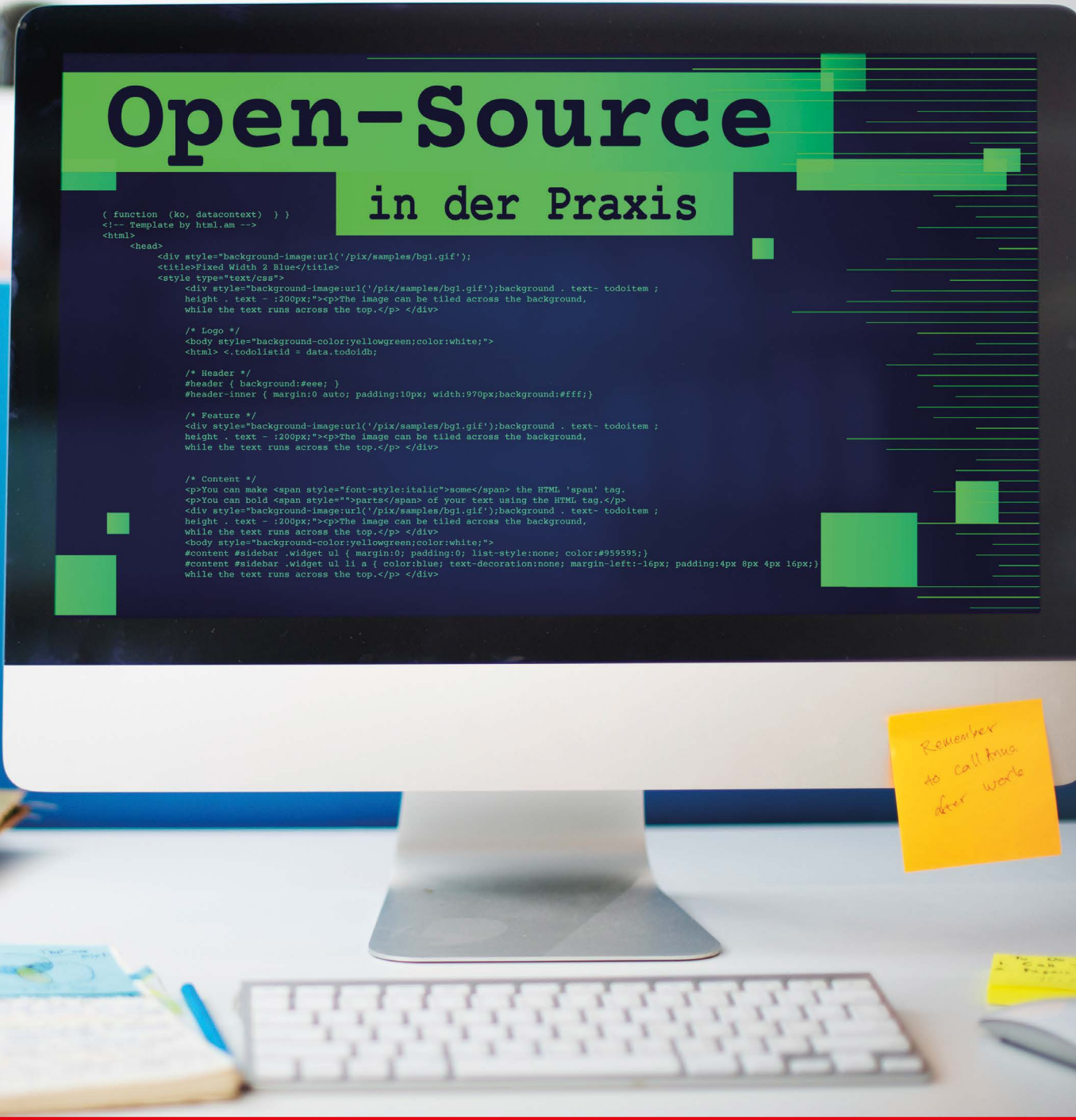


Red Stack

Magazin



Topaktuell

Cloud@Customer mit der
Exadata Cloud Machine

Im Interview

Die Führung der Aoug,
DOAG und SOUG

Unzufriedene Kunden

Die aktuelle DOAG-
Support-Umfrage

APEX
connect
by DOAG

9. – 11.
Mai
in Berlin





Klaus-Michael Hatzinger
AOUG-Präsident

Liebe Mitglieder, liebe Leserinnen und Leser,

der weitreichende Einsatz der Open-Source-Produkte ist heute aus der IT-Welt nicht mehr wegzudenken. Ich erinnere mich noch an die Zeit vor etwa fünfzehn Jahren, da waren die IT-Verantwortlichen in den Unternehmen deren produktivem Einsatz noch skeptisch gegenüber eingestellt. Linux als Betriebssystem auf dem Heim-PC war okay, aber sollte man zum Betrieb von hochkritischen Systemen einer von einer Community entwickelten Open-Source-Software vertrauen?

Mittlerweile hat sich Open Source in unzähligen Bereichen – angefangen vom Betriebssystem über Datenbank- und Applikationsserver bis hin zu fertigen Anwendungen wie CRM- und ERP-Lösungen – als kostengünstige Alternative etabliert und macht mit der OpenStack-Initiative auch vor dem Cloud-Zeitalter nicht halt. Auch Oracle fährt eine klare Open-Source-Strategie und setzt unter dem Motto „Choice, Flexibility and Lower Cost“ auf zahlreiche Open-Source-Produkte im eigenen Portfolio.

In dieser Ausgabe widmen wir uns voll und ganz dieser Thematik und geben Ihnen neben spannenden Fachartikeln, etwa zu MySQL und Apache Cassandra, auch einen tiefgehenden Einblick in das Grundverständnis von Open Source und OpenStack.

Ich wünsche Ihnen viel Vergnügen beim Lesen und ein erfolgreiches Jahr mit vielleicht dem einen oder anderen Open-Source-Projekt.

Ihr *Michael Hatzinger*



MUNIQSOFT

Consulting

Hochverfügbarkeit mit IQ

Sicherheit vor teuren Ausfallzeiten: Mit dem richtigen Konzept sind Ihre Daten und Server vor Systemausfällen optimal geschützt.

Nutzen Sie die Erfahrung von Muniqsoft

ORACLE® Gold Partner

Specialized
Oracle Database



Jetzt Beratungstermin vereinbaren:
+49 (0)89 6228 6789-21

www.muniqsoft.de



Der Quelltext der Open-Source-Software kann eingesehen werden



Die beiden weltweit wichtigsten Datenbank-Systeme im Vergleich



Oracle führt unter dem Oberbegriff „Software in Silicon“ eine Reihe neuer Technologien ein

- 3 Editorial
- 5 Timeline
- 8 „Das Beste für den Oracle-Anwender ...“
Interview mit Klaus-Michael Hatzinger, Stefan Kinnen und Peter Gübeli

Open Source

- 12 Open Source
Bernhard Halbetel
- 14 Apache Cassandra, wieso brauche ich das?
Jan Ott
- 19 MySQL vs. Oracle-Datenbank
Oliver Sennhauser und Matthias Jung
- 23 MySQL Branches und Forks
Oliver Sennhauser
- 26 Oracle sagt: Fachabteilungen und IT-Teams müssen besser zusammenarbeiten
- 27 MySQL-Auswahl gegen die Distribution
Jörg Brüche
- 33 MySQL in der Oracle Public Cloud – den Durchblick behalten
Mario Beck und Carsten Thalheimer
- 39 OPatch ferngesteuert: Automatisiertes Patchen mit Open Source Puppet
Raphael Daum
- 42 Entspannt virtualisieren mit dem offenen Oracle-Multipurpose-Hypervisor
Nico Henglmüller und Dr. Thomas Petrik
- 46 OpenDAX-APIs: Software in Silizium als Turbo für eigene Anwendungen
Franz Haberhauer
- 50 Datenbank-zentrische Anwendungen mit Spring Boot und jOOQ
Michael Simons

Aktuell

- 58 Cloud@Customer mit der Exadata Cloud Machine
Frank Schneede
- 63 Akzeptanz von „My Oracle Support“ steigt, allgemeine Zufriedenheit sinkt
Christian Trieb
- 65 Die DOAG Support-Umfrage
Mario Kowalski
- 72 Oracle-Forms-Trends von der DOAG 2016 Konferenz und wohin die Reise geht
Frank Hoffmann

Datenbank

- 68 In-Memory-Datenbanken: Perfekte Addon-Technologie für schnellere Analysen
Mathias Golombek

Intern

- 73 Termine
- 74 Neue Mitglieder
- 74 Inserenten
- 74 Impressum

Timeline

9. November 2016

In Brugg-Windisch in der Schweiz findet der Oracle Cloud Day statt. Die SOUG war mit einem eigenen, gut besuchten Stand vertreten. Ein Wettbewerb bringt dem Sieger ein iPhone und der SOUG eine Statistik über die Teilnehmer, die sich trotz des Themas „Cloud“ mehrheitlich zur Infrastruktur und Datenbank hingezogen outeten.

14. November 2016

Am Vortag der DOAG 2016 Konferenz + Ausstellung treffen sich in Nürnberg 31 Teilnehmer zum EOUC Leader Forum. Die Vertreter aus 23 User Groups sowie drei Mitarbeiter von Oracle diskutieren den Auftritt der User Groups auf der vergangenen Oracle Open-World (OOW) und die zukünftige Zusammenarbeit in Europa. Ein weiteres Thema ist die verstärkte Vernetzung bei der Planung von Events. Gerade benachbarte User Groups sollen diese in Zukunft besser abstimmen. Damit können auch die Speaker ihre Termine leichter planen. Um bessere Vernetzung geht es auch Ami Aharonovich, President der Israel Oracle User Group (iIOUG), der zum Abschluss die Möglichkeit einer virtuellen Konferenz vorschlägt. Man einigt sich darauf, dass in einem ersten Schritt auf EMEA-Ebene Webinare basierend auf den Vorträgen der OOW angeboten werden. Auch Oracle schenkt den User Groups Beachtung: Die anlässlich des Oracle Cloud Day anwesende Cloud Focus Group lädt die User Group Leader zu einem Austausch ein, um das weltweite Feedback der Anwender direkt an den CEO Mark Hurd weiterzugeben. Neben Berichten zu den ersten Aktivitäten zum Thema „Cloud“ in den User Groups wird deutliche Kritik an der derzeitigen Oracle-Strategie „Cloud first“ geäußert. Diese führe zu einem Vertrauensverlust bezüglich künftiger Geschäftsbeziehungen zu Oracle.



Die Teilnehmer des EOUC Leader Forums diskutieren die zukünftige Zusammenarbeit der User Groups in Europa

15. November 2016

Die 29. Ausgabe der Jahreskonferenz der DOAG öffnet im Nürnberg Convention Center ihre Pforten. Die gemeinsam mit der AOUG und der SOUG vorbereitete Veranstaltung erwartet in den drei Tagen insgesamt rund 2.500 Datenbank-Administratoren, Entwickler und weitere IT-Experten. Die mehr als 470 Vorträge von renom-

mierten Referenten und internationalen Top-Speakern in zwanzig parallelen Tracks bieten den Besuchern die einmalige Gelegenheit in Deutschland, sich an nur einem Ort über die aktuellen Technologien und Best Practices der Branche zu informieren. Zahlreiche praktische Anwender- und Erfahrungsberichte decken insbesondere die Trendthemen „Mobile“, „Cloud“ und „Big Data“ umfassend ab. Neben den klassischen Vorträgen bieten zahlreiche interaktive Formate wie Unconference, Community-Aktivitäten oder Panel-Diskussionen unzählige Networking-Möglichkeiten. Auch bei der begleitenden Ausstellung mit rund 60 Partnern haben die Teilnehmer viele Gelegenheiten, um mit Referenten, Unternehmen und anderen Besuchern ins Gespräch zu kommen. Beim anschließenden Schulungstag am 18. November können die Teilnehmer in elf Seminaren ihr Wissen vertiefen. Zu den Teilnehmern gehören auch die rund 400 Besucher des integrierten Oracle Cloud Day, der nach dem erfolgreichen Debüt im letzten Jahr nun zum zweiten Mal parallel zur DOAG 2016 Konferenz + Ausstellung stattfindet.



Die DOAG-Jahreskonferenz informiert die Oracle-Anwender über wichtige Themen wie „Cloud“, „Security“ und „Support“

15. November 2016

Spielt PL/SQL eine Schlüsselrolle in der IT-Landschaft, kann der Mangel an Mitarbeitern sogar dazu führen, dass ein Umstieg auf eine andere technologische Basis in Erwägung gezogen wird? Diese Frage stellt Christian Schwitalla, Leiter DOAG PL/SQL Community, im Rahmen der DOAG 2016 Konferenz auf der Unconference-Session „PL/SQL Young Generation“. Anscheinend ist das Problem kein rein deutsches Phänomen, denn es kommen auch viele Teilnehmer aus den USA, aus England, aus den Niederlanden und aus der Schweiz. Die lebhafte Diskussion ergibt schnell eine einhellige Meinung, dass hier tatsächlich ein Problem vorliegt: Die Kenntnisse bezüglich Datenbank-Programmierung in PL/SQL sind bei jungen Programmierern nicht weit verbreitet und das Berufsbild des Datenbank-Programmierers ist weitgehend unbekannt. Die Teilnehmer bestätigen, was die Aktiven der DOAG seit Jahren hören: Es wird immer schwieriger, geeignete Mitarbeiter mit entsprechenden Kenntnissen zu finden. Viele Unternehmen stellen sich deshalb an Hochschulen vor oder bieten Studierenden Praktika und Diplomarbeiten an. Alle Teilnehmer sind sich darin einig, dass den Schulen bei der Verbreitung der PL/SQL-Kenntnisse eine Schlüsselrolle zufällt. Studierende sollen tiefgehende Skills über den Umgang mit relationalen Datenbanken erwerben, die weit über die SQL-Grundkenntnisse hinausgehen. Dafür soll optimalerweise Oracle und PL/SQL genutzt werden. Dabei sollen Lehrende die Tatsache, dass PL/SQL herstellereigen ist, nicht ne-

gativ betrachten. Auch scheinbar banale Maßnahmen können große Wirkung zeigen, so sollten im Oracle-Beispiel-Schema statt der angestaubten „EMP“- und „DEPT“-Tabellen aktuellere Beispiele vorkommen, etwa „TWEETS“ und „POSTS“.

15. November 2016

Rolf Scheuch, Leiter der DOAG Business Intelligence Community, präsentiert im Rahmen der DOAG 2016 Konferenz die Position der DOAG zur Oracle-Cloud-Strategie. Demnach begrüßt die DOAG das Commitment von Oracle in Richtung „Cloud“. Allerdings führt der Mangel an klaren Aussagen seitens Oracle häufig zu Verunsicherungen bei den Kunden. Ein aktuelles Beispiel ist der fehlende Veröffentlichungstermin der On-Premise-Version der Datenbank. Die Vielzahl der angebotenen Cloud-Lösungen dürfte aufgrund der unzureichenden Transparenz und Architekturvorgaben zu einer weiteren Verunsicherung bei den interessierten Kunden führen; das Oracle-PaaS-Angebot allein umfasst rund zwanzig Sub-Angebote. Die DOAG empfiehlt daher bei der Evaluierung der Oracle-Cloud-Angebote die dringende Prüfung von Preisgestaltung, Datenschutz-Compliance sowie (Un-) Abhängigkeit und nennt die drei wichtigsten Punkte. Erstens: Welche Preisgestaltung ist für den Kunden sinnvoll? Pay-by-Use (metered) oder fixe Volumen (un-metered) bieten Vor- und Nachteile, die es individuell zu prüfen gilt. Außerdem sollten die preislichen Angebote für die Migration bestehender Lizenzen von On-Premise in die Cloud evaluiert werden. Zweitens: Wo werden die Daten gespeichert? Deutsche Firmen sollten sich vergewissern, dass der Datenschutz und die Service-Level-Agreements der angebotenen Cloud-Lösungen den regulatorischen Anforderungen der EU, Deutschlands und des eigenen Hauses entsprechen. Wenn möglich, sollte man die Anforderungen vertraglich festhalten. Drittens: Wie kann Investitionssicherheit bezüglich Daten, Programmen und Lizenzen erreicht werden? Es sollte sichergestellt sein, dass bestehende Optionen in die Cloud migrierbar sind, aber auch der Wechsel des Cloud-Providers oder der Ausstieg aus dem Cloud-Angebot kostengünstig möglich sind.

15. November 2016

Christian Trieb, Leiter des DOAG Competence Centers Support präsentiert die Ergebnisse der diesjährigen Support-Umfrage. Das Serviceportal „My Oracle Support“ schnitt dabei deutlich besser ab als im Jahr 2014: 48 Prozent der Teilnehmer vergaben die Note „gut“. Damit verbesserte sich der „My Oracle Support“ um 7 Prozentpunkte. Die allgemeine Zufriedenheit ist dagegen gesunken. Weiterhin viel Luft nach oben bietet der Bekanntheitsgrad einzelner Support-Tools, wie etwa der „Oracle Proactive Support“. Das Tool ist bei 40 Prozent der Befragten unbekannt und nur sechs Prozent nutzen es. Der angekündigte Oracle-Vertreter Mario Kowalski, Customer Support Country Leader bei Oracle Deutschland, fehlt bei der Präsentation der Ergebnisse krankheitsbedingt. Christian Trieb richtet die sich aus der anschließenden Diskussion ergebenden Fragen an Oracle und erhält schriftlich eine Antwort darauf (siehe Seite 65).

15. November 2016

Am ersten Abend der DOAG Konferenz lädt die AOUG alle österreichischen Teilnehmer in die Altstadt von Nürnberg zum traditionellen Österreicher Abend ein. Nach dem Sektempfang im

Untergeschoss des Kongresszentrums geht es mit dem Bus zum Networking in das Lokal „Alte Küch'n & im Keller“.

16. November 2016

Comedy-Hacker und Security-Experte Tobias Schrödel läutet die traditionelle Abendveranstaltung ein. In seiner Keynote „Hacking für Manager“ klärt er auf unterhaltsame Weise über Sicherheitslücken und Datensicherheit in Unternehmen auf. Kaum jemand verlässt vorzeitig den Saal, die Zuhörer nehmen interessante Aspekte für ihre Arbeit mit. Die mitgebrachten 100 Exemplare seines neuesten Buchs, die Tobias Schrödel vor Ort signiert, sind im Nu ausverkauft.



Tobias Schrödel demonstriert anhand vieler praktischer Beispiele die Sicherheitslücken aktueller Webseiten

16. November 2016

Kurz bevor die Band bei der Abendveranstaltung aufspielt, wird Günther Stürner, Oracle Vice President Server Technologies und Sales Consulting im Vorruhestand, zum Ehrenmitglied der DOAG ernannt. Während seiner gesamten Zeit bei Oracle hat er die DOAG stets auf das Engste begleitet und sich in besonderem Maße für den Verein engagiert. So organisiert er im Jahr 1985 das allererste Meeting der DOAG in Triberg im Schwarzwald mit: „Wir trafen uns zu acht im Nebenzimmer eines Gasthauses. Schon von Anfang an war der Verein ein Kristallisationspunkt für die Nutzer der Oracle-Produkte.“



Der DOAG-Vorstandsvorsitzende Stefan Kinnen (links) überreicht Günther Stürner die Ehrenurkunde

18. November 2016

Die DOAG 2016 Konferenz, größte Oracle-Konferenz in Europa, geht mit dem Schulungstag zu Ende. Fried Saacke, Vorstandsmitglied und Leiter der DOAG-Geschäftsstelle, zieht ein positives Fazit: „Die Signalwirkung der Veranstaltung reicht bis in die Oracle-Headquarters. So hat sich Thomas Kurian, Oracle President für das Product Development, bereits kritisch geäußert, dass der Konkurrent Rimini Street als Aussteller in Nürnberg vertreten war.“

23. November 2016

Der Vorstand der AOUG trifft sich in Wien zur letzten Vorstandssitzung im Jahr 2016. Es werden zunächst ein Jahresplan für das Jahr 2017 erstellt und die Termine für die Anwenderkonferenz 2017 sowie für alle Vorstandssitzungen fixiert. Anschließend folgt eine Diskussion über potenzielle Keynotespeaker der AOUG Anwenderkonferenz.

24. November 2016

Das Organisationsteam der DOAG-Geschäftsstelle hält in Berlin eine Feedback-Runde zur DOAG 2016 Konferenz + Ausstellung. Alle eingegangenen Kommentare und Vorschläge werden bewertet und daraus bereits die Weichen für eine erfolgreiche Jahreskonferenz 2017 gestellt.

1. Dezember 2016

Der DOAG-Vorstand trifft sich zu seiner letzten Sitzung im Jahr 2016 in Berlin. Im Vordergrund stehen die Planung für das Jahr 2017 sowie die anstehenden Neuwahlen der Delegierten und Vorstände. Ein weiterer Punkt ist die DOAG BI Konferenz, die zukünftig mit der Oracle-Data-Warehouse-Veranstaltung zusammengelegt wird, um den Teilnehmern das Beste aus beiden Events bieten zu können.

1. Dezember 2016

Der DOAG-Vorstand trifft sich nach seiner Sitzung mit den Mitarbeiterinnen und Mitarbeitern der DOAG-Geschäftsstelle zum Jahresabschluss. Fried Saacke, DOAG-Vorstand und Geschäftsführer, dankt allen für die tolle Arbeit im Jahr 2016.

1. Dezember 2016

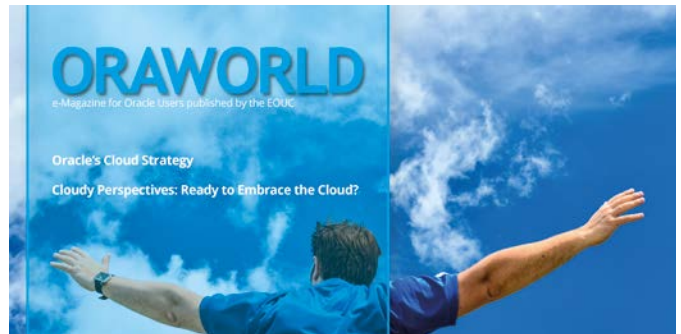
Die Vorstandssitzung der SOUG seht unter dem Zeichen der Vorbereitung für das Jahr 2017 und dem stetigen Thema, strategisch die optimalen Services für die Mitglieder zu definieren. Die Agenda 2017 sieht neben den erfolgreichen SOUG-Days wieder einen Training-Day sowie einen Jubiläums-Event vor.

7. Dezember 2016

Die traditionellen AOUG-Adventtreffs finden in Wien und Graz statt – zum Jahresabschluss noch einmal ein geselliges, kulinarisches Zusammentreffen der österreichischen Oracle-Community.

19. Dezember 2016

Die dritte Ausgabe des EOUC E-Magazine ORAWORLD erscheint. Themen sind „Debugging in Apex“, „Oracle-Datenbank 12.2“, „Herkunft des MySQL-Delfins“ und „Oracle Open World 2016“. Der unter der Regie der DOAG von den European Oracle User Groups herausgegebene, englischsprachige Newsletter wird von Tausenden Anwendern in Europa und in den USA gelesen.



Das E-Magazine ORAWORLD, Ausgabe 3/2016

23. Dezember 2016

Die DOAG-Geschäftsstelle beendet das Jahr 2016 und schließt das Berliner Büro über die Feiertage. Fried Saacke, DOAG-Vorstand und Geschäftsführer, blickt mit seinem Team auf ein sehr erfolgreiches Jahr zurück. Schlüsselprojekte waren die Weiterentwicklung der DOAG-Webseiten sowie die organisatorische Umstrukturierung, um die großen Veranstaltungen abzusichern. Nicht zuletzt war auch der Kinder-Boom in der Geschäftsstelle eine große Herausforderung. Weil das Know-how auf viele Schultern verteilt ist, konnte diese Situation erfolgreich gemeistert werden.

27. Dezember 2016

Fried Saacke, DOAG-Vorstand und Geschäftsführer, und Wolfgang Taschner, Chefredakteur der DOAG-Zeitschriften, treffen sich zur redaktionellen Planung für das Jahr 2017. Wichtigster Punkt ist die Erhöhung der Java aktuell zunächst auf fünf Ausgaben pro Jahr. Ab dem Jahr 2018 wird die Zeitschrift dann auf eine zweimonatliche Erscheinungsweise umgestellt.

1. Januar 2017

Die DOAG wünscht allen Mitgliedern und Interessenten ein gesundes und erfolgreiches neues Jahr!

9. Januar 2017

Das Organisationsteam der JavaLand 2017 legt in einer Abstimmungs-Telko die letzten Details für die Veranstaltung fest. Schon jetzt zeichnet sich bei den eingehenden Anmeldezahlen ein erneuter Besucherrekord ab.



Die Java-Community trifft sich vom 28. bis 30. März 2017 im Phantasialand in Brühl



Klaus-Michael Hatzinger, Peter Gübeli und Stefan Kinnen (von links nach rechts)

„Das Beste für den Oracle-Anwender ...“

Anwendergruppen sind eine sehr wichtige Säule des Oracle-Universums. Wolfgang Taschner, Chefredakteur des Red Stack Magazin, sprach darüber mit Ing. Klaus-Michael Hatzinger, Präsident der Austrian Oracle User Group (AOUG), Stefan Kinnen, Vorstandsvorsitzender der DOAG Deutsche ORACLE-Anwendergruppe, und Peter Gübeli, Präsident der Swiss Oracle User Group (SOUG).

Wie sind die AOUG, DOAG und SOUG aufgebaut?

Klaus-Michael Hatzinger: Die AOUG gibt es bereits seit achtundzwanzig Jahren. Wir sind mit unseren rund hundert-siebzig Firmenmitgliedern eine eher kleine User Group, freuen uns aber jährlich über laufenden Zuwachs. Insgesamt haben wir ungefähr zweieinhalbtausend registrierte Personen, die sich aus aktiven Mitgliedern und Interessenten zusammensetzen. Im Vorstand gibt es einen Präsidenten, einen Vizepräsidenten und drei Vorstände mit speziellen Aufgaben und Verantwortlichkeiten. Jedes Jahr im Juni

veranstalten wir eine Anwenderkonferenz mit rund einhundert-zwanzig Teilnehmern. Darüber hinaus bieten wir Vormittagsveranstaltungen in Form eines sogenannten „Technischen Frühstücks“ zu bestimmten Themen an, was sehr gut angenommen wird. Neu sind regionale Foren, also Veranstaltungen, die nicht in Wien stattfinden. Das ergibt insgesamt etwa dreißig Events im Jahr.

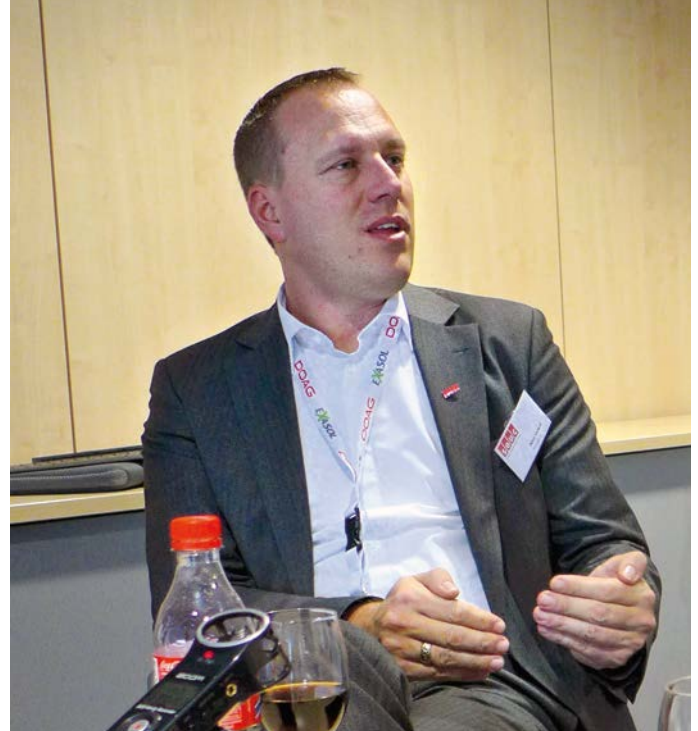
Stefan Kinnen: Die DOAG ist über neunundzwanzig Jahre hinweg permanent gewachsen und hat Dimensionen erreicht, die



Ing. Klaus-Michael Hatzinger

seit Langem ehrenamtlich nicht mehr zu bewältigen sind. In unserem Berliner Büro sind fast zwanzig festangestellte und freiberufliche Mitarbeiterinnen und Mitarbeiter mit den Veranstaltungen und Medien der DOAG beschäftigt. Die Mitgliedschaft wurde vor einigen Jahren auf benannte Personen umgestellt, momentan sind wir bei knapp achttausend angelangt. Hinzu kommen etwa nochmal so viele Interessenten. Es gibt rund einhundert Veranstaltungen im Jahr, viele davon auf regionaler Basis. Die jährliche Anwenderkonferenz wird von rund zweitausendfünfhundert Teilnehmern besucht. Die DOAG teilt sich in sechs Communities, die alle Fachbereiche von der Datenbank über BI, Middleware und Development bis hin zu den Business Applications abdecken. Hinzu kommt eine gerade erst gegründete Community für die „Next Generation“, um unseren Nachwuchs zu fördern.

Peter Gübeli: Die SOUG feiert nächstes Jahr ebenfalls das dreißigste Jubiläum. Wir haben knapp vierhundert Mitgliedschaften, wobei wir gerade am Überlegen sind, wie wir den administrativen Aufwand dafür verringern können und vor allem für Firmen ein attraktiveres Mitgliedschaftsmodell schaffen. Unsere Hauptaktivitäten sind die SOUG Days, die wir als Halbtagsveranstaltungen viermal im Jahr durchführen. Es gibt meistens drei parallele Streams mit jeweils vier Vorträgen, zu denen fünfzig bis sechzig Teilnehmer kommen. Hinzu kommt „SOUG bi de Lüt“, wo wir Kunden oder Partner besuchen, um deren Arbeitsweise mit den Oracle-Produkten kennenzulernen. Darüber hinaus bieten wir Veranstaltungen in der Romandie für unsere französischsprachigen Kollegen an. Die ganztägigen SOUG Training Days zu einem bestimmten Thema haben sehr großen Anklang gefunden. Im Vorstand sind wir alle ehrenamtlich tätig, für die Organisation steht uns eine festangestellte Halbtagskraft zur Verfügung.



Peter Gübeli

Wo siehst du die Stärke deiner User Group?

Peter Gübeli: Wir besitzen eine gute und stabile Basis. Die meisten Mitglieder sind schon sehr lange dabei. Jede Stärke hat auch einen Nachteil. So tun wir uns schwer, neue Themen zu etablieren. Bei der Kostenstruktur haben wir ein schlankes Modell gefunden.

Klaus-Michael Hatzinger: Wir sind unabhängig und sehr anwendernah orientiert und haben beispielsweise im letzten Jahr verschiedene Lizenzierungsthemen aufgegriffen. Generell möchten wir uns mit allen typischen Themen der Oracle-Anwender auseinandersetzen, auch wenn so manches davon bei unserer lokalen Oracle-Organisation nicht so gerne gesehen wird. Die meisten Oracle-Anwender haben nicht nur Oracle-Produkte im Einsatz, sondern ein Zusammenspiel von Oracle und anderen Herstellern.

Stefan Kinnen: Eine unserer großen Stärken liegt darin, dass wir trotz des immensen Wachstums die Unabhängigkeit von Oracle behalten konnten. Wir entscheiden die Themen unserer Veranstaltungen selbst und sprechen die Probleme der Anwender auch öffentlich an, selbst wenn wir Oracle damit keine große Freude bereiten. Hinzu kommen die großen Ressourcen unseres Berliner Büros, was uns Möglichkeiten in vielerlei Hinsicht öffnet.

Am Vortag der DOAG 2016 Konferenz haben sich 25 Anwendergruppen aus Europa, Afrika, dem Mittleren Osten und den USA in Nürnberg getroffen. Wie habt ihr dieses Treffen wahrgenommen und welchen Stellenwert räumt ihr der internationalen Zusammenarbeit der User Groups ein?

Klaus-Michael Hatzinger: Ich finde es sehr interessant zu erfahren, wie andere Anwendergruppen arbeiten und ihre Programme gestalten, um vielleicht die eine oder andere Idee für die eigene User Group mitnehmen und umsetzen zu können.



Stefan Kinnen

Peter Gübeli: Meine Meinung dazu ist geteilt. Den Austausch untereinander finde ich sehr gut, von der angestrebten Zusammenarbeit im EMEA-Umfeld halte ich jedoch nicht viel. Für uns ist primär die Kooperation im deutschsprachigen Raum ausreichend. Zeitpunkt und Location des Treffens waren hingegen gut getroffen. Das sollte man unbedingt so weiter fortführen.

Stefan Kinnen: Nach unserer Erfahrung können wir auf Länderebene wenig bei Oracle erreichen. Unsere Motivation für eine internationale Zusammenarbeit der Anwendergruppen liegt also darin, ein starkes Signal an Oracle zu senden, was wir gemeinsam geändert haben möchten, um unseren Kritiken und Verbesserungsvorschlägen mehr Gehör zu verschaffen. Wenn alle Anwendergruppen aus ganz Europa zusammen ein bestimmtes Anliegen vortragen, wird sich sicher leichter etwas bewegen lassen.

Wie schätzt du die aktuelle Palette der Oracle-Produkte ein?

Klaus-Michael Hatzinger: Das Angebot ist sehr breit und unübersichtlich geworden. Kerntechnologie ist für mich nach wie vor die Datenbank. Vor dem Hintergrund der Cloud habe ich den Eindruck, dass die Oracle-Mitarbeiter enorm unter Druck stehen. Für uns als AOUG ist es sehr schwierig, mit den raschen Produkt-Änderungen Schritt zu halten und die Anwender entsprechend zu unterstützen.

Stefan Kinnen: Die DOAG hat immer noch das Ziel, die Interessen der Anwender aller Oracle-Produkte zu vertreten. Das ist heute schwieriger denn je und es gibt mittlerweile auch Bereiche, die wir nicht abdecken, insbesondere im Applications-Umfeld. Es ist jedoch offensichtlich, dass sich Oracle von den Datenbank-Themen lösen



Zur Person: Ing. Klaus-Michael Hatzinger

Klaus-Michael Hatzinger ist seit dem Jahr 2000 bei der AOUG und seit Anfang 2014 Präsident der ehrenamtlichen User Group. Beruflich ist er einer der Geschäftsführer und Eigentümer der Firma DBConcepts GmbH, ein in Österreich etablierter führender Oracle Platinum Partner mit Fokus auf Oracle-Infrastrukturen und Managed Services. Über die Jahre hinweg hat er sich zudem zum Oracle-Lizenz-Experten entwickelt, unterstützt laufend Firmen bei der Oracle-Lizenzierung und hält Lizenzierungsseminare im deutschsprachigen Raum ab. Privat geht Klaus-Michael Hatzinger gerne wandern und verbringt seine Urlaube am liebsten in Griechenland am Strand oder auf der Motoryacht in Kroatien.

will. Früher hat der Hersteller bei jedem neuen Datenbank-Release Touren durch das Land veranstaltet, um das Produkt bekannt zu machen. Heute ist die neueste Datenbank nur in der Cloud verfügbar und Oracle ist nicht einmal in der Lage, einen Termin zu nennen, wann das Produkt On-Premise verfügbar sein wird.

Peter Gübeli: Oracle hat zu seinem heutigen Ruf sehr viel selbst beigetragen. Es wurden viele Produkte auf den Markt geworfen und wieder eingestellt, sodass es den Anwendern schwerfällt, alles zu glauben, was das Oracle-Marketing verbreitet. Die Datenbank als etablierte Technologie ist für mich nach wie vor eine verlässliche Säule, auf die man auch in Zukunft setzen kann. Bei den neuen Produkten hingegen fehlt die Kontinuität. Wir haben oft versucht, in der SOUG neue Bereiche neben den Coretech-Themen zu etablieren. Das ist nur teilweise gelungen, weil die Anwender Oracle in vielen Bereichen nicht als strategischen Partner sehen, und bei der Cloud ist Oracle zu spät dran. Andere Cloud-Anbieter sind in den letzten Jahren stark auf die europäischen und insbesondere auch auf die Schweizer Bedürfnisse eingegangen. Das spüre ich bei Oracle noch nicht genügend.



Zur Person: Stefan Kinnen

Stefan Kinnen ist seit Mai 2016 Vorstandsvorsitzender der DOAG Deutsche Oracle-Anwendergruppe e.V. Zuvor war er stellvertretender Vorsitzender und Vorstand Finanzen und ist seit dem Jahr 2005 – mit kurzer Unterbrechung – Vorstandsmitglied, unter anderem als Leiter der Development Community. Außerdem war er Leiter der Regionalgruppe Nordrhein Westfalen. Als Vice President Business Development beim Oracle-Platin-Partner Apps Associates schafft er mit seinem Team Lösungen in den Bereichen „Business Software“, „Custom Development“, „Business Intelligence“ und „Managed Services“. In seiner Freizeit ist Stefan Kinnen gerne sportlich aktiv – zu seinen liebsten Sportarten gehören Golf, Wandern und Radeln. Außerdem interessiert er sich sehr für Musik.

Wie ist dein Kontakt zum Leiter der Oracle-Niederlassung in deinem Land?

Peter Gübeli: Es gibt eine gute Zusammenarbeit mit der lokalen Niederlassung, beispielsweise waren wir am Oracle Cloud Day in der Schweiz auch vor Ort und haben den Event mitgestaltet.

Klaus-Michael Hatzinger: Mein persönlicher Kontakt zum Niederlassungsleiter und vielen anderen Mitarbeitern bei Oracle ist sehr gut, bei der Zusammenarbeit zwischen Oracle und der User Group sehe ich jedoch noch einiges Potenzial. Ich denke, nachdem wir ja immer wieder mal sehr heikle Themen aufgreifen, hält sich Oracle bei der Unterstützung zurück.

Stefan Kinnen: Die Zusammenarbeit mit dem Hersteller funktioniert unterschiedlich gut, in jeder unserer Communities ist zumindest ein Oracle-Mitarbeiter vertreten. Lediglich bei den Business Applications gibt es keinen Kontakt mehr. Mit dem Leiter von Oracle Deutschland sind wir im sehr guten Dialog.



Zur Person: Peter Gübeli

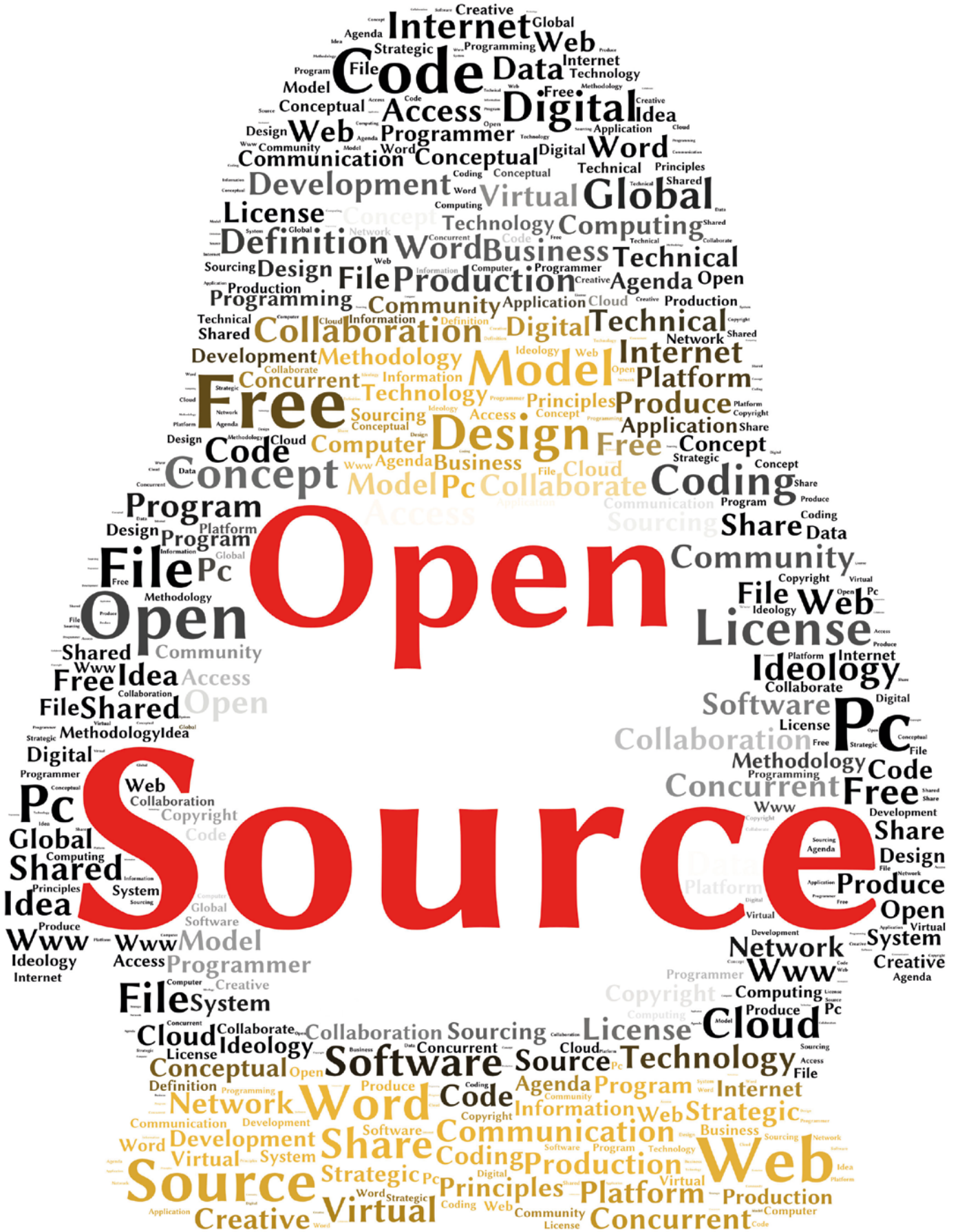
Peter Gübeli führt seit dem Jahr 2014 die SOUG als Präsident. Beruflich ist er Leiter des Procurements und Partnermanagements bei der Helsana Versicherungen AG, dem Schweizer Marktführer bei den Krankenversicherungen. Zuvor war er schon in diversen Führungsfunktionen innerhalb der Helsana-IT, aber auch in anderen Branchen in den Bereichen „Projekte“, „Entwicklung“ und „Beratung“ anzutreffen. Sein großes Hobby ist die Seefahrt, weshalb er regelmäßig mit dem Boot auf dem Rhein oder auf dem Meer anzutreffen ist.

Wie sollte sich deine Anwendergruppe weiterentwickeln?

Stefan Kinnen: Das ist wirklich spannend und wir haben das im Vorstand auch schon diskutiert. Die Frage ist, wie der Anwender künftig aussieht. Wenn man die Anwendergruppen in den USA betrachtet, steigt dort der Anteil der Anwender aus den Fachbereichen stark an, während die Zahl der IT-Mitarbeiter weniger wird. Das heißt, dass sich Personalverantwortliche, Marketingleiter oder Vertriebsmitarbeiter über Erfahrungen mit ihren SaaS-Lösungen austauschen möchten, statt über Datenbanken, Apex oder PL/SQL zu reden. Das Verhältnis ist in Europa sicher noch deutlich umgekehrt, aber ich glaube schon, dass wir uns in diese Richtung bewegen werden. Insofern wird sich auch unsere Arbeit entsprechend ändern.

Peter Gübeli: Das sehe ich genauso. Die Unternehmen kaufen fertige Anwendungen als Service ein. Mit dieser Veränderung der Fertigungstiefe sind Mitarbeiter mit tiefgehenden technischen Kenntnissen nicht mehr gleich gefragt. Das wird auch unsere Zielgruppe massiv verändern.

Klaus-Michael Hatzinger: Erfahrungsaustausch wird weiterhin eine der wichtigsten Säulen einer Anwendergruppe sein, wenn auch mit einem etwas anderen Fokus.



Die Open-Source-Initiative (OSI) geht auf das Jahr 1998 zurück und wurde durch die Unternehmen IBM, HP und Debian gefördert. Eine Liste der OSI-zertifizierten Software ist unter „<https://opensource.org/licenses>“ einzusehen.

Open Source hat gemeinhin den Ruf, als „freie Software“ keinerlei Lizenzpflichten zu unterliegen. Open Source bedeutet jedoch nur, dass der Quelltext der betreffenden Software von Dritten eingesehen werden kann. Open-Source-Software unterliegt immer gewissen Nutzungsrechten und darf gemeinhin nicht vorbehaltlos verwertet, vervielfältigt und bearbeitet werden. Die Nutzungsrechte der Open-Source-Software sind in Lizenzbedingungen abgebildet, die unter der Verwaltung der OSI stehen, oder in einer speziellen, von der Distribution der Open-Source-Software zur Verfügung gestellten Form der Nutzungsbedingung (EULA oder Ähnliches) festgelegt. Dies ist der wesentliche Unterschied zu Public-Domain-Software, die dem Nutzer uneingeschränkt das Recht der Verbreitung, Veränderung und Vervielfältigung einräumt.

Open-Source-Software wird von Programmierern geschrieben, die diese oftmals frei zur privaten Nutzung oder Unternehmen gegen Entgelt zur Verfügung stellen. Meistens wird der Code jedoch von vielen Programmierern innerhalb eines Unternehmens zu einem funktionalen Ganzen zusammengefügt. Diese Unternehmen verteilen dann die aggregierten Arbeiten in die Welt. Solche Distributoren sind zum Beispiel Oracle (OEL), Red Hat und Micro Focus (SUSE). Jede dieser Distributionen bietet eine kostenfreie Version der Software ohne Support sowie einen kostenpflichtigen Support für die Software an, manchmal auch eine Version der Software, die nur mit kostenpflichtigem Support zu erhalten ist. Die großen Vorteile der Open-Source-Software sind:

- Eine beinahe beliebig große Anzahl von Personen kann sich an der Entwicklung eines Open-Source-Programms beteiligen
- Der Nutzer ist nur eingeschränkt von einer bestimmten Herstellerfirma abhängig
- Die Nutzung ist nur an wenige Bedingungen geknüpft
- Bei der Vervielfältigung und Weiterverwendung fallen keine Lizenzkosten an

- Einblick in den Quelltext und üblicherweise auch in die Versionshistorie
- Prüfbarkeit eines Programms auf bewusst eingebrachte Mechanismen wie Backdoors

Der Support

Ein vorsichtiger Umgang empfiehlt sich bei allen Open-Source-Distributionen mit dem angebotenen Support. Bei den meisten Anbietern ist im Support-Vertrag eine „All or Nothing“-Klausel enthalten, die besagt, dass alle von der jeweiligen Distribution angebotenen Systeme in der erworbenen Support-Kategorie unter Support genommen werden müssen. Somit ist es nicht erlaubt, etwa einen Linux-Server einer geschäftskritischen Applikation unter Support zu nehmen und andere Linux-Server der gleichen Distribution nicht. Es muss also Support für jeden betriebenen Linux-Host der gewählten Distribution erworben werden. Im schlimmsten Fall kann es sein, dass bei gemischten Umgebungen mit mehreren Distributionen innerhalb einer virtuellen Umgebung die Host-Rechner mehrfach lizenziert werden müssen.

Aufgrund der Open-Source-Politik ist immer die Möglichkeit gegeben, den Anbieter des Supports zu wechseln, besonders wenn die zur Verfügung gestellten Support-Packs und -Patches auf dem gleichen Programmkernel aufsetzen, wie es zum Beispiel bei Red Hat und Oracle Linux der Fall ist. Hier reicht ein Entfernen aller Red-Hat-Markenzeichen mit einem von Oracle zur Verfügung gestellten Skript aus der bestehenden Software, um Oracle als neuen Distributor hinzufügen zu können. Ein Preisvergleich in der Sache lohnt immer, da zwischen den Distributionen der Markt heiß umkämpft ist.

OpenStack

Aus der Open-Source-Initiative hat sich in den letzten Jahren OpenStack entwickelt. Initiatoren des OpenStack-Projekts waren

Rackspace und NASA. OpenStack ist unter der Apache-Lizenz lizenziert und hat das Ziel, eine freie Architektur für Cloud Computing zur Verfügung zu stellen. Namhafte Unterstützer der OpenStack-Initiative sind unter anderem Oracle, Suse Linux GmbH, Red Hat, Dell, Citrix, AMD, Intel, HP, IBM, EMC, NetApp, Cisco, VMware und SAP.

OpenStack besteht aus mehreren Komponenten, zu denen die Unterstützer mehr oder weniger kostenfreie Software anbieten und mit Entwicklungsleistungen die Community unterstützen: OpenStack Compute (Nova), OpenStack Object Storage, OpenStack Image Service (Glance), OpenStack Identity (Keystone), OpenStack Dashboard (Horizon), OpenStack Block Storage, OpenStack Networking (Neutron), OpenStack Database Service (Trove), OpenStack Orchestration, OpenStack Telemetry (Ceilometer) und OpenStack Data Processing Service (Sahara). Die Verwaltung liegt bei der OpenStack-Foundation, die zur Förderung der weltweiten Entwicklung, Distribution und Adaption des OpenStack-Cloud-Betriebssystems gegründet wurde. Als weltweit unabhängige Dachorganisation für OpenStack betreut die Stiftung mehr als 30.000 Einzelmitglieder aus über 170 Ländern weltweit.

Das Partizipieren am OpenStack über die OpenStack-Foundation ist wie die Mitgliedschaft kostenlos. Es wird von den Mitgliedern jedoch eine Mitarbeit in Form technischer Beiträge erwartet, um die Community zu unterstützen.



Bernhard Halbetel
bernhard.halbetel@comparex.at



Apache Cassandra, wieso brauche ich das?

Jan Ott, Trivadis AG

Apache Cassandra ist eine NoSQL-Datenbank und konkurriert somit nicht nur mit anderen NoSQL-Datenbanken wie zum Beispiel MongoDB, sondern auch mit traditionellen SQL-Datenbanken wie der Oracle-Datenbank.

Wieso brauche ich Cassandra? Diese Frage hat sich der Autor mit seiner 20-jährigen Oracle-Datenbank-Erfahrung gestellt. Die Antwort seiner Kollegen und auf Konferenzen war:

- Einfach horizontal skalierbar mit gängiger Hardware
- Hohe Ausfallsicherheit mittels Redundanz und die Kenntnis von Rack/Data Center
- Herausragende Performance unter anderem durch sogenannte „Tunable Consistency“

- Kein aufwändiges Programmieren wie mit Map/Reduce; Cassandra hat eine SQL-ähnliche Sprache zur Daten-Manipulation (Cassandra Query Language, CQL)

In den letzten Jahren ist viel passiert. Cassandra findet immer mehr Verwendung. Die Apache Software Foundation und DataStax als Distributor können eine erlauchte Liste von Firmen präsentieren, die Cassandra einsetzen. Hier nur eine Auswahl: Apple, Twitter, Digg, Reddit, ING, UBS und Netflix. Apple betreibt im

Moment die wohl größte Cassandra-Installation mit mehr als 75.000 Nodes.

Große Unternehmen setzen Cassandra erfolgreich ein und verarbeiten enorme Datenmengen – dies macht neugierig. Nehmen wir Netflix als Beispiel. Der Film-Streaming-Dienst hat mit einer Oracle-Datenbank angefangen. Leider sind damit mehrere Probleme aufgetreten. Netflix wurde Opfer seines Erfolgs. Die Last nahm enorm zu und die Lösung mit der Oracle-Datenbank stieß an ihre Grenzen. Ein weiteres Problem waren die Offline-Zeiten für Struktur-Änderungen,

denn die Kunden wollen zu jeder Zeit streamen können. Diese Einschränkungen widersprachen den Expansionsplänen von Netflix. Es musste eine andere Datenbank gefunden werden, die die Last tragen kann, Struktur-Änderungen ohne Ausfallzeit erlaubt und flexibel für neue Standorte ist.

Nach einer Evaluationsphase entschied sich Netflix für Cassandra. Ausschlaggebend war, dass Cassandra die Last verarbeiten und für die ehrgeizigen Expansionspläne entsprechend skalieren kann. Dies ist gewährleistet durch die horizontale Skalierung. Mit der Redundanz und dem Verteilen über mehrere Racks und Data Center gibt es keinen „single point of failure“. Cassandra verkraftet sowohl den Ausfall einzelner Knoten wie auch des gesamten Data Center. Es braucht zudem kein Zeitfenster, in dem Netflix nicht verfügbar ist, um Schema-Änderungen vorzunehmen; sie können online appliziert werden. Somit stehen für die wichtigsten Problemstellungen entsprechende Lösungen bereit.

Der Streaming-Dienst betreibt mehr als 2.700 Nodes in 90 Clustern verteilt über vier Daten-Center. Cassandra kann die Last von mehr als einer Billion Operationen pro Tag verarbeiten. Wenn Netflix ein neues Land aufsetzen will, ist das in zehn Minuten erledigt. Das war im Jahr 2014 [1, 2].

Woher Cassandra stammt

Cassandra wurde ursprünglich von Facebook entwickelt. Eingeflossen ist das Wissen von Avinash Lakshman und Prashant Malik. Avinash hatte zuvor an Amazon Dynamo gearbeitet. Prashant Malik war bei Microsoft angestellt und arbeitete unter anderem an P2P-Netzwerken. Im Jahr 2008 wurde der Quellcode freigegeben – Open Source. Dadurch konnten auch andere Unternehmen wie LinkedIn, Apple und Twitter zu Cassandra beitragen. Im Jahr 2009 nahm Apache das Projekt auf, 2010 wurde es zu einem „Top Level“-Projekt erklärt und 2011 kam dann die Cassandra Query Language (CQL) hinzu.

Cassandra verwendet verschiedenste Konzepte, um die Anforderungen zu erfüllen. Beginnen wir mit der kleinsten Einheit, dem sogenannten Node. Als Node bezeichnet man einen einzelnen Rechner

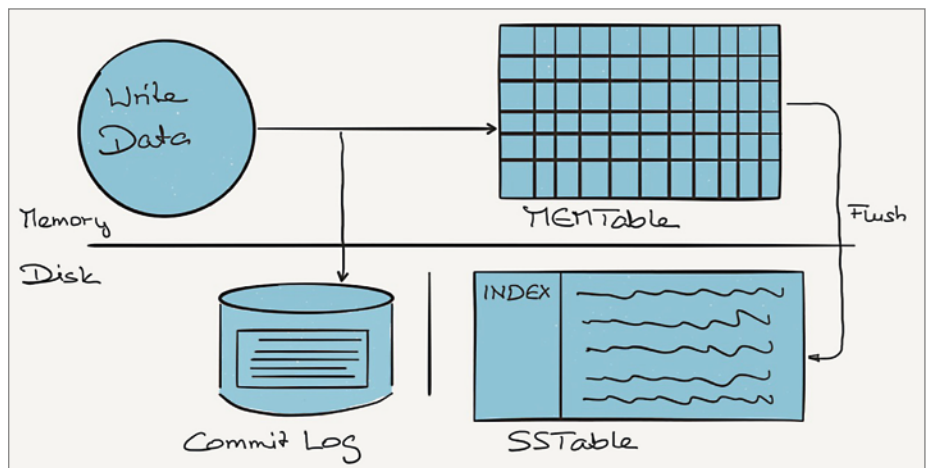


Abbildung 1: Schreiben von Daten in Cassandra

im Cluster. Es ist zwar möglich, Cassandra auf nur einem Rechner beziehungsweise Node zu betreiben, dies ergibt in der Praxis aber wenig Sinn, da die wichtigsten Vorteile von Cassandra verloren gehen. In der Praxis werden produktive Systeme daher immer als Cluster, bestehend aus mehreren Nodes, betrieben.

Ein einzelner Node ist dabei immer nur für einen Teil der Daten verantwortlich. Diese sollen jederzeit zuverlässig geschrieben und wieder gelesen werden können. Jeder Node muss zudem für jeweils einzelne Anfragen die Aufgabe des Koordinators übernehmen. Mehr dazu später.

Schreiben und Lesen von Daten

Beim Schreiben von Daten führt ein Node die in *Abbildung 1* dargestellten Schritte durch:

- Schreiben der Daten auf Disk – Commit Log
- Schreiben der Daten ins RAM – MEM-Table
- Bestätigung an den Client, dass die Daten geschrieben wurden

Die Daten werden dabei so lange in die MEMTable im Memory geschrieben, bis diese voll ist. Ist dies der Fall, schreibt („flushed“) ein Prozess die Daten in ein File auf Disk, eine sogenannte „SSTable“. Eine MEMTable wird so zu einer SSTable auf Disk. Ist die erfolgreich, können das Commit Log und MEMTable gelöscht wer-

den. Eine einmal geschriebene SSTable ist unveränderlich.

Für das Lesen der Daten führt ein Node die in *Abbildung 2* dargestellten Schritte durch:

- Prüfen, ob die Daten im Memory sind – MEMTable
- Prüfen, ob die Daten auf den Disks sind – SSTables
- Zusammenführen der gefundenen Daten (dabei kommt ein Zeitstempel zum Zug, den jeder Datensatz und jedes Attribut enthält, wobei immer der neueste Eintrag gewinnt)
- Das Resultat wird zurückgeben

Dies kann zeitaufwändig sein, da die Daten potenziell über mehrere SSTables verteilt sein können. Cassandra führt daher regelmäßig im Hintergrund einen Prozess aus, der die Einträge aus mehreren SSTables vereint und das Resultat in einer neuen SSTable speichert. Dies wird „Compaction“ genannt und beschleunigt das Lesen enorm, da eine Row nicht mehr über mehrere SSTables hinweg zusammengeführt werden muss.

Damit nicht immer die ganzen Files gelesen werden, gibt es Bloom Filter, Key Cache und mehr. Diese werden aus Platzgründen hier nicht weiter erläutert. Es reicht zu verstehen, dass ein Node für die Lese-/Schreib-Operationen optimiert ist. Wichtig ist: Beim Schreiben von Daten führt Cassandra, neben dem Hinzufügen zur MEMTable, lediglich eine Disk-Operation auf dem Commit-Log durch, wobei Daten immer nur am Ende einer Datei angefügt („append only“) werden. Sobald

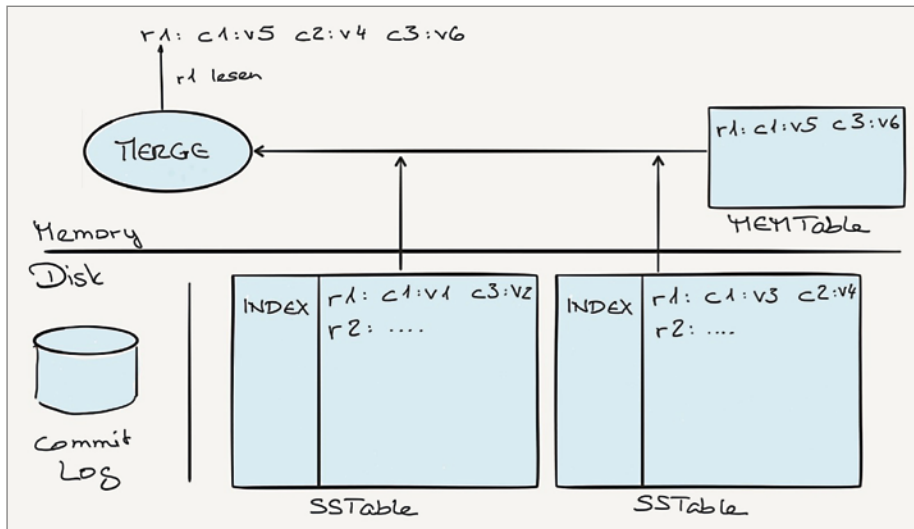


Abbildung 2: Lesen von Daten aus Cassandra

dies getan ist, wird die Bestätigung gegeben, dass alles erledigt ist. Dies ist enorm schnell und erlaubt daher konstant hohe Schreibraten in einen Cassandra Cluster.

Die Hardware

Für den Betrieb eines Nodes wird folgende Hardware empfohlen:

- RAM: 16 – 64 GB
- CPU: 8 Core (im Moment das beste Preis/Leistungs-Verhältnis)
- Disk: 500 GB – 1 TB bis zu einem Maximum von 3 – 5 TB
- Vorzugsweise SSD; HD-Spindeln sind auch möglich, wobei dabei zwingend Commit-Log und Datenfiles voneinander getrennt auf physisch unterschiedlichen Disks geschrieben werden sollen
- Disks müssen direkt mit dem Computer verbunden sein, also kein SAN – wenn es ein Netzwerk-Kabel braucht, ist etwas falsch
- LAN: mindestens 1 GB

Ein so ausgestatteter Node erreicht 3.000 bis 5.000 Schreib-/Lese-Operationen pro Sekunde und Core [3]. Anstatt eigene Hardware anzuschaffen, kann man Cassandra natürlich auch in der Cloud betreiben. Netflix verwendet dazu zum Beispiel Amazon Web Services (AWS) und Oracle hat auf der letzten OpenWorld die Unterstützung von Apache Cassandra auf dem Oracle Bare Metal Cloud Service angekündigt [4].

Ein Node lässt sich über das Node-Tool verwalten. Damit kann er gestartet, gestoppt und aus dem Cluster entfernt werden. Zudem lassen sich damit verschiedene Informationen über den Node und Cluster abrufen. Ein Node ist vergleichbar mit einer Ameise im Ameisenhaufen. Alleine nicht sehr stark, aber im Verbund können Datenberge versetzt werden. Für das Zusammenspiel solcher Nodes in einem Cluster braucht es eine Organisation. In Cassandra sind die Nodes virtuell in einem Ring angeordnet (siehe Abbildung 3).

Der Cassandra-Cluster

Ein Ring/Cluster ist ein Verbund von mehreren Nodes. Diese haben keinen Master, sämtliche Nodes sind gleichgestellt. Fällt einer aus, wird die Arbeit einfach von einem anderen Node übernommen.

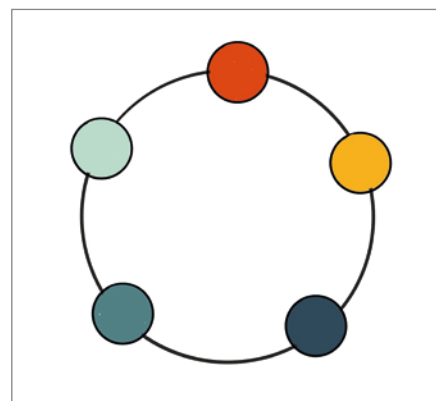


Abbildung 3: Ein Cassandra-Ring

Wie aber werden die Daten in einem Cassandra Cluster gespeichert beziehungsweise wer sorgt dafür, dass die Daten richtig verteilt werden? Diese Aufgabe übernimmt der sogenannte „Kordinator“. Jeder Node in einem Cluster kann Koordinator sein, der Koordinator wird von Cassandra jeweils für jede einzelne Anfrage bestimmt und die Zuordnung gilt genau nur für die eine Anfrage. Die Wahl des Koordinators kann über verschiedene Kriterien geschehen, wie zum Beispiel der Latenzzeit des Nodes, die Daten-Lokalität je Node etc. Ist der Koordinator bestimmt, bestehen grundsätzlich die folgenden Möglichkeiten:

- Der Koordinator ist gleichzeitig der Node, der auch die Daten besitzt, er kann die Daten also direkt schreiben beziehungsweise lesen
- Der Koordinator ist nicht für diese Daten zuständig, er muss also einen anderen Node kontaktieren, der die Daten schreibt beziehungsweise liest

Auf welchem Node die Daten gespeichert sind, wird mit dem Partition Key und einer Hash-Funktion eruiert. Eine Hash-Funktion ermittelt das sogenannte „Token“, das dann die Datenlokalität im Cluster bestimmt. Dieser Vorgang heißt „Consistent Hashing“.

Im folgenden Beispiel wird von einem Replikationsfaktor von drei ausgegangen. Dies bedeutet, dass jeder Datensatz auf drei Nodes repliziert wird, um die Verfügbarkeit zu erhöhen. Jeder Node hält die Daten für einen bestimmten Token-Ränge, in unserem Beispiel in Abbildung 4 sind dies für Node A der Token-Ränge 1 – 7, für Node B 8 – 14 und für Node C 15 – 21. Die Nodes C und D enthalten die Replikate von Node B. Wie auch die anderen Nodes zusätzlich noch Replikationen für andere Nodes speichern. Eine Schreib-Operation läuft wie in Abbildung 4 dargestellt ab:

1. Der Client verbindet sich mit einem beliebigen Node, der zum Koordinator wird, und sendet ihm die Daten, die geschrieben werden sollen. In unserem Beispiel ist dies Node E.
2. Node E, der Koordinator, empfängt die Daten und bestimmt das Token über den Partition Key, hier im Beispiel ist dies 12.
3. Die Daten werden gleichzeitig an

Node B, C und D gesendet; drei Nodes deshalb, weil der Replikations-Faktor „drei“ ist.

4. Jeder der drei Nodes (B, C, und D) bestätigt dem Koordinator (E), dass er die Daten erhalten und gespeichert hat.
5. Node E, der Koordinator, sendet ein „OK“ zurück an den Client.

Beim Schritt 5 lässt sich über den Konsistenz-Level je Operation steuern, wann die Bestätigung an den Client zurückgeschickt werden soll. So könnte das „OK“ auch schon nach der Bestätigung vom ersten Node (Schritt 4) zurückgegeben werden, was die Latenzzeit für den Client wesentlich verringert, dafür ist die Daten-Konsistenz nicht unmittelbar garantiert. Dies nennt man in Cassandra „Tunable Consistency“, mehr dazu später. Eine Lese-Operation läuft wie in *Abbildung 5* dargestellt ab:

1. Der Client verbindet sich mit einem beliebigen Node, der zum Koordinator wird, und schickt den Partition Key der Daten, die gelesen werden sollen. In unserem Beispiel ist dies Node E.
2. Node E, der Koordinator, empfängt den Partition Key und ermittelt über dieselbe Hash-Funktion den Token, also wiederum 12.
3. Node E macht eine Abfrage für die Daten auf B inklusive der Checksumme; er bittet Node C und D um eine Checksumme.
4. Node E erhält die Daten und die Checksummen.
5. Node E prüft, ob die Checksummen übereinstimmen.
6. Wenn alles ok ist, werden die Daten an den Client geliefert.

Ob beim Lesen tatsächlich alle drei Replikate angefragt und geprüft werden, hängt vom Konsistenz-Level ab, der bei der Read-Operation angegeben werden kann. Dies nennt man ebenfalls „Tunable Consistency“.

Tunable Consistency

Bei verteilten Systemen und dabei auch bei verteilten Datenbanken geht es um drei Faktoren: Consistency, Availability und Partition Tolerance – das sogenann-

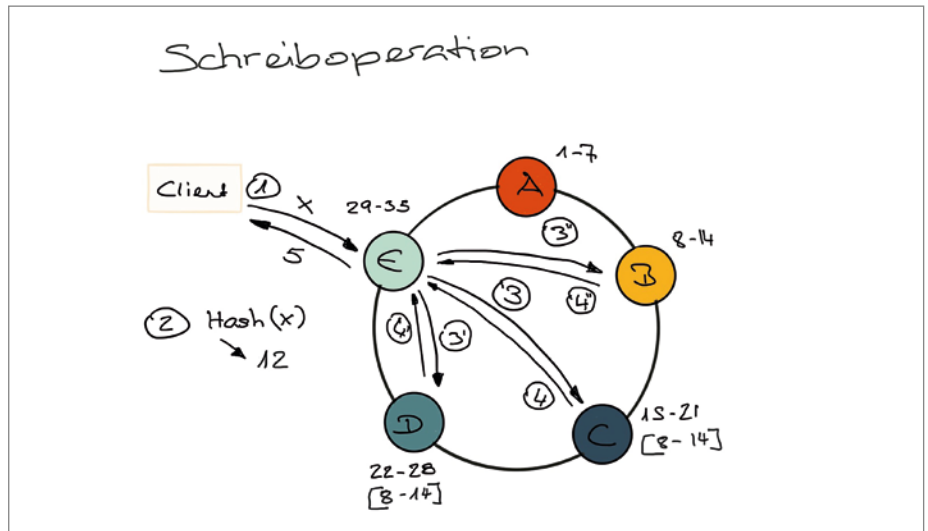


Abbildung 4: Eine Schreib-Operation

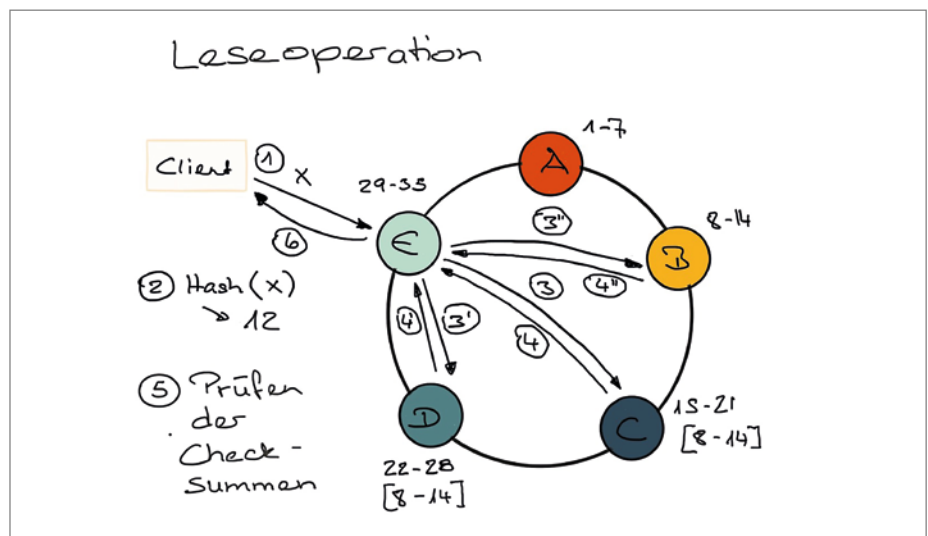


Abbildung 5: Eine Lese-Operation

te „CAP-Theorem“ (siehe *Abbildung 6*). Es ist jedoch nicht möglich, alle drei Faktoren gleichzeitig zu erfüllen. In [5] ist das gut erklärt.

Es geht somit nicht, dass man 100 Prozent konsistent, gleichzeitig 100 Prozent verfügbar und noch partitionstolerant sein kann. Also muss ein Kompromiss gefunden werden.

Cassandra setzt die Priorität auf Availability und Partition Tolerance. Das bedeutet aber nicht, dass Cassandra nicht auch konsistent sein kann. Es ist aber dem Entwickler überlassen, wie die Gewichtung erfolgen soll. Dafür steht in Cassandra die Tunable Consistency zur Verfügung. Hierzu zwei Beispiele: Schreiben-ONE und Schreiben-QUORUM (siehe *Abbildung 7*):

Consistency Level = ONE:

- Der Client verbindet sich mit B, der die Aufgabe des Koordinators übernimmt.
- B schreibt an alle drei Nodes (C, D und E), die die Daten erhalten müssen.
- Wenn der erste Node die Bestätigung schickt, ist das Consistency Level erfüllt
- B schickt eine Bestätigung zurück an den Client

Consistency Level = QUORUM: Mehr als 50 Prozent der involvierten Nodes müssen bestätigen, in unserem Fall mit Replikations-Faktor „drei“ sind dies zwei Nodes:

- Der Client verbindet sich mit B, der die Aufgabe des Koordinators übernimmt.

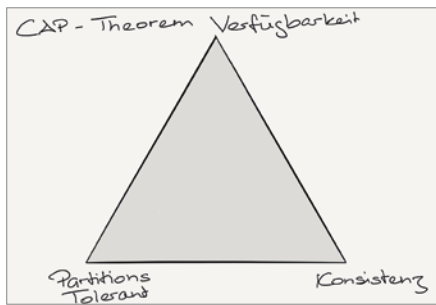


Abbildung 6: Das CAP-Theorem

- B schreibt an alle drei Nodes (C, D und E), die die Daten erhalten müssen.
- Wenn der zweite die Bestätigung geschickt hat, ist das Consistency Level QUORUM erfüllt
- B schickt die Bestätigung an den Client

Es gibt auch noch andere Level:

- ANY: Nur der Koordinator betätigt, dass er die Daten erhalten hat. Sie wurden aber noch nicht an die Nodes geschickt, die die Replikate speichern.
- ALL: Alle Nodes müssen bestätigen.
- ONE, TWO, THREE: Ein, zwei beziehungsweise drei Nodes müssen bestätigen.
- QUORUM: Mehr als 50 Prozent der Nodes müssen bestätigen
- LOCAL_ONE: Nur ein Node aus dem lokalen Data Center muss bestätigen.
- LOCAL_QUORUM: Mehr als 50 Prozent der Nodes des lokalen Data Center müssen bestätigen.
- EACH_QUORUM: Mehr als 50 Prozent der Nodes aus allen Data Center müssen bestätigen.

Dazu ein Beispiel: Bei Netflix ist nicht wichtig, bis zu welcher Sekunde man einen Film gesehen hat, also Level „ANY“ oder „ONE“. Die Filme, die man schon gesehen hat, sollten aber möglichst konsistent angezeigt werden, somit Level „QUORUM“. Ein Entwickler kann mit Cassandra je nach Anwendungsfall und Daten entscheiden, wie hoch die Konsistenz-Sicherheit sein muss, und dadurch die Geschwindigkeit und Verfügbarkeit des Systems beeinflussen – Tunable Consistency.

Ausfallsicherheit

Um die Ausfallsicherheit zu erhöhen, ist bei Cassandra die Unterstützung für

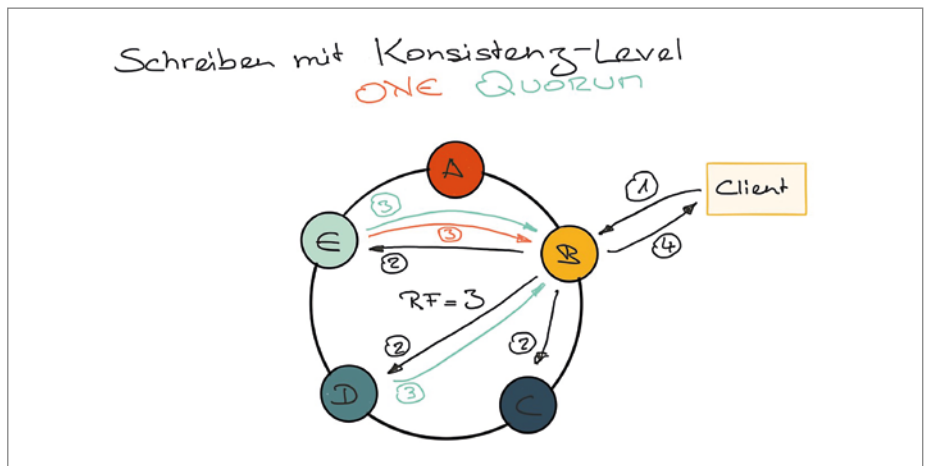


Abbildung 7: Schreiben mit Konsistenz-Level

Racks und Data Center von Haus aus eingebaut. Die Handhabung ist daher denkbar einfach. Es ist nur ein Parameter im Parameter File zu setzen, und Cassandra verteilt alles richtig über die Data Center hinweg. Im Vergleich dazu ist der Betrieb einer Oracle-Datenbank über mehrere Data Center hinweg sehr komplex. Generell sollte ein Oracle-Datenbank-Cluster wegen der Inter-Node-Kommunikation nicht mehr als ein Dutzend Nodes beinhalten. Doch wie steht das mit einem Ring von Netflix mit 2.700 Nodes? Wie kann das Cassandra handhaben?

In einem Cluster müssen Status-Mitteilungen fortlaufend ausgetauscht werden. Es wäre aber falsch, wenn nun jeder Node allen mitteilen würde, wenn er dem Ring beitrifft oder diesen verlässt. Dies würde das Netzwerk zu sehr beanspruchen. Hinzu kommt, dass jeder Node auch mitteilen soll, dass er noch am Leben ist.

Die Cassandra-Entwickler haben sich ein Verhalten der Menschen zunutze gemacht – das Geschwätz – sie nennen es „Gossip“. Ein Mensch erzählt etwas seinen Freunden und Bekannten. Die erzählen es wiederum ihren Freunden und Bekannten und so weiter und so fort. Das Ergebnis ist, dass innerhalb kurzer Zeit alle alles wissen. So funktioniert Gossip auch bei Cassandra. Ein Node teilt drei Nodes mit, dass er da ist. Die drei Nodes geben die Information an je drei Nodes weiter. So erfahren alle Nodes, dass ein neuer im Ring ist. Dieses Verfahren ist sehr effizient und belastet das Netzwerk nur gering. Selbst bei einer Größe von 2.700 Nodes wie bei Netflix. Was wird dabei mitgeteilt? Informationen zum Status, zum Data Cen-

ter, zu den Racks, zu den Schemata und noch weiteres. Das Ganze hat zudem einen Rhythmus, „Herzschlag“ genannt.

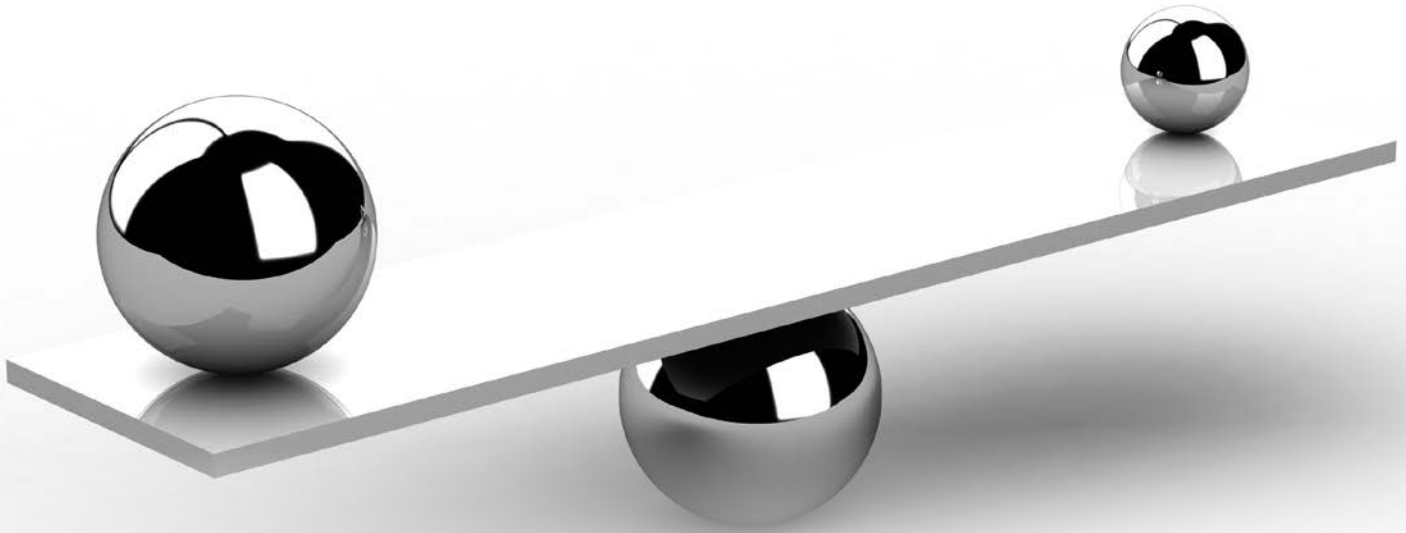
Somit wissen wir nun, wie Cassandra funktioniert. Wir haben ein paar der Methoden gesehen, mit denen auftretende Probleme gelöst werden können. In der nächsten Ausgabe zeigen wir, wie die Daten reingebracht sowie rausgeholt werden und wie man die Strukturen für die Daten erstellt.

Weitere Informationen

- [1] Netflix, Fallstudie 2011: <http://www.datastax.com/resources/casestudies/netflix>
- [2] Cassandra Summit 2014, Cassandra@Netflix: <http://www.slideshare.net/planetcassandra/cassandra-summit-2014-cassandra-netflix-building-a-house-of-cards-on-a-solid-foundation>
- [3] DataStax, Wahl der richtigen Hardware: https://docs.datastax.com/en/cassandra/2.0/cassandra/architecture/architecturePlanningHardware_c.html
- [4] Oracle-Ankündigung Cassandra auf „bare metal cloud“-Service: https://community.oracle.com/community/cloud_computing/bare-metal/blog/2016/10/24/datastax-certified-nosql-cassandra-clusters-on-bare-metal-cloud-service
- [5] CAP-Theorem, Wikipedia: <https://de.wikipedia.org/wiki/CAP-Theorem>



Jan Ott
jan.ott@trivadis.com



MySQL vs. Oracle-Datenbank

Oliver Sennhauser und Matthias Jung, DOAG Datenbank Community

MySQL und die Oracle-Datenbank sind zwei relationale Datenbanken aus dem Haus Oracle, die beide von sich behaupten, die ACID-Anforderungen zu erfüllen. Sie liegen beim „DB Engine Ranking“ (siehe „<http://db-engines.com/en/ranking>“), das weltweit alle wichtigen Datenbank-Systeme erfasst, zurzeit auf Platz eins und zwei. MySQL wird voraussichtlich in den nächsten zwölf Monaten die Oracle-Datenbank vom ersten Platz verdrängen. Der Artikel zeigt die Unterschiede, Einsatzbereiche und die Vorzüge des jeweiligen Produkts.

MySQL wurde in den 1990er-Jahren zu Beginn des Web-Booms als SQL-Layer um C-ISAM-Dateien entwickelt. Es war um die Jahrtausendwende eine relativ einfache und frei verfügbare Datenbank, die in zahlreiche neu aufkommende Web-Anwendungen integriert wurde und heute als eine der Haupt-Komponenten des sogenannten „LAMP-Stacks“ (Linux, Apache, MySQL, PHP) gilt. Daher ist MySQL aus der Web-Welt nicht mehr wegzudenken. Ohne MySQL würden sich die meisten Angebote des Internets quasi in nichts auflösen.

Die Oracle-Datenbank ist in den 1970er-Jahren in den USA entstanden und kommt seitdem in zahlreichen Unternehmen für geschäftskritische Prozesse zum Einsatz. Daher ist dieses Produkt meist eng mit den Geschäftsprozessen verflochten und oft nur schwer daraus herauszulösen.

Eigenschaften

Beide Datenbanken bezeichnen sich als relationales Datenbank Management System (RDBMS), wobei beide auch über den Tellerrand hinausschauen und versuchen, im Umfeld der relationalen Welt weitere Funktionalitäten abzudecken.

MySQL bietet mit seinem Storage-Engine-API zahlreiche Möglichkeiten, um die Funktionalität des Produkts zu erweitern. Dies führt dazu, dass wir heute die Wahl haben zwischen einer transaktionalen, Crash-sicheren, relationalen Storage-Engine namens „InnoDB“, einer Storage-Engine, die Daten ausschließlich im Speicher ablegt („In-Memory“), einer spaltenorientierten Storage-Engine („Column Store“), einer Graphen-Storage-Engine und einer Storage-Engine, die für das „Sharding“ zuständig ist. Auf der anderen Seite verfolgt

die Oracle-Datenbank eher den Ansatz von Integration. XML und OODBMS sind in das Datenbanksystem integriert; Java läuft im Datenbank-Kernel; In-Memory und Column Stores sind in das Produkt integriert. Die Qual der Wahl hat man eher nicht, man darf oder muss nehmen, was der Hersteller für einen als gut befunden hat. Dies kann ein Vor- oder auch ein Nachteil sein. Von Nachteil ist jedoch die komplizierte Lizenzpolitik seitens Oracle, die zusätzlich beim Einsatz bestimmter Funktionen der Oracle-Datenbank zu betrachten ist.

Umfang

MySQL hat aus der Not eine Tugend gemacht und verfolgt recht stark das KISS-Prinzip (Keep It Stupid and Simple). Das heißt in der Praxis, dass sich in MySQL al-

les recht einfach, schnell und schnörkellos gestaltet. Dafür hat man nicht denselben Funktionsumfang und dieselben Möglichkeiten wie bei der Oracle-Datenbank und muss sich gegebenenfalls selbst einen Weg überlegen, um eine vergleichbare Lösung zu implementieren. Das beginnt schon mit der Installation: Jede gängige Linux-Distribution bietet von Haus aus eine MySQL-Distribution an. Die Installation erfolgt oft über das Paket-Management-System der Distribution. Alternativ lässt sich auch das Repository von Oracle einbinden. Eine Installation von MySQL benötigt in der Regel genau einen Befehl wie „yum install mysql-server“. In zwei bis drei Minuten ist ein lauffähiges MySQL-System verfügbar, das je nach Bedarf noch weiter konfiguriert werden kann.

Eine Oracle-Datenbank-Installation ist meist deutlich aufwändiger und beginnt mit dem Download von mehreren GB umfassenden Software-Archiven. Im Anschluss daran muss die Software auf den entsprechenden Server kopiert und der Oracle Universal Installer (OUI) aufgerufen werden. Über mehrere Dialog-Seiten hinweg werden Parameter zur Installation der Software und – sofern gewünscht – zum direkten weiteren Aufbau einer Datenbank ermittelt. Natürlich lässt sich dieser Prozess auch automatisieren („Silent Installation“). Allerdings bedarf es hier zur Vorbereitung einer gewissen Einarbeitungszeit, um sich mit den Mechanismen dieses Verfahrens vertraut zu machen.

Wie auch die Oracle-Datenbank unterstützt MySQL die gängigsten Plattformen wie Oracle Linux, Red Hat Enterprise Linux, SuSE Enterprise Linux, Ubuntu Server und natürlich auch Windows. Auf den Linux-Plattformen wird man am wenigsten auf Probleme stoßen und Bugs sowie entsprechende Workarounds dazu finden. Da MySQL die Pakete regelmäßig und automatisiert baut, kommen die Versionen für alle unterstützten Plattformen praktisch zeitgleich heraus. Ein neues Minor-Release wie MySQL 5.7.17 (in der Oracle-Sprache „Patch“) kann etwa alle sechs bis acht Wochen erwartet werden, ein Major-Release (wie MySQL 8.0) etwa alle achtzehn Monate. Aktuelle Versionen bei MySQL sind zurzeit MySQL 5.6 und 5.7. Die nächste Version wird MySQL 8.0 heißen und ist bereits als Beta-Release verfügbar. Die Freigabe dieser neuen Version wird wahrscheinlich im Jahr 2017 für den produktiven Einsatz erfolgen.

Wie auch die Oracle-Datenbank verfügt MySQL über eine Konfigurationsdatei namens „my.cnf“ (auf Windows „my.ini“). Diese liegt üblicherweise unter „/etc“ (Red Hat und SuSE) oder „/etc/mysql“ (Ubuntu und Debian). Sie kann aber auch an einer beliebigen anderen Stelle liegen. In diesem Fall ist einfach dem Start-Prozess mitzuteilen, wo er sie findet.

Im praktischen Einsatz

Das Starten und Stoppen einer MySQL-Instanz erfolgt über den Betriebssystem-eigenen Mechanismus („init“-Prozess). Zurzeit sind die Linux-Distributionen im Übergang von SysV/Upstart nach SystemD. Es herrscht daher noch etwas Verwirrung, wie es richtig geht, und es hat sich möglicherweise die eine oder andere unsaubere Implementierung eingeschlichen. Dies sollte aber nur ein vorübergehendes Problem während dieser Übergangsphase sein. Multi-Instanzen-Set-ups sind in der Oracle-Welt üblich. Dies ist unter anderem auf die Lizenz-Politik von Oracle zurückzuführen, da die Lizenzierung auf Socket und Kernen basiert. Bei MySQL fallen in den meisten Fällen sowieso keine Lizenz-, sondern höchstens Support-Subskriptions-Kosten an. Von daher wird MySQL typischerweise nicht als Multi-Instanzen-Set-up installiert. Im Bedarfsfall kann man MySQL natürlich auch in diesem Modus betreiben.

Was für den Oracle-DBA „SQL*plus“, ist für den MySQL-DBA „mysql“. Viele Administratoren arbeiten gerne mit dem Kommandozeilen-Interface (CLI). Im Vergleich zur Oracle-Datenbank ist das MySQL-CLI wesentlich weniger mächtig. Dies ist unter anderem darauf zurückzuführen, dass zu der Zeit, als MySQL entwickelt wurde, der Bedarf nach starken CLIs nicht mehr so groß war, weil grafische Benutzeroberflächen schon weitverbreitet waren. Im Unterschied dazu war man in den Anfängen der Oracle-Datenbank noch auf eine gute CLI zwingend angewiesen.

Die grafische Oberfläche der Wahl ist bei MySQL heute die MySQL-Workbench. Sie beinhaltet Werkzeuge für den Admin auf der einen, für den Entwickler auf der anderen Seite. Zusätzlich bietet dieses Toolset einen recht brauchbaren ER-Diagrammer sowie ein Tool für die Datenmigration von anderen Datenbank-Systemen wie Oracle. Sämtliche Anwendungen,

die eine ODBC- oder JDBC-Schnittstelle nutzen, sollten in der Lage sein, mit MySQL zu kommunizieren und Daten zu verarbeiten. Sogar der SQL Developer von Oracle kann mit MySQL arbeiten.

Wenn man sich mit der MySQL-Datenbank verbunden hat, wird der Oracle-DBA sehr wahrscheinlich etwas verwirrt sein. Sowohl die User als auch die Schemata und Datenbank-Objekte sowie deren Beziehungen zum User sind anders, als dies bei der Oracle-Datenbank implementiert wurde. Ein Account setzt sich immer aus einem User-Namen und einem Host oder einer Host-Range zusammen, von wo aus sich ein User verbinden darf. Das kann dazu führen, dass es zwei User mit demselben Namen in der Datenbank geben kann, etwa „oli@localhost“ und „oli@192.168.%“. Diese können zu allem Überdross zusätzlich noch unterschiedliche Passwörter und Privilegien haben.

Bei der Oracle-Datenbank entsteht ein Schema mit dem Kreieren eines Objekts, das einem User gehört. Der User darf als Besitzer des Objekts dieses vollständig verwalten. Bei MySQL ist das anders: Ein Objekt (Tabelle, View, Trigger etc.) gehört immer dem System. Ein User kann nur Rechte haben, mit dem Objekt bestimmte Aktionen durchzuführen (EXECUTE, INSERT, SELECT etc.). Rollen wie bei der Oracle-Datenbank kennt MySQL zurzeit noch nicht. Das ist allerdings nicht weiter tragisch, da heute Rollen-Konzepte üblicherweise in der Applikation und nicht wie früher üblich in der Datenbank abgebildet sind. Mit der Version 8.0 wird es in naher Zukunft jedoch auch Rollen in MySQL geben.

Eine Oracle-Datenbank ist mit mehr Objekt-Typen ausgestattet als MySQL. Dort kennt man nur Tabellen, Indizes, Views und Stored Programs (Procedures, Functions, Triggers, Events), wobei man Letztere heute tendenziell eher nicht mehr verwendet, da Business-Logik heute im Business-Layer angesiedelt ist und weniger häufig im Daten-Layer (DBMS), wie das früher noch häufig der Fall war und auch von den Datenbank-Herstellern stark gefördert wurde. Wenn ein Oracle-PL/SQL-Entwickler Stored Programs für MySQL erstellen soll, wird er voraussichtlich etwas enttäuscht sein. Der Funktionsumfang, die Integration in zahlreiche Entwicklungs-Umgebungen, die Debugging-Möglichkeiten und vor allem auch das intuitive Arbeiten wird man bei MySQL

stark vermissen. Wie bereits gesagt, heute entwickelt man Software anders als früher. Eine stark integrierte Sprache ist heute nicht mehr zwingend erforderlich und bindet Business-Prozesse unnötig an dedizierte Datenbank-Systeme.

Bei den Daten-Typen kennt MySQL alle wichtigen Basiselemente wie „INT“ (klein bis groß), „FLOAT“, „DECIMAL“, „VARCHAR“, „DATE“, „TIME“, „DATETIME“, „BLOB“, „TEXT“ (= CLOB) etc. Komplexe oder zusammengesetzte Datentypen wie bei der Oracle-Datenbank kennt MySQL derzeit nicht. Neue Datentypen wie „GIS“ (POINT, POLYGON etc.) oder „JSON“ gehen zwar in diese Richtung, sind aber eher aus Sicht der Applikation und nicht von der Datenbank aus zu betrachten.

Da MySQL schon früh weltweite Verbreitung fand und auch auf Feedback von weit entfernten Kulturen einging, ist die Verwendung verschiedener Character-Sets wie UTF8, latin1 oder auch ein Mischen von Sets kein Problem. MySQL kann das Character-Set auf Tabellen- oder sogar auf Spalten-Ebene festlegen. Ein nachträgliches Ändern des Character-Sets auf Instanz, Schema, Tabelle oder Spalte ist kein Hexenwerk.

Wie die Oracle-Datenbank kennt MySQL inzwischen auch das Konzept von Tablespaces. Aus verschiedenen Gründen ist dieses allerdings nicht so ausgefeilt wie bei der Oracle-Datenbank, was den Vorteil bietet, dass es im Handling recht einfach ist. MySQL kann entweder alle Tabellen in einem Tablespace ablegen (System Tablespace, ähnlich wie es ganz früher auch bei der Oracle-Datenbank der Fall war), oder aber jede Tabelle hat ihren eigenen Tablespace. Das führt natürlich zu Problemen, wenn man Instanzen mit Zehntausenden bis Millionen von Tabellen hat. Seit Kurzem kennt MySQL daher auch das Konzept von allgemeinen Tablespaces („General Tablespaces“). Darin lassen sich, wie von der Oracle-Datenbank gewohnt, Tabellen nach Wunsch ablegen. Das hat auf der einen Seite den Vorteil von weniger Dateien im Filesystem, auf der anderen Seite aber den Nachteil, dass man sich Gedanken darüber machen muss, wo die Tabellen abgelegt werden sollen.

Auf physischer Ebene verwendet MySQL das Filesystem des Betriebssystems. Eine entsprechende Zwischenschicht wie Oracle ASM gibt es nicht. Das kommt dem Linux-Administrator entgegen, der neben MySQL noch zahlreiche andere Anwendun-

gen zu betreuen hat und sich nicht in solche Spezialitäten einarbeiten muss (KISS). Der Spatial-Option der Oracle-Datenbank entsprechen bei MySQL die GIS-Datentypen und -Indizes (zweidimensionale Indizes). In MySQL sind diese OpenGIS standardkonform und für zahlreiche neue Anwendungen nutzbar, die dank OpenStreetMap und Google Maps sowie der mobilen Devices mit Standort-Angaben möglich werden. So lassen sich Abfragen wie „Welche Restaurants liegen im Umkreis von hundert Metern von meiner Position und haben jetzt gerade geöffnet?“ problemlos verarbeiten.

Was die Oracle-Datenbank mit seiner Text-Option anbietet, entspricht der MySQL-Volltext-Suche mit Volltext-Indizes. Dies ist eine recht praktische Möglichkeit, ganze Texte von kleinerer bis mittlerer Größe in der Datenbank zu durchsuchen. Bei großen Textmengen im Bereich von Tera- bis Petabyte weicht man besser auf eine spezialisierte Lösung aus, etwa das heute moderne Elasticsearch. Vor rund fünfzehn Jahren wurde XML populär. Oracle hat diese Strömung stark unterstützt und mit der XML-Option in die Datenbank integriert (heute Standard mit 12c). Es hat sich aber, bedingt durch seine Komplexität, nicht in allen Bereichen gleichermaßen durchgesetzt. MySQL unterstützt ebenfalls marginale XML-Funktionalitäten. Zielführender und zukunftsweisender ist jedoch der neue JSON-Trend, der zahlreiche XML-Einsatzgebiete effizienter abdecken kann – angefangen bei einem eigenen Datentyp „JSON“ und zahlreichen Funktionen, um die Suche und die Manipulation von JSON-Dokumenten zu erleichtern und auch stark zu beschleunigen. Mit der Einführung von JSON in der Version 5.7 ist MySQL jetzt also auch zum Document-Store geworden. Dies stößt aktuell auf eine sehr gute Resonanz in der Oracle-Entwickler-Gemeinde bei der Neuentwicklung von Projekten. Die Oracle-Datenbank hat mit ihrer neuesten Version ebenfalls im Bereich von JSON Fortschritte erzielt und unterstützt auch diesen neuen Trend.

Wie bereits erwähnt, bietet MySQL mit seinen vielseitigen Plug-in-APIs zahlreiche Möglichkeiten für Erweiterungen. Dies sind einerseits Storage-Engines zum Ablegen der Daten in unterschiedlichen Formaten oder Speicherformen. Das Volltext-Parse-Plug-in bietet die Möglichkeit, Volltextsuchen für westliche und asiatische Schriften zu implementieren. Die Auditing-, Authentication- und Password-Val-

idation-APIs erlauben hier die Anbindung an sämtliche verfügbaren externen Quellen und das Query-Rewrite-API ermöglicht es, Plug-ins zur Manipulation von Queries zu erstellen, wenn sich zum Beispiel die alte Applikation nicht an das neue Datenbank-Verhalten adaptieren lässt.

Datensicherheit

Im Bereich „Backup/Restore“ hat MySQL dem RMAN der Oracle-Datenbank nichts Gleichwertiges entgegenzusetzen. Zwar schreibt MySQL sogenannte „Binary Logs“, die in etwa den Oracle-Archive-Logs entsprechen, und hat ebenfalls eine physische Backup-Methode. Diese entspricht aber eher den alten Oracle-Tablespaces.

Für kleinere Datenbanken empfiehlt es sich, ein logisches Backup mit dem Werkzeug „mysqldump“ zu erstellen. Dieses ähnelt „exp/imp“ bei der Oracle-Datenbank. Im Unterschied dazu kann jedoch auf einem mit „mysqldump“ erzeugten Backup ein Point-in-Time-Recovery mit den Binary-Logs erfolgen. Allerdings muss der DBA selbst dafür sorgen, dass er die richtige Position des benötigten Binary-Log erwischt – ansonsten können beim PiTR Lücken mit fehlenden Daten entstehen. Mit dieser Methode können praktisch statementgenau Recoveries erfolgen, wenn der Admin das genaue Statement beziehungsweise den Zeitpunkt (Log-Position des Statements) kennt. Bei größeren Datenmengen ist ein logisches Backup mittels „mysqldump“ nicht mehr das adäquate Mittel der Wahl. In diesem Fall empfiehlt es sich, die physische Backup-Methode namens „MySQL Backup“ zu verwenden. Damit lassen sich Datenbanken bis in den Terabyte-Bereich sichern, sofern die darunterliegende Infrastruktur auch für diese Datenmengen ausgelegt ist. Nach dem erfolgten Restore findet, wie bereits bei der logischen Backup-Methode, ein Point-in-Time-Recovery mithilfe der MySQL-Binary-Logs statt. Auch in diesem Fall gilt es wieder, die richtige Position im richtigen Log-File zu finden. Diese kann in der Regel jedoch – bei sorgsamer Vorbereitung – problemlos ermittelt werden.

Hochverfügbarkeit

Bei den Hochverfügbarkeits-Lösungen kann sich MySQL durchaus mit der

Oracle-Datenbank messen. Failover-Cluster lassen sich bei MySQL genauso implementieren, da diese hauptsächlich durch Betriebssystem-Mittel beziehungsweise durch eine entsprechende Cluster-Software abgedeckt sind. Es gilt nur, die richtigen Module zu finden und MySQL entsprechend richtig mit diesen zu integrieren. Failover-Cluster sind jedoch tendenziell eine eher veraltete Technologie. Heute löst man Hochverfügbarkeit mit MySQL durch den Ansatz der Replikation. Analog zu Oracle Data Guard kennt MySQL die sogenannte „Master/Slave-Replikation“. Dies klingt zwar weniger spektakulär, ist aber ähnlich: Die Applikation schreibt in den aktiven Master-Knoten, der wiederum im Streaming-Verfahren die ganzen Daten-Änderungen auf den Slave repliziert. Dort werden diese abgearbeitet und auf die Datenbank appliziert. Im Unterschied zur Oracle-Datenbank können die Daten auf dem Slave-System gelesen und auch geschrieben werden. Letzteres sollte allerdings unterlassen werden, wenn die Daten-Konsistenz der Replikation gewährleistet werden soll.

Der Replikationsvorgang in der MySQL-Master/Slave-Replikation ist asynchron. Im Falle eines Hardware-Ausfalls kann es daher sein, dass Transaktionen auf dem Master bereits committet wurden, auf dem Slave hingegen noch nicht eingetroffen sind. Im schlimmsten Fall würde dies bedeuten, dass einige Transaktionen verloren gehen können. Wenn dies unter gar keinen Umständen geschehen darf, bietet MySQL die semi-synchrone Replikation. Im Unterschied zur asynchronen Replikation wird hier die Transaktion erst als vollständig akzeptiert, wenn mindestens ein Slave deren Erhalt bestätigt hat. Dies hat den großen Vorteil, dass bei einem Hardware-Ausfall garantiert ist, dass kein Datenverlust mehr entstehen kann. Es hat aber auch den Nachteil, dass der Commit verzögert wird, bis mindestens ein Knoten die Transaktion bestätigt hat, was zu höheren Commit-Latenzen führt und damit einen Einfluss auf die Performance der Applikation hat.

Eine Master/Slave-Replikation hat gewisse operative Nachteile, insbesondere bei Upgrades und sonstigen Wartungsarbeiten, denn die Rollen von Master und Slave müssen wechseln. Das ist zwar kein Hexenwerk, erfordert jedoch eine gewisse Konzentration seitens des DBA, da ansonsten unnötig lange Unterbrechungen oder gar Fehler auftreten können.

Der genannte Rollentausch wird mit dem virtuell synchronen Multi-Master-Galera-Cluster für MySQL vermieden. Dabei kann gleichzeitig auf alle drei Knoten (eine ungerade Anzahl größer eins) geschrieben und von allen Knoten gelesen werden. Es ist daher ein voll symmetrischer Cluster mit mehreren gleichwertigen Knoten. Mit Galera-Cluster für MySQL lässt sich im laufenden Betrieb ein Knoten aus dem Cluster entfernen. Auf diesem Knoten können dann Wartungsarbeiten erfolgen. Anschließend wird der Knoten wieder in den Cluster eingefügt und synchronisiert sich selbstständig. Fortlaufend wird derselbe Vorgang mit den weiteren Knoten im Cluster durchgeführt. Diese Operation nennt man auch „Rolling-Cluster-Restart“. Der Datenbank-Cluster lässt sich also wartungsfensterfrei im laufenden Betrieb pflegen. Das Äquivalent dazu ist der Real Application Cluster (RAC) bei der Oracle-Datenbank.

Beim Monitoring für MySQL gibt es zwei Ansätze: Entweder verwendet man den MySQL-hauseigenen MySQL Enterprise Monitor, der wohl eine der häufigsten Monitoring-Lösungen für MySQL ist, oder das Oracle Cloud Control Plug-in für MySQL. Dieses ist noch wesentlich weniger mächtig als die MySQL-Monitoring-Lösung. Über kurz oder lang sollen wohl Pläne im Hause Oracle bestehen, den vollen Funktionsumfang vom MySQL Enterprise Monitor in Cloud Control zu integrieren. Das wird allerdings wohl noch eine Zeit dauern, was verständlich ist, wenn man die Mächtigkeit und die Spezifika des MySQL Enterprise Monitor näher betrachtet.

Einsatzgebiet der Oracle-Datenbank und von MySQL

Welche Datenbank soll man wann und wo einsetzen? Dies ist eine schwere Frage, die man praktisch selten mit technischen Stärken oder Schwächen beantworten kann. MySQL ist sehr stark im Web-Umfeld verankert und lässt sich wohl nur schwer aus dem LAMP-Stack durch eine Oracle-Datenbank substituieren. Hat man auf der anderen Seite eine Anwendung, die nur mit einem spezifischen Datenbank-Backend funktioniert, ist eigentlich bereits klar, welche Lösung einzusetzen ist. Daher empfiehlt es sich, als Erstes auf die Empfehlung des Herstellers der Applikation zu hören. Weitere Fragen sind im Bereich

der Kosten zu klären. Lohnt sich der Kostenaufwand für den Betrieb einer Oracle-Datenbank bei der Wertschöpfung des zu betreibenden Dienstes? Weitere wichtige Punkte sind: Gibt es bereits Know-how in der entsprechenden Technologie beziehungsweise wie schnell kann man dieses aufbauen (DBA und Entwickler) und wie teuer kommt es, sich dieses Know-how zu generieren oder einzukaufen? Darüber hinaus kommen natürlich auch politische Punkte zum Tragen: Ist eine Anwendung vollständig in Oracle-PL/SQL geschrieben, so ist der Aufwand enorm, diese Anwendung in eine andere Sprache auf eine andere Datenbank zu portieren. Wir haben hier den perfekten Vendor-Lock-in.

Eine Studie, die bereits vor etlichen Jahren durchgeführt wurde, hat ergeben, dass etwa 75 bis 80 Prozent der Anwendungen aufgrund der technischen Anforderungen mit MySQL hätten gelöst werden können. Die Probleme liegen also meist nicht im technischen Bereich, sondern im Umfeld. Eine technische Einschränkung besteht mit MySQL allerdings: Der Einsatz im zwei- oder gar dreistelligen Terabyte-Bereich ist mit MySQL schwierig und gewisse Probleme sind nicht mehr mit vernünftigem Aufwand lösbar. Aber bei solchen Größen dürfte auch die klassische Oracle-Datenbank Schwierigkeiten bekommen ...



Oliver Sennhauser
oliver.sennhauser@doag.org



Matthias Jung
matthias.jung@doag.org



MySQL Branches und Forks

Oliver Sennhauser, DOAG Datenbank Community

MySQL, die Datenbank, die das Internet antreibt, gehört heute zu Oracle. Ihren Anfang hat die Firma in der ersten Hälfte der 1990er-Jahre genommen.

Gesucht wurde eine Firma, die einen einfachen SQL-Layer für C-ISAM-Dateien schreibt. Drei gewitzte, finnische Entwickler trauten sich diese Aufgabe zu und führten in wenigen Monaten diesen Auftrag aus. Das Ganze scheint ein Erfolg gewesen zu sein, denn einige Jahre später entstand daraus eine Firma namens „MySQL“. „My“ ist der Name der Tochter eines der Entwickler, daher stammt auch der Name der Firma.

Über die Zeit wurde das Produkt immer umfangreicher und mächtiger. Der limitierte Funktionsumfang der ISAM-Storage-En-

gine wurde erweitert und es entstand die MyISAM-Storage-Engine, die heute immer noch ab und zu genutzt wird. In dieser Zeit wurde MySQL auch unter die Open-Source-Lizenz GPL v2 gestellt. Als weitere Storage-Engine hat die Berkeley-Datenbank (BDB) Eingang in MySQL gefunden, ein transaktionaler Tabellentyp, der im Unterschied zur MyISAM-Engine, die ausschließlich Table-Level-Locks macht, Block-Level-Locks beherrscht.

Kurz nach der Jahrtausendwende hat MySQL während des großen Internet-Booms Eingang in den sogenannten

„LAMP-Stack“ (Linux-Apache-MySQL-PHP) gefunden. Man brauchte zu der Zeit eine einfache, preiswerte SQL-Datenbank, um die Bedürfnisse der boomenden Internet-Industrie befriedigen zu können. Da war MySQL genau zur richtigen Zeit am richtigen Ort – der Hauptgrund für den sensationellen Erfolg dieses Produkts.

Ebenfalls in diese Zeit fällt eine weitere Storage-Engine, die ebenfalls eine kleine, finnische Firma entwickelt hat, die InnoDB-Storage-Engine. Sie war ursprünglich als kleiner, schlanker Oracle-Ersatz gedacht. Diese Storage-Engine er-

füllt die ACID-Anforderungen (Atomicity, Consistency, Isolation, Durability), ist daher Crash-sicher und skalierte dank ihres zeilenbasierten Lockings wesentlich besser als die dazumal gängige MyISAM-Storage-Engine.

Einem unaufhaltsamen Aufstieg von MySQL stand nichts mehr im Wege. Man plante bei MySQL sogar bereits einen Börsengang, zuerst an der Stockholmer Börse, dann aber mit etwas mehr Finanzappetit an der Technologiebörse NASDAQ in den USA. Doch wie so oft im Leben kommt es manchmal anders, als man denkt: Im November 2005 kündigte Oracle den Kauf von Innobase OY, der Firma hinter der InnoDB-Storage-Engine, für eine nicht näher genannte Summe an. Das Herzstück wurde aus dem MySQL-Konglomerat herausgekauft. Schlechte Karten für einen Börsengang. Dieser Tag ging in die MySQL-Annalen auch als der InnoDB-Black-Friday ein.

Das MySQL-Management ließ sich dadurch nur kurz beeindruckten. Mit einigen geschickten Schachzügen konnte man das InnoDB-Problem umgehen und es gelang im April 2008, MySQL für eine Milliarde US Dollar an Sun Microsystems zu verkaufen. Der Enthusiasmus war groß und die MySQL-Mitarbeiter wurden bei Sun freudig begrüßt. Doch die Freude währte nur kurz: Nach zahlreichen Gerüchten übernahm Oracle ein knappes Jahr später Sun Microsystems. MySQL gehörte damit ebenfalls zu Oracle. InnoDB und MySQL waren somit wieder unter demselben Dach vereint.

MariaDB

Bis zu diesem Zeitpunkt war die Welt für die MySQL-Nutzer noch einfach und in Ordnung. Es gab nur eine Quelle der Wahrheit: MySQL aus dem Hause Oracle. Doch einigen Mitarbeitern, den ursprünglichen Gründern von MySQL, passte diese Entwicklung nicht (nachdem man eine Milliarde US Dollar abkassiert hatte). Da MySQL unter der Open-Source-Lizenz GPL v2 steht, ist jedermann dazu berechtigt, den öffentlich zugänglichen Quellcode von MySQL zu nehmen, einzusehen, zu ändern und sogar wieder zu verteilen, sofern man sich an die Regeln der GPL-v2-Lizenz hält. Dies wurde auch von besagten Entwicklern prompt getan.

Sie nahmen den MySQL-Quellcode und brachten ihn, ursprünglich nur leicht modifiziert, unter dem neuen Namen „MariaDB“ heraus. Diesen Vorgang bezeichnet man technisch auch als „Branchen“.

An diesem Punkt stellt sich die Frage, wie Oracle oder eine Firma MariaDB Geld mit MySQL verdient, um die Entwickler, Supporter, QA-Tester, Berater, Trainer, Manager, Verkäufer, Marketing-Leute und das ganze restliche Fußvolk zu bezahlen? Bei einer klassischen Software-Firma (nicht Open Source) geschieht dies primär über das Lizenzgeschäft und sekundär über das Support-Abo-Geschäft. Weitere Dienstleistungen wie Schulung und Beratung sind bedingt durch ihre relativ geringen Gewinnmargen von 20 bis 40 Prozent nicht sonderlich lukrativ und daher nicht so beliebt. Außerdem skalieren die letztgenannten Bereiche nicht besonders gut und werden eher als Kundenbindungs-Instrument gesehen. Sie sind quasi nur Enabler für das eigentliche Lizenz- und Support-Abo-Business.

Was die Mehrheit der MySQL-Nutzer nicht weiß, ist, dass Oracle das Recht hat, MySQL auch unter einer Oracle-proprietären Nicht-GPL-Lizenz zu vertreiben. Dieses Recht hat seinen Ursprung schon bei der alten Firma MySQL, die sich in weiser Voraussicht von allen Community-Kontributoren eine Abtretung des Verwertungsrechts in einem sogenannten „Contributor License Agreement“ (CLA) hat zusichern lassen. Daher ist Oracle heute die einzige Firma, die für MySQL auch Geld durch Lizenzen schöpfen darf. Alle anderen Mitbewerber müssen sich ihren Umsatz durch Dienstleistungen im MySQL-Umfeld erwirtschaften.

Die Theorie besagt, dass Open-Source-Firmen ihren Ertrag nicht durch Lizenzen erwirtschaften (mit einer Marge von gegen 90 Prozent), sondern ausschließlich durch Dienstleistungen wie Support-Abos (Marge 60 bis 80 Prozent), Schulung (Marge 40 Prozent) und Beratung (Marge 20 Prozent); ein gewisser Nachteil im hart umkämpften Software-Markt.

Somit haben die Firma MariaDB und weitere Dienstleister im MySQL-Umfeld heute nur die Möglichkeit, mit Support-Subskriptionen, Schulung und Beratung Geld zu verdienen. Es ist daher klar, dass diese Firmen sich, wenn möglich, auf das lukrativste Geschäft konzentrieren,

die Support-Subskriptionen. Mit dem dadurch erwirtschafteten Geld müssen nicht nur die Mitarbeiter entlohnt, sondern es muss zusätzlich auch noch in die Weiterentwicklung des Produkts investiert werden. Da das Lizenzgeschäft entfällt, muss entsprechend effizienter mit den Ressourcen gehaushaltet werden, um mit den Mitbewerbern, die nicht aus der Open-Source-Szene kommen, mithalten zu können.

Neben den beiden wichtigsten Branchen MySQL und MariaDB gibt es noch zahlreiche weitere Branchen und Forks. Viele davon sind allerdings nicht mehr relevant oder gar existent und daher meist nur für Software-Archäologen von Interesse. Eine spannende Entwicklung ist zudem, dass einige große Web-Firmen (Facebook, Google etc.) einen eigenen MySQL-Branch für ihre eigenen Bedürfnisse pflegen. Auch im fernen Asien scheint es Firmen zu geben, die ihre eigenen MySQL-Branches haben (wie AliSQL), die wir hier im Westen, wenn überhaupt, höchstens vom Namen her kennen. Spannend wäre noch zu wissen, von was wir noch, sprach- und schriftbedingt, keine Kenntnis haben.

Einige dieser Entwicklungen von Fremdfirmen fließen auch wieder zurück zu den eigentlichen Software-Entwicklungshäusern (Upstream) mit dem Wunsch, dass diese Fremd-Entwicklungen dann dort in den Softwarehäusern gepflegt und weiterentwickelt werden. Dies nennen einige Exponenten der MySQL-Szene dann auch „Community Contributions“. InnoDB und BDB waren die ersten namhaften Beiträge der Community. Heute sind es vor allem Firmen wie Facebook, Google, Booking.com, Alibaba etc., die solche Beiträge leisten.

Das Problem dabei ist, dass diese Firmen ihr Hauptgeschäft nicht im Entwickeln von Software sehen und daher ihre Beiträge gerne an die Software-Hersteller abgeben möchten. Diese sehen sich jedoch mit dem zusätzlichen Wartungsaufwand konfrontiert, der aber nur für wenige Endnutzer wirklich relevant ist. Die meisten MySQL-Nutzer spielen nicht in derselben Liga wie Facebook und Alibaba – also keine Einkünfte durch das Lizenzgeschäft, aber zusätzlichen Aufwand durch die Pflege von Fremd-Entwicklungen. Das geht voraussichtlich auf die Dauer nicht auf.

Das Open-Core-Modell

MySQL verfügt über zahlreiche APIs (Schnittstellen) für Erweiterungen. Diese bieten die Möglichkeit, MySQL unabhängig von der Kernfunktionalität modular zu erweitern. Ein Beispiel ist das Storage-Engine-API, das eine definierte Schnittstelle für das Anbinden weiterer Storage-Engines anbietet. Storage-Engine-Plug-ins, die dieses API nutzen, sind InnoDB, RocksDB, Connect, Federated-X etc. Daneben umfasst MySQL auch APIs für Volltext-Suche, Replikation, Auditing, Authentifizierung, Passwortvalidierung, Query Rewriting, Keyring-Anbindung etc. Für die meisten davon sind auch bereits Plug-ins verfügbar.

Dies führt zu einem neuen Geschäftsmodell in der Open-Source-Welt: dem Open-Core-Modell. Ein bereits zu MySQL-Zeiten praktizierter Ansatz, Nutzer der kostenlosen MySQL-Datenbank mit subskriptionspflichtigen, kommerziellen Erweiterungen zum Abschluss eines lukrativen Support-Abos zu motivieren. Das

daraus resultierende Produkt nennt sich MySQL Enterprise Server.

Erweiterungen für solche Enterprise-Kunden sind MySQL Enterprise Monitor für das Überwachen der MySQL-Datenbank; das MySQL Enterprise Backup für das Sichern und Zurückspielen von großen Datenbeständen; MySQL Enterprise Security mit Encryption, Audit, SQL Firewall und dem MySQL Enterprise Thread Pool für Anwendungen, die Tausende von Verbindungen gleichzeitig ermöglichen müssen. Die Mitbewerber versuchen natürlich, die spannenden Erweiterungen nachzubauen, um die Gunst der Kunden nicht zu verlieren.

Einen Schritt weiter hat sich MariaDB vor einigen Monaten gewagt. Das selbstentwickelte Router- und Proxy-Produkt MaxScale wurde in der neuesten Version unter die Business Source License (BSL) gestellt. Diese besagt, dass Installationen zu Testzwecken und für bis zu drei Installationen kostenlos sind. Größere Installationen sind kostenpflichtig und um den Gedanken von Open Source nicht zu vernachlässigen,

fallen diese BSL-Produkte nach einer Dauer von drei Jahren automatisch unter die GPL-v2-Lizenz. Der Gedanke dahinter ist: Der Kunde hat Investitionsschutz dank Open Source. Nach spätestens drei Jahren ist er nicht mehr an den Software-Hersteller gebunden und dieser hat die Möglichkeit, durch Lizenzgeschäfte mehr Umsatz zu generieren, um so mehr in die Software-Entwicklung zu investieren oder die Gewinne zu optimieren.

Ein Plug-in, das in den letzten sieben Jahren mehr und mehr Zuspruch gefunden hat, ist ein Produkt namens „Galera“, von einer kleinen, finnischen Firma namens „Codership“. Es implementiert eine synchrone Replikation (genauer gesagt eine virtuell synchrone Replikation) zwischen verschiedenen MySQL-Servern. So entsteht das erste Mal die Möglichkeit, parallel auf verschiedene MySQL-Knoten gleichzeitig zu schreiben und von ihnen zu lesen. Dieses Produkt, „Galera Cluster für MySQL“ genannt, ist insbesondere heute, wo zunehmend 7 x 24 Stunden Verfüg-

Flexible SLAs. Weil Ihre IT einmalig ist.



dbi FlexService für Datenbanken und Middleware: Ihr flexibles, kosteneffizientes und ISO 20000-zertifiziertes Service Management. Profitieren Sie insbesondere von unserem dedizierten Service Desk für validierte Systeme im Pharma-Bereich.

Phone +41 32 422 96 00 · Basel · Nyon · Zürich · dbi-services.com



Infrastructure at your Service.



barkeit verlangt wird, genau das, was sich viele Administratoren wünschen. Keine Nacht- und Wochenend-Arbeiten mehr. Wartungen können im laufenden Betrieb während der regulären Arbeitszeiten unter Last ausgeführt werden.

Fazit

MySQL hat eine bewegte Vergangenheit. Dank ihrer flexiblen Architektur und Erweiterbarkeit wird MySQL immer mehr und neue Funktionalität relativ schnell und einfach anbieten können. Dank der Open-Source-Lizenz GPL v2 dürfen sozusagen alle mitmischen, die dazu Lust und Laune haben. Stand heute gibt es zwei namhafte Player: MySQL von Oracle und MariaDB. Bei so vielen Freiheitsgraden stellt sich natürlich die Frage: Wo geht die Reise hin? Voraussichtlich wird auch hier der zweite Hauptsatz der Thermodynamik zum Tragen kommen. Salopp gesagt: Die Natur strebt das größtmögliche Chaos an ...

Bereits jetzt ist eine zunehmende Inkompatibilität zwischen den beiden Branchen MySQL und MariaDB erkennbar. Dies hat man zuerst bei der Replikation in der Version 5.5 gesehen. Später sind Änderungen am Optimizer und Parser hinzugekommen. Zahlreiche Erweiterung bei beiden Produkten tragen ebenfalls zur zunehmenden Inkompatibilität bei.

Man sieht dies auch an den verwendeten Formulierungen: Früher hieß es „Drop-in Replacement“, heute nur noch „File- oder Schnittstellen-kompatibel“ (was genau damit gemeint ist, wird nicht näher spezifiziert), und man findet zunehmend Hinweise zu Migrationsvorgängen. In naher Zukunft, also bei MariaDB 10.2 oder 10.3 und MySQL 8.0, muss damit gerechnet werden, dass diese beiden Produkte nicht mehr ohne Weiteres durch das jeweils andere substituiert werden können. Es wird dann, wie bei anderen Datenbank-Produkten auch, einen Migrationsprozess brauchen, der sich über die Zeit zunehmend aufwändiger gestalten wird.

Wer weiß heute noch, dass das Produkt Microsoft SQL Server auf dem Code von Sybase ASE Server basiert? Und wer würde heute noch wagen zu behaupten, dass diese beiden Datenbank-Systeme noch kompatibel zueinander sind? Genau diese Entwicklung steht uns bei MySQL und MariaDB ebenfalls bevor. Streng dem Marketing-Leitsatz folgend: „Differenziere dich von deinem Mitbewerber ...“



Oliver Sennhauser
oliver.sennhauser@doag.org

Oracle sagt: Fachabteilungen und IT-Teams müssen besser zusammenarbeiten

Laut der Oracle-Studie „IaaS: Building a business that innovates“ verhindern in vielen Betrieben bestehende Prozesse und Strukturen eine reibungslose interne Zusammenarbeit. Die Studie hat untersucht, wie Unternehmen in der EMEA-Region das Thema „Cloud Computing“ angehen und welche organisatorischen Faktoren signifikanten Veränderungen und Innovationen im Wege stehen. Befragt wurden 1.200 IT-Entscheider in mittleren bis großen Unternehmen. Die Ergebnisse:

- 65 Prozent der Befragten in Deutschland (62 Prozent in EMEA) sind der Ansicht, dass Fachabteilungen und IT-Teams zusammenarbeiten müssen, damit ein Enterprise-Cloud-Modell erfolgreich sein kann
- 45 Prozent (34 Prozent in EMEA) geben an, dass Schatten-IT (bei der Fachbereiche Technologien ohne Wissen der IT-Abteilung einkaufen) einen integrierten Cloud-Ansatz erheblich erschwert

- 49 Prozent (41 in EMEA) sagen, dass mangelndes Verständnis der Fachabteilungen für die Notwendigkeit von integrierten Cloud-Ressourcen eine der aktuellen IT-Infrastruktur-Herausforderungen für den Cloud-Einsatz ist.

Im aktuellen Plattform-Modell der Unternehmen gibt es einiges zu optimieren. Als besonders ineffizient nennen die Befragten die folgenden Bereiche: die Integration von verwendeten Services (50 Prozent in Deutschland und EMEA), die Bereitstellungszeit (53 Prozent in Deutschland, 49 Prozent in EMEA) und die schlechte Applikationsintegration (44 Prozent in Deutschland, 47 Prozent in EMEA). Alle drei Faktoren sind entscheidend für effektive und schnelle Innovationen.

Ein integrierter Cloud-Ansatz kann die Kollaboration verbessern. Dem stehen jedoch einige Hürden im Weg: 45 Prozent der Befragten in Deutschland (34 Prozent in EMEA) sehen das Management von

Schatten-IT als Hindernis. 31 Prozent (34 in EMEA) kämpfen mit Infrastrukturen, die nicht zusammenpassen. Auch größere Sicherheitsrisiken (36 Prozent in Deutschland, 34 Prozent in EMEA), erhöhte Kosten (30 Prozent in Deutschland, 32 Prozent in EMEA) und der ROI-Nachweis (26 Prozent in Deutschland, 31 Prozent in EMEA) werden als Hürden genannt.

Mark Borgmann, Cloud Sales Leader Oracle Deutschland kommentiert: „Verantwortlich für diese Probleme ist unter anderem eine Infrastruktur, die festgefahren, konfus und komplex ist. Das führt zu einer zersiedelten Organisation, die nicht schnell genug reagieren kann. Schlecht integrierte Systeme und abgegrenzte Fachabteilungen bremsen die Innovationsfähigkeit. Mit Infrastructure-as-a-Service (IaaS) können Unternehmen die Basis für echte Kollaboration schaffen. Sie ist das Fundament für Innovation, indem sie Ressourcen gemeinsam verfügbar macht und Silos auflöst.“



MySQL-Auswahl gegen die Distribution

Jörg Brüche, FromDual GmbH

Von MySQL gibt es verschiedene Varianten (Oracle Community, Oracle Enterprise, MariaDB, Percona) jeweils in mehreren Versionen, und die Linux-Distributionen liefern ihre eigenen Builds auf Basis einer dieser Varianten. Wenn der System- oder Datenbank-Administrator eine andere Variante als die der Distribution installieren will, gibt es häufig Abhängigkeitskonflikte. Dieser Artikel beschreibt, warum das so ist, wie man diese Konflikte analysiert und sie dann löst, ohne das Paket-Management zu umgehen. Im Ergebnis wird so genau die gewünschte Variante installiert, und alle Abhängigkeiten sind erfüllt.

Der Begriff „MySQL“ wird mit zwei verschiedenen Bedeutungen benutzt: Zum einen bezeichnet er ein ganz bestimmtes Produkt der Firma Oracle, das sie in zwei Varianten anbietet („Community Edition“ und „Enterprise Edition“), zum anderen alle Produkte, die von den GPL-Quellen dieses Produkts abgeleitet wurden. In diesem Artikel wird MySQL im zweiten Sinn benutzt; bei den abgeleiteten Produkten geht es um „MariaDB“ und „Percona Server“. Die Na-

men gehören markenrechtlich natürlich den jeweiligen Anbietern, das hat aber die allgemeine Nutzung (und begriffliche Ausweitung) von „MySQL“ nicht verhindert.

Wer die Wahl hat ...

Wie bereits angedeutet, stehen als MySQL-Datenbanksystem etliche Varianten zur Wahl:

- MySQL Enterprise Edition (Oracle)
- MySQL Community Edition (Oracle)
- Percona Server (Percona)
- Percona XtraDB Cluster (Percona)
- MariaDB Server (MariaDB)
- Galera Cluster (Codership)

Meist sind zwei Releases als „allgemein verfügbar“ deklariert, also bei Oracle derzeit MySQL 5.6 und MySQL 5.7. Nähere Informationen zu den Versionen, ihren

Features etc. gibt es bei den jeweiligen Anbietern.

In diesem Artikel geht es um MySQL auf Linux. Hier wird die Vielfalt noch dadurch vergrößert, dass die meisten Distributionen selbst MySQL-Pakete bauen und ausliefern, der Anwender beziehungsweise Administrator ein MySQL-Paket also nicht separat vom Anbieter (Oracle, Percona oder MariaDB) holen muss. Typischerweise nutzt die Distribution diese Pakete dann auch, um anderen Komponenten ein Datenbank-System verfügbar zu machen. So gibt es Abhängigkeiten, durch die das MySQL-Paket der Distribution zwingend installiert wird.

Jede Distribution entscheidet sich vor der Herausgabe eines Release für eine MySQL-Variante und nimmt davon das Release, das als „allgemein verfügbar“ erklärt ist. Innerhalb der Support-Zeit für das Distributions-Release wird dann die MySQL-Version auch aktualisiert, es gibt jedoch keinen Release-Wechsel. Das gilt selbst dann, wenn der Anbieter („Upstream“) dieses MySQL-Release nicht mehr unterstützt.

Ein Beispiel dafür ist zurzeit Red Hat Enterprise Linux 6 (RHEL, einschließlich kompatibler Distributionen wie CentOS, Scientific Linux und Oracle Linux). RHEL 6.0 wurde am 10. November 2010 veröffentlicht, das Release soll bis zum 30. November 2020 Support erhalten (Angaben laut Wikipedia). Damals war MySQL 5.1 das aktuelle Release (5.5 wurde erst im Dezember 2010 als „GA“ erklärt), und so enthält RHEL 6 weiterhin MySQL 5.1 – während Oracle den Support dafür im Dezember 2013 eingestellt hat. Hier zeigen sich die unterschiedlichen Interessen von Distributionen und Software-Herstellern:

- Eine Distribution liefert einen Satz von aufeinander abgestimmten Paketen, die auf gemeinsamen Basis-Funktionen aufbauen und für die jeweils höhere Schicht ihre Leistung durch konstante Schnittstellen (APIs) verfügbar machen. Das Ziel einer Distribution ist, die insgesamt angebotenen Funktionen über die gesamte Lebensdauer für den Anwender stabil bereitzustellen.
- Ein Software-Hersteller liefert eine bestimmte Komponente (wie MySQL), die gewisse Basis-Funktionen (System Calls, C-Bibliothek) voraussetzt und so ihre Leistung für andere Komponenten (Anwendungen) anbietet. Ziel des Herstellers ist, seine Komponente

stets aktuell zu halten, sodass der Anwender möglichst schnell von Verbesserungen profitiert.

Während also für die Distribution die stabile Nutzbarkeit (zu der natürlich auch Fehlerbehebungen, also Upgrades, gehören) aller Komponenten gemeinsam an erster Stelle steht, ist es für den Hersteller die Aktualität seiner Komponente (unter der Einschränkung einer für den Anwender hinreichend langen Lebensdauer).

... hat die Qual

„Qual“ ist hier die „Qual der Entscheidung“: Selbstverständlich bietet jede MySQL-Variante und -Version die üblichen Grundfunktionen, ist also für viele Zwecke ausreichend. Ebenso selbstverständlich führt aber die Weiterentwicklung der Anbieter zu Änderungen (meist: Verbesserungen) durch funktionale Erweiterungen und Fehlerbehebungen, die für manche Zwecke entscheidend sind. Als Beispiele seien genannt:

- Seit MySQL 5.6 unterstützt auch InnoDB die Volltext-Suche; Anwendungen, für die dieses Feature wesentlich ist, sind also nicht mehr zur Benutzung von MyISAM gezwungen
- Seit MySQL 5.7 unterstützt auch InnoDB die speziellen Indizes für geografische Objekte und die dazugehörigen Funktionen
- MySQL 5.7 bringt weitere neue Features wie den Datentyp „JSON“ mit dazu passenden Funktionen sowie „generierte Spalten“
- Nur die Enterprise Edition von Oracle enthält die zusätzlichen Tools wie „Enterprise Monitor“ und „Audit Plug-in“
- Percona hat MySQL um „TokuDB“ als weiteren Table-Handler erweitert und weitere Performance- sowie Diagnose-Features implementiert
- Wer für Hochverfügbarkeit oder Lastverteilung eine produktionsreife Cluster-Lösung haben möchte, muss eine Variante mit Galera-Cluster nehmen (von Codership auf Basis Oracle-Community, von MariaDB oder von Percona)
- Der Hersteller einer Anwendung kann eine bestimmte MySQL-Version vorschreiben, und eine Anwendung kann

eine bestimmte Fehlerbehebung benötigen

- Das Auslaufen des Anbieter-Supports kann ein Kriterium für den Einsatz einer neueren Version sein
- Auch Vorgaben aus der Firmenpolitik oder Einschätzungen der Anbieter-Qualität können die Wahl der Variante beeinflussen

Pakete und Lieferdienst

Üblicherweise gibt es MySQL-Versionen für Linux in zwei unterschiedlichen Formaten: als „tarball“ („tar.gz“) und als Distributions-spezifisches Paket („rpm“ beziehungsweise „deb“). Für tarball kann man den Installationsort frei wählen (muss dann gegebenenfalls die Variable „\$PATH“ anpassen), für ein RPM- oder DEB-Paket wird er beim Paketbau festgelegt.

Die meisten Installationen benutzen Pakete, weil für sie die Anpassung an die Distribution sichergestellt ist und weil die Installation und das Upgrade ohne manuelle Eingriffe erfolgen. Voraussetzung für ein automatisiertes Upgrade ist, dass das jeweilige Repository im System eingerichtet ist, sodass das Upgrade-Tool („yum“, „zypper“, „apt-get“, „synaptic“ etc.) die Existenz einer neueren Version feststellen und die Pakete dann auf die Maschine holen kann. Pakete bringen allerdings die Einschränkung, dass nur eine Version installiert sein kann, sonst käme es ja zu Namenskonflikten – es kann auf einer Maschine nur ein „/usr/bin/mysql“ geben.

Sofern nicht ungewöhnliche Gründe die parallele Installation mehrerer MySQL-Versionen erzwingen, ist die Entscheidung für Pakete die bessere Wahl. Wenn also eine bestimmte MySQL-Variante in einer bestimmten Version installiert werden soll, dann muss dafür ein etwa bereits installiertes MySQL-Paket entfernt werden. Das kann schwierig werden, wenn andere installierte Pakete von diesem MySQL-Paket abhängen: Wird es entfernt, dann werden diese Abhängigkeiten verletzt. Die Tools zum Paket-Management („rpm“ beziehungsweise „dpkg“) verhindern das, wenn der Administrator es nicht durch eine „force“-Option explizit verlangt – dann ist jedoch die Konsistenz des Gesamtsystems nicht mehr gesichert.

Solange im System nur die Pakete der Distribution benutzt werden, ist das alles kein Problem: Die Distribution stellt

die Upgrade-Möglichkeit sicher, und durch ihre Festlegung auf ein bestimmtes MySQL-Release ist auch das korrekte API für die installierten Anwendungen garantiert. Schwieriger wird es, wenn die Pakete der Distribution durch Pakete eines anderen Anbieters ersetzt werden sollen, typischerweise also durch solche von Oracle, MariaDB, Percona oder Codership: Meistens führt ein solcher Versuch zu einer Fehlermeldung we-

gen Datei-Konflikten oder unerfüllten Abhängigkeiten und scheitert dann. Nachfolgend ist das näher beschrieben, analysiert und dann sauber umgangen.

Wie ein Upgrade scheitert

Das Beispiel in *Listing 1* zeigt den Versuch, auf einem System mit CentOS 7

die Community Edition von MySQL 5.7 (Oracle) zu installieren. Zunächst wird der Ausgangszustand bestimmt und im CentOS-Repository nach MySQL-Paketen gesucht.

Man sieht, dass CentOS 7 keine Pakete auf Basis von Oracles MySQL liefert, sondern MariaDB 5.5 enthält. Bei Oracle endete der Support für MySQL 5.5 im Dezember 2015, bei MariaDB ist das für April 2017 angekündigt. Die gewünschten Pakete müssen also direkt von Oracle geholt werden. Das könnte über „yum“ erfolgen, aber mit einer Trennung zwischen Download und lokaler Installation (mit „rpm“) gibt es bessere Diagnose-Möglichkeiten. Oracle bietet einen „tarball“ aller RPM-Pakete („bundle“) an, durch Download und Auspacken erhält man diese Dateien (siehe *Listing 2*, weitere sind ausgelassen). Ein Versuch, das Client- und das Server-Paket zu installieren, scheitert gründlich (siehe *Listing 3*).

Der erste Schritt ist natürlich, die fehlenden System-Pakete aus dem CentOS-Repository zu installieren; das ist einfach und gelingt mit „demo\$ sudo yum install libaio net-tools“. Nun wird erneut versucht, die MySQL-Pakete zu installieren, außer „client“ und „server“ auch die benötigten Pakete „libs“ und „common“ (siehe *Listing 4*).

Diese Fehlermeldung ist durch ihre schlechte Formulierung nur schwer verständlich. Gemeint ist: Die Metadaten von „mysql-community-libs“ verlangen, „mariadb-libs“ zu ersetzen, aber die „install“-

```
demo$ cat /etc/redhat-release
CentOS Linux release 7.2.1511 (Core)
demo$ rpm -qa | egrep -i 'mysql|maria'
mariadb-libs-5.5.35-3.el7.x86_64
demo$ yum info 'mysql*'
...
Verfügbare Pakete
Name      : MySQL-python
...
Name      : mysql-connector-java
...
Name      : mysql-connector-odbc
...
```

Listing 1

```
demo$ ls -l /Download/
mysql-5.7.14-1.el7.x86_64.rpm-bundle.tar
mysql-community-client-5.7.14-1.el7.x86_64.rpm
mysql-community-common-5.7.14-1.el7.x86_64.rpm
...
mysql-community-libs-5.7.14-1.el7.x86_64.rpm
mysql-community-libs-compat-5.7.14-1.el7.x86_64.rpm
mysql-community-server-5.7.14-1.el7.x86_64.rpm
...
```

Listing 2

```
demo$ sudo rpm -i /Download/mysql-community-{client,server}-5.7.14-1.el7.x86_64.rpm
Warnung: ... NOKEY
Fehler: Fehlgeschlagene Abhängigkeiten:
mysql-community-libs(x86-64) >= 5.7.9 wird benötigt von mysql-community-client-5.7.14-1.el7.x86_64
libaio.so.1()(64bit) wird benötigt von mysql-community-server-5.7.14-1.el7.x86_64
libaio.so.1(LIBAIO_0.1)(64bit) wird benötigt von mysql-community-server-5.7.14-1.el7.x86_64
libaio.so.1(LIBAIO_0.4)(64bit) wird benötigt von mysql-community-server-5.7.14-1.el7.x86_64
mysql-community-common(x86-64) = 5.7.14-1.el7 wird benötigt von mysql-community-server-5.7.14-1.el7.x86_64
net-tools wird benötigt von mysql-community-server-5.7.14-1.el7.x86_64
```

Listing 3

```
demo$ sudo rpm -i /Download/mysql-community-{client,common,libs,server}-5.7.14-1.el7.x86_64.rpm
Warnung: ... NOKEY
Fehler: Fehlgeschlagene Abhängigkeiten:
mariadb-libs wird ersetzt durch mysql-community-libs-5.7.14-1.el7.x86_64
```

Listing 4


```
demo$ sudo rpm -U /Download/mysql-community-(client,common,libs,server)-5.7.14-1.el7.x86_64.rpm
Warnung: ... NOKEY
Fehler: Fehlgeschlagende Abhängigkeiten:
libmysqlclient.so.18() (64bit) wird benötigt von (installiert) postfix-2:2.10.1-6.el7.x86_64
libmysqlclient.so.18(libmysqlclient_18) (64bit) wird benötigt von (installiert) postfix-2:2.10.1-6.el7.x86_64
```

Listing 5

Aktion von „rpm“ (Option „-i“) tut das nicht: Nur ein „update“ (Option „-U“) führt eine solche Verdrängung durch. Also ein neuer Versuch, mit „update“ statt „install“ (siehe Listing 5).

Hier ist nun das inhaltliche Problem sichtbar: Das Paket „postfix“ (das vom Upgrade nicht direkt erfasst wird) benötigt die Datei „libmysqlclient.so.18“. Wenn man ahnt, dass diese Bibliothek vom Upgrade erfasst wird, kann man diese Vermutung sehr einfach prüfen (siehe Listing 6).

Ohne diese Ahnung muss man anders forschen (siehe Listing 7). Beide Wege zeigen: Die Bibliothek ist im zu verdrängenden Paket enthalten, würde also gelöscht werden; dieser Bruch der Abhängigkeiten ist nicht zulässig.

Exkurs: Unterpakete und Abhängigkeiten

Für das bessere Verständnis der folgenden Abschnitte ist es hilfreich, die Einteilung der MySQL-Komponenten in Unterpakete zu kennen. Die Vorstellung, es gebe einfach ein Paket „MySQL“, stimmt nicht – sowohl bei den RPM- als auch bei den DEB-Paketen wird zwischen dem Server, den mitgelieferten Client-Programmen und den Bibliotheken für Anwendungen unterschieden. Die genaue Einteilung entscheidet jeder Paket-Ersteller selbst.

Tabelle 1 zeigt, wie diese etwa bei Red Hat 6 mit MySQL 5.1 war.

Das Client-Paket erkennt man also nicht direkt am Namen, und sowohl das Server- als auch das Client-Paket hängen vom Bibliotheks-Paket ab. Oracle hat die Einteilung verfeinert und „common“ für gemeinsame Dateien eingeführt, benennt das Client-Paket ausdrücklich und erzwingt die Installation der Client-Programme auch beim Server (siehe Tabelle 2). „ED“ steht für „Edition“, also „community“ beziehungsweise „commercial“ im Paketnamen, die Einteilung ist bei beiden gleich.

```
demo$ rpm -ql mariadb-libs | fgrep libmysqlclient
/usr/lib64/mysql/libmysqlclient.so.18
/usr/lib64/mysql/libmysqlclient.so.18.0.0
```

Listing 6

```
demo$ find /usr/lib* -name libmysqlclient.so.18 2>/dev/null
/usr/lib64/mysql/libmysqlclient.so.18
demo$ rpm -qf /usr/lib64/mysql/libmysqlclient.so.18
mariadb-libs-5.5.37-1.el7_0.x86_64
```

Listing 7

Server	Client	Anwendung
mysql-server		
	mysql	
mysql-libs	mysql-libs	mysql-libs

Tabelle 1

Server	Client	Anwendung
mysql-ED-server		
mysql-ED-client	mysql-ED-client	
mysql-ED-libs	mysql-ED-libs	mysql-ED-libs
mysql-ED-common	mysql-ED-common	mysql-ED-common

Tabelle 2

```
demo$ sudo rpm -e mariadb-libs
Fehler: Fehlgeschlagende Abhängigkeiten:
libmysqlclient.so.18() (64bit) wird benötigt von (installiert) postfix-2:2.10.1-6.el7.x86_64
libmysqlclient.so.18(libmysqlclient_18) (64bit) wird benötigt von (installiert) postfix-2:2.10.1-6.el7.x86_64
```

Listing 8

Ursachenforschung

Die bisherigen Erkenntnisse sind:

- Sowohl das Server- als auch das Client-Paket lassen sich nicht ohne das Libs-Paket installieren
- Das Paket „mysql-community-libs“ kollidiert mit „mariadb-libs“ und will es verdrängen. Das beruht auf Gegensei-

tigkeit, „mariadb-libs“ würde ein installiertes „mysql-community-libs“ ebenfalls verdrängen.

- Diese Verdrängung scheitert, weil sie die Anforderungen des installierten „postfix“-Pakets verletzen würde

Den dritten Punkt sollte man sich noch einmal genauer anschauen (siehe Listing 8). Diese Fehlermeldung sieht etwas sel-

sam aus, weil sie „mariadb-libs“ nicht erwähnt. Was ist die genaue Abhängigkeit? *Listing 9* zeigt, was „rpm“ antwortet. Wenn

das Paket „mariadb-libs“ nicht verlangt wird, was ist dann die Abhängigkeit? Auch das sagt „rpm“ in *Listing 10*.

```
demo$ rpm -q --whatrequires mariadb-libs
Kein Paket benötigt mariadb-libs
```

Listing 9

```
demo$ rpm -q --requires postfix
...
libmysqlclient.so.18() (64bit)
libmysqlclient.so.18(libmysqlclient_18) (64bit)
...
```

Listing 10

```
demo$ rpm -q --provides mariadb-libs
...
libmysqlclient.so.18() (64bit)
libmysqlclient.so.18(libmysqlclient_18) (64bit)
...
```

Listing 11

Die Abhängigkeit besteht also nicht gegenüber dem Paket, sondern gegenüber der enthaltenen Bibliothek. Unabhängig davon, ob der Paket-Ersteller das angibt, erzeugt „rpm“ standardmäßig für jede im Paket enthaltene Bibliothek auch „provides“-Angaben (*siehe Listing 11*). Die Lösung besteht also darin, die Bibliothek „libmysqlclient.so.18“ aus einem anderen Paket zu installieren, sodass dann „mariadb-libs“ zugunsten der Oracle-MySQL-Pakete verdrängt werden kann.

Die saubere Lösung

Unter den ausgepackten Oracle-MySQL-RPMs waren auch weitere, die bisher nicht betrachtet wurden (*siehe Listing 12*). „rpm“ liefert die Auskunft darüber, was „libs-compat“ und „libs“ zur Verfügung stellen und wie sie sich dabei unterscheiden (*siehe Listing 13*).

Während also das „libs“-RPM die Version 20 der Client-Bibliothek enthält, ist in „libs-

Alles, was die SAP-COMMUNITY wissen muss, finden Sie monatlich im E-3 MAGAZIN.
Ihr WISSENSVORSPRUNG im Web, auf iOS und Android sowie PDF und Print:
e-3.de/abo

Wer nichts weiß,
muss alles glauben!

Marie von Ebner-Eschenbach



SAP® ist eine eingetragene Marke der SAP AG in Deutschland und in den anderen Ländern weltweit.

www.e-3.de

compat“ noch die Version 18. Das ist genau der Zweck eines solchen „compatibility“-Pakets: Nach dem Wechsel auf neue Software-Versionen (hier: von MySQL) soll den anderen, benutzenden Programmen noch ein Stand angeboten werden, der mit ihrer Übersetzung kompatibel ist. Das Postfix-Paket von CentOS 7 ist ja übersetzt worden für die Client-Bibliothek aus MySQL (genauer: MariaDB) 5.5, die trägt als „shared object“ die Version 18.

Anders formuliert: Das „libs-compat“-Paket von MySQL 5.7 für CentOS 7 enthält diejenige „libmysqlclient“, die mit dem Original von CentOS 7 kompatibel ist, also die aus MySQL 5.5. Es gibt auch ein „libs-compat“-Paket von MySQL 5.7 für CentOS 6, das enthält die „libmysqlclient“ aus MySQL 5.1, weil CentOS 6 noch MySQL 5.1 benutzt.

Ein „libs-compat“-RPM hat seinerseits wieder Abhängigkeiten – aber nicht zur alten Version 5.5 (die soll ja durch das Upgrade ersetzt werden), sondern zur aktuellen Version 5.7 (siehe Listing 14). „libs-compat“ muss also zusammen mit „libs“ (und darum mit „common“) installiert werden und die Aktion muss „upgrade“ sein, um „mariadb-libs“ zu verdrängen (siehe Listing 15).

Diese Verdrängung ist erfolgt, „mariadb-libs“ ist durch „mysql-community-libs-compat“ ersetzt. Jetzt können das Server- und das Client-Paket installiert werden (siehe Listing 16). Bei Bedarf können natürlich auch weitere Pakete installiert werden, etwa die Tests und die Header für Software-Entwicklung.

Fazit

- Distributionen und Software-Hersteller liefern unterschiedliche Pakete
- Die Hersteller-Pakete sind neuer als die Distributions-Pakete und bieten die größere Auswahl
- Wenn schon ein Distributions-Paket installiert ist, kann die Installation eines Hersteller-Pakets seine Verdrängung erfordern. Bei „rpm“ und „yum“ ist dafür eine „Upgrade“-Aktion nötig
- Falls das zu verdrängende Paket (im Beispiel „mariadb-libs“) von einem anderen Paket (im Beispiel „postfix“) benötigt wird, ist diese Abhängigkeit zu analysieren und es muss ein passender Ersatz gefunden werden (im Beispiel „mysql-libs-compat“).

```
demo$ ls -l /Download/
mysql-5.7.14-1.el7.x86_64.rpm-bundle.tar
mysql-community-client-5.7.14-1.el7.x86_64.rpm
mysql-community-common-5.7.14-1.el7.x86_64.rpm
...
mysql-community-libs-5.7.14-1.el7.x86_64.rpm
mysql-community-libs-compat-5.7.14-1.el7.x86_64.rpm
mysql-community-server-5.7.14-1.el7.x86_64.rpm
...
```

Listing 12

```
demo$ rpm -qp --provides mysql-community-libs-compat-5.7.14-1.el7.x86_64.rpm
...
libmysqlclient.so.18() (64bit)
...
demo$ rpm -qp --provides mysql-community-libs-5.7.14-1.el7.x86_64.rpm
...
libmysqlclient.so.20() (64bit)
...
```

Listing 13

```
demo$ rpm -qp --requires mysql-community-libs-compat-5.7.14-1.el7.x86_64.rpm
...
mysql-community-libs(x86-64) >= 5.7.9
...
demo$ rpm -qp --requires mysql-community-libs-5.7.14-1.el7.x86_64.rpm
...
mysql-community-common(x86-64) >= 5.7.9
...
```

Listing 14

```
demo$ sudo rpm -U /Download/mysql-community-{libs-compat,libs,common}-5.7.14-1.el7.x86_64.rpm
Warnung: ... NOKEY
demo$ rpm -qa | egrep -i 'mysql|maria'
mysql-community-common-5.7.14-1.el7.x86_64
mysql-community-libs-compat-5.7.14-1.el7.x86_64
mysql-community-libs-5.7.14-1.el7.x86_64
```

Listing 15

```
demo$ sudo rpm -i /Download/mysql-community-{client,server}-5.7.14-1.el7.x86_64.rpm
Warnung: ... NOKEY
demo$ rpm -qa | egrep -i 'mysql|maria'
mysql-community-common-5.7.14-1.el7.x86_64
mysql-community-libs-compat-5.7.14-1.el7.x86_64
mysql-community-server-5.7.14-1.el7.x86_64
mysql-community-libs-5.7.14-1.el7.x86_64
mysql-community-client-5.7.14-1.el7.x86_64
```

Listing 16

- Die Abhängigkeit besteht nicht unbedingt zwischen den Paketen, sondern oft zu einer enthaltenen Bibliothek („shared library“). Dabei ist auf die „so“-Version zu achten
- Häufig ist die Analyse einfacher, wenn man zunächst die Pakete herunterlädt und sie dann lokal analysiert (mit „rpm“ beziehungsweise „dpkg“), als wenn man das Repository befragt (mit „yum“ beziehungsweise Apt-Tools).

Mit dieser Analyse ist es möglich, installierte MySQL-Pakete durch solche eines ande-

ren Paketierers zu ersetzen (die natürlich für die gegebene Plattform passend erstellt sein müssen), ohne dabei die Abhängigkeiten des Paket-Managements zu übergehen oder zu brechen. Der kritische Bereich sind die Anforderungen von installierten Anwendungen, die durch Kompatibilitäts-Pakete (meist „libs-compat“) erfüllt werden.

Falls das (mangels Angebot) nicht möglich ist, ist die Installation aus „tarball“ („tar.gz“) die saubere Alternative. Dafür gibt es allerdings keine Upgrade-Unterstützung, sodass der Administrator das selbst periodisch prüfen muss.



Jörg Brühe
joerg.bruehe@fromdual.com

MySQL[®] in der Oracle Public Cloud – den Durchblick behalten

Mario Beck und Carsten Thalheimer, ORACLE Deutschland B.V. & Co. KG

Während die MySQL-Datenbank bereits seit einigen Jahren im Amazon-, Microsoft- und Google-Cloud-Angebot ist, hat sich Oracle als Eigentümer von MySQL erstaunlich lange mit einem MySQL Cloud Service Zeit gelassen und den bereits seit Juli 2016 laufenden Testbetrieb während der Oracle OpenWorld freigegeben.

Auch im Jahr 2016 nutzte die Oracle-MySQL-Gruppe wieder die OpenWorld als Plattform, um praktisch alle Produkte der MySQL Enterprise Product Suite auf den neuesten Stand zu bringen. Der Fokus der rund fünfzehn MySQL Product Announcements (siehe „<http://forums.mysql.com/list.php?3>“) lag vor allem auf der neuen Version 8.0, die wie üblich als Development Milestone Release 1 (Beta Quality) als Binary und im Source Code für gängige Plattformen zur Verfügung gestellt wurde. Die Versionsnummern MySQL 6 (Namenskonflikt mit älteren Sun-Microsystems-MySQL-Plänen) und MySQL 7 (reserviert für MySQL NDB Cluster) werden somit übersprungen.

Nicht minder interessant sind die Neuigkeiten in der Oracle Public Cloud (OPC). Das Presse-Release liest sich zunächst recht nüchtern: Es wird von einem sicheren, kosteneffektiven und unternehmerorientierten MySQL Cloud Service (CS) gesprochen, ohne allerdings auf die technischen Details einzugehen.

Die in der Cloud verwendete MySQL-Version basiert auf MySQL Enterprise 5.7.16 und ist somit 100 Prozent kompatibel zu den im Jahr 2015 veröffentlichten MySQL 5.7 Community und kommerziellen Versionen. Somit ist gewährleistet, dass die Daten sowohl in der Cloud als auch On-Premise in eigenen, lokalen Rechenzentren verwendet werden. Eine Daten-Migration

in die Cloud beziehungsweise zurück in das Rechenzentrum ist somit aus Datenbank-Sicht denkbar einfach.

Das im MySQL Cloud Service eingesetzte Release ist die kommerzielle Version von MySQL. Das bedeutet, dass Anwender über die normale MySQL-Funktionalität hinaus noch weitere Features nutzen können. Hervorzuheben sind hier vor allem die Security-Erweiterungen wie transparente Datenverschlüsselung (TDE), Auditing und die integrierte SQL-Firewall.

Das erste Aufsetzen einer MySQL-Instanz in der Oracle-Cloud geht schnell von der Hand. Auf Basis eines vorkonfigurierten und MySQL-optimierten Oracle-Compute-Images wird eine virtuelle Betriebssystem-

ORACLE CLOUD My Services doud.admin ▾

Oracle MySQL Cloud Service
Create Service

[Zurück](#) [Abbrechen](#) Service Details Bestätigung [Weiter](#) ☎

Service Details
Provide details for this Oracle MySQL Cloud Service [Enterprise Edition] instance. [Selection Summary](#)

Configuration

- * Usable Database Storage (GB) ?
- * Administration User ?
- * Administration Password ?
- * Confirm Administration Password ?
- * Database Schema Name ?
- Server Character Set ?
- Configure MySQL Enterprise Monitor ?
- * Manager User ?
- * Manager Password ?
- * Confirm Manager Password ?

Backup and Recovery Configuration

- Backup Destination ?
- * Cloud Storage Container ?
- * Cloud Storage User Name ?
- * Cloud Storage Password ?
- Create Cloud Storage Container ?

Abbildung 1: Setup einer MySQL-Cloud-Service-Instanz

Instanz mit MySQL ausgerollt. Notwendige Basisdaten wie Datenbankgröße, Zugangsdaten und CPU/Memory/Disk/Userdaten werden initial ausgewählt. Diese sind im Nachgang jederzeit anpassbar.

Der MySQL-Service kann auf Basis von „metered/messbar“ (OCPU pro Stunde) oder als „non-metered“ (OCPU per Monat) bezogen werden. „OCPU“ (Oracle Compute Unit) stellt dabei das Äquivalent zu zwei Hardware-Ausführungs-Threads mit 15 GB Arbeitsspeicher dar. Zusätzlich müssen noch der Compute Block Storage (vergleichbar zu lokaler Festplatte) und der Storage Cloud Service (für Datenbank-Backups) angegeben werden (siehe Abbildung 1).

MySQL CS ist nur zum Teil gemanagt. Wenn ein neuer Datenbank-Anwender oder ein neues Datenbank-Schema in MySQL benötigt wird, muss der Endanwender diese Konfiguration selbst durchführen. Sollte er bei administrativen Arbeiten oder beim MySQL-Betrieb Probleme haben, kann er jederzeit auf den Oracle Premier Support

zurückgreifen. Support ist in allen Abrechnungsmetriken enthalten und identisch zum Support der kommerziellen MySQL-konsultativen Support, der für proaktive Ansätze zur Diagnose von Leistungs- oder Stabilitäts-Problemen, Architekturberatung, aber auch zum Code-Review verwendet werden kann.

Die Verknüpfung des Oracle Premier/konsultativen Supports und der Automatisierung von Maintenance-Aufgaben von MySQL CS macht den MySQL-Betrieb auch ohne tiefgehendes Datenbank-Wissen sehr einfach. Vollautomatisiert sind heute bereits die Datensicherung und Wiederherstellung, tiefgehendes Monitoring sowie stark vereinfachtes Patch-Management.

Die MySQL-Betriebssystem-Umgebung

MySQL steht derzeit auf Oracle Linux 6.6 mit Unbreakable Enterprise Kernel/ext4

zur Verfügung. MySQL-Tasks werden durch Oracle Linux priorisiert. Als Storage-Backend kommt die Oracle ZFS Appliance zum Einsatz.

Der Zugriff auf Oracle Linux erfolgt auf Basis von SSH für den User „opc“. Die „opc“-Authentifizierung erfolgt auf Basis von SSH-Keys, die während der MySQL-Konfiguration automatisch oder auf Basis von existierenden SSH-Keys durchgeführt werden. Nach dem Einloggen kann man über „sudo“ sehr einfach administrative Befehle ausführen und so auf Wunsch jederzeit vollständigen Einfluss auf MySQL und auf das Betriebssystem nehmen.

Netzwerkseitig steht standardmäßig ein Netzwerk-Interface mit einer öffentlichen Adresse zur Verfügung. Dieses ist durch eine Firewall für jeden Zugriff von außerhalb (außer SSH) gesperrt. Die Konfiguration auf Basis eigener Netzwerke zur Interkommunikation mit anderen Oracle Cloud Services, privaten Netzwer-

The screenshot shows the Oracle Cloud My Services MySQL Backup Dashboard. At the top, there's a navigation bar with 'ORACLE CLOUD My Services', 'Dashboard', 'Benutzer', and 'Benachrichtigungen'. Below this, the dashboard title is 'Oracle MySQL Cloud Service / MySQLDemo'. The left sidebar contains 'Überblick' (1 Knoten) and 'Administration' (1 Patches verfügbar, 10.01.2017 06:20 Uhr UTC Most recent backup failed, 16.12.2016 06:20 Uhr UTC Last Successful Backup). The main content area has a 'Backup' tab and a notification: 'Last Activity BACKUP 10.01.2017 06:20 Uhr UTC failed. See Activity Log for Details'. A summary card shows 'Storage Cloud Volume Used', 'Backup Volume Used', and 'Backup Volume Percent Used' (12%). Below this, it lists backup schedules: 'Daily at 06:20 Uhr UTC Incremental Backups', 'Every Tuesday at 06:20 Uhr UTC Full Backups', and '16.12.2016 06:20 Uhr UTC Last Successful Backup'. The 'Available Backups' section includes a search filter and a list of backups:

Backup Date	Type	Notes
16.12.2016 06:20 Uhr UTC	Incremental, initiated by scheduled	Automatisches Backup
15.12.2016 06:20 Uhr UTC	Incremental, initiated by scheduled	Automatisches Backup
14.12.2016 06:20 Uhr UTC	Incremental, initiated by scheduled	Automatisches Backup
14.12.2016 00:18 Uhr UTC	Full, initiated by cloud.admin	Backup for applying patch ...

Abbildung 2: Das MySQL-Backup-Dashboard

ken oder VPN ist durch entsprechende Menüpunkte möglich.

Installationsdetails zum MySQL CS

Der MySQL CS verwendet die gleichen Binaries wie die kommerzielle MySQL-Version der MySQL-Homepage. Allerdings werden die CS derzeit nur etwa einmal im Quartal aktualisiert, während die Originalpakete etwa alle sechs bis acht Wochen erneuert werden. Wer MySQL von verschiedenen Linux-basierten Installationen kennt, muss sich zunächst erstmal ein wenig umgewöhnen: Die Installation

richtet sich nach dem Optimal Flexible Architecture Standard (OFA). Der für Oracle DBAs bekannte Standard spielt bei herkömmlichen MySQL-Installationen oftmals keine Rolle. Daher findet man die zentrale Konfigurationsdatei „my.cnf“ auch nicht, wie in Linux-Installationen üblich, in „/etc“, sondern in „/u01/bin/mysql-5.7.x“. Das MySQL-Basisverzeichnis („basedir“) zeigt auf „/u01/bin/mysql“ und die MySQL-Daten („datadir“) sind in „/u01/data/mysql“.

Der MySQL-Dienst läuft nicht, wie bei Linux üblich, unter dem Userkontext „mysql“, sondern dem Oracle-Public-Cloud-Standard folgend unter „oracle“. Für die Arbeit mit dem MySQL-Client muss daher

zunächst von „opc“ zum Benutzer „oracle“ gewechselt werden.

Die On-Premise-Standard-Konfiguration von MySQL ist typischerweise stark vereinfacht, um möglichst viele Anwendungsfälle und unterschiedlichste Hardware-Ressourcen abzudecken. Im Gegensatz hierzu ist die Konfiguration von MySQL CS hochgradig an die Bedürfnisse von MySQL in der Oracle Public Cloud angepasst.

Die Konfiguration von „innodb-buffer-pool-size“ ist automatisch an den verwendeten Compute Shape (OCPU/Memory) angepasst. Zusätzlich sind zahlreiche weitere Optimierungen in „my.cnf“ eingeflossen: „Number of IO threads“, „innodb-

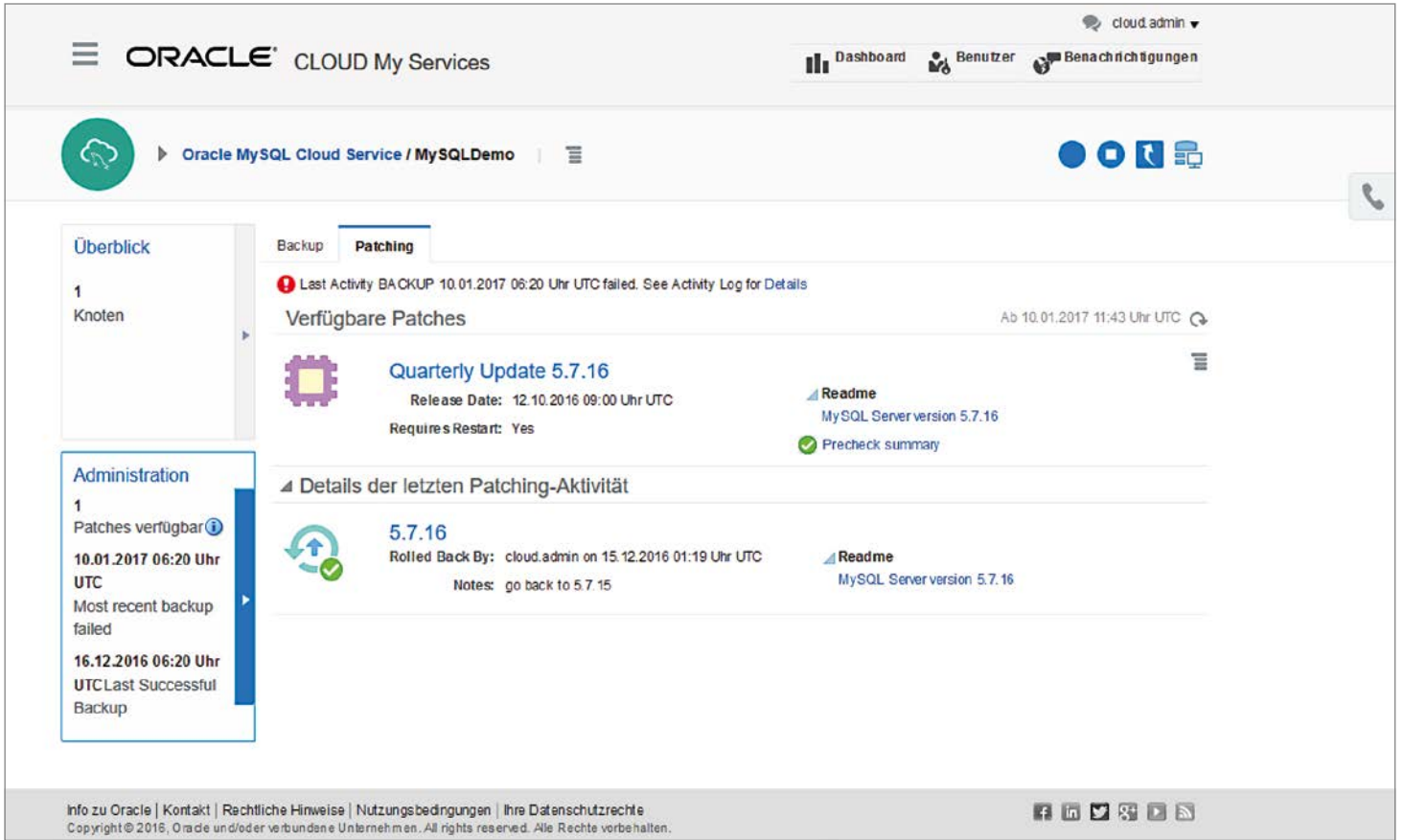


Abbildung 3: Das MySQL-Patch-Dashboard

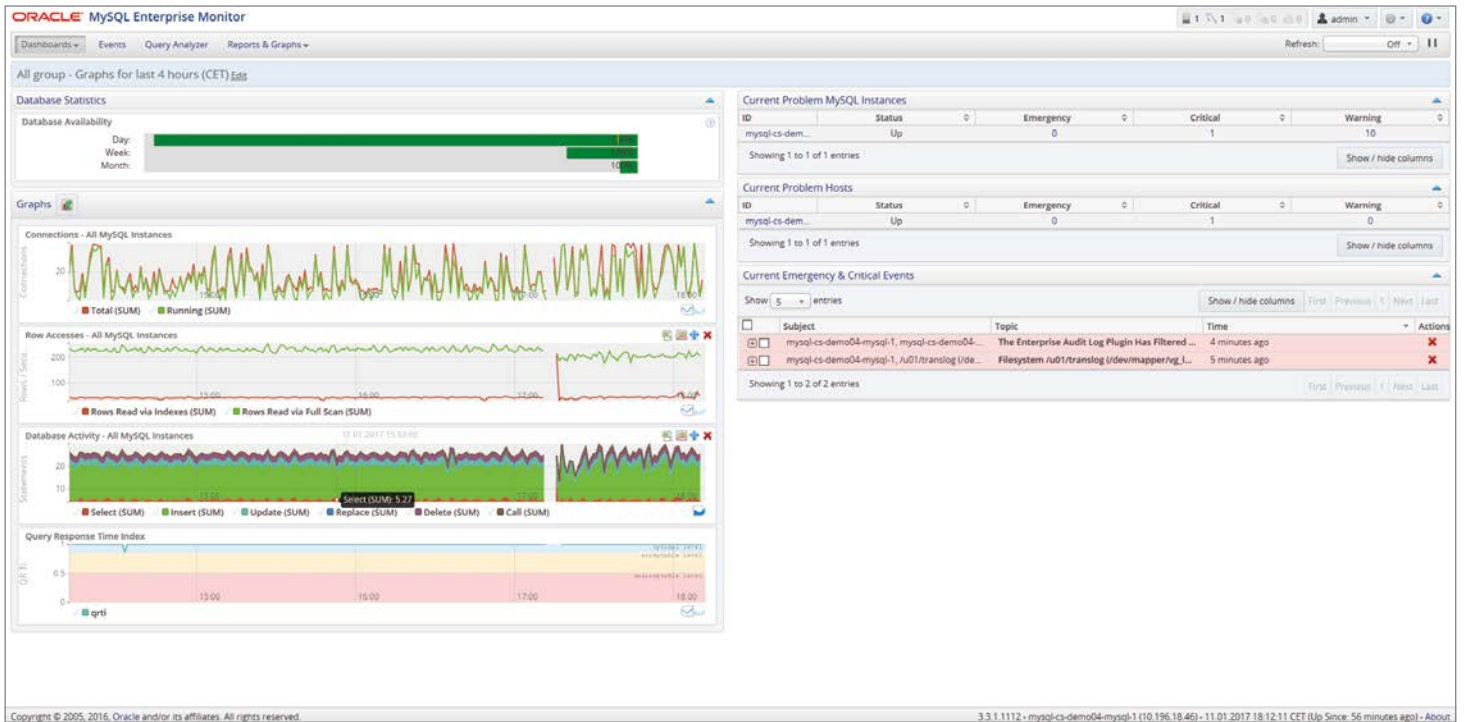


Abbildung 4: Das MySQL-Patch-Dashboard

autoinc-lock-mode=2", „innodb-flush-method=O_DIRECT_NO_FSYNC“, „Thread Pool Enabled“.

Alle Einstellungen können sehr einfach auf On-Premise-Installationen angewendet werden. Die „my.cnf“-Konfigu-

ration von MySQL CS stellt daher auch einen sehr guten Startpunkt für MySQL (auf Basis Linux) dar. Auch die Transak-

tionslogs (Binlogs) werden entgegen der MySQL-On-Premise-Standard-Konfiguration mit aufgezeichnet und stehen für eine eventuelle MySQL-Replikation und/oder für „Point in Time Recovery“-Funktionalitäten zur Verfügung. Die MySQL/Binlog-Instanz ist als MySQL-Master konfiguriert. Eine zweite beziehungsweise beliebig viele MySQL-Instanzen können daher sehr einfach an eine Standard-Instanz angebunden werden, um eine MySQL-Replikationsfarm für Skalierungszwecke oder zwecks Redundanz zu bilden.

MySQL Backup/Restore und Patching

Jeder MySQL-Service wird automatisch durch MySQL Enterprise Backup täglich gesichert. In dem vordefinierten Setup werden die Backups täglich inkrementell und wöchentlich vollständig ausgeführt. In Verbindung mit den Transaktionslogs lassen sich die Daten sekundengenau wiederherstellen.

Im Rahmen des Storage Cloud Service sind die Backups für dreißig Tage gespei-

chert. Alle Backup-Daten, Konfigurations-Dateien und MySQL-Logs können jederzeit von der Oracle Public Cloud heruntergeladen und für On-Premise-Zwecke verwendet werden. Alle Schritte für das Backup/Restore sind sehr einfach über intuitive Oberflächen des MySQL-Dashboards konfigurierbar. Die Wiederherstellung eines Backups erfolgt ebenfalls über das Dashboard. Besonders erwähnenswert ist hier die automatische Integration von Restore und Binary-Log-Recovery. Im Dashboard wählt der Administrator lediglich den wiederherzustellenden Zeitpunkt. Die Ausführung von Full-Restore, inkrementellem Restore und anschließendem Recovery anhand der Binary-Logs erfolgt voll automatisiert (siehe Abbildung 2).

Ein ähnliches Bild ergibt sich bei den MySQL-Patches: MySQL-CS-Updates werden etwa vierteljährlich im Rahmen der monatlichen OPC-Maintenance ausgerollt. Allerdings wird dabei die neue MySQL-Version nicht installiert, sondern nur in Verbindung mit einem Hinweis „One or more Patches available“ bereitgestellt. Das eigentliche Patching wird vom Administrator angestoßen.

Bevor ein Patch angewendet wird, erfolgt eine tieferegehende Prüfung der Kompatibilität und verfügbaren Ressourcen (Plattenplatz). Außerdem erfolgt ein Full-Backup, um notfalls ein Rollback durchführen zu können. Erst nach diesem Pre-Check erfolgt das Update, das eine kurze Downtime von einigen Minuten nach sich zieht. Das Einspielen des Patches kann über das gleiche Vorgehen widerrufen/zurückgesetzt werden (siehe Abbildung 3).

MySQL-CS-Monitoring

Es gibt zwei Arten der MySQL-CS-Überwachung. Während sich das Überwachen auf Basis der Service-Konsole auf die Überwachung der verbrauchten Einheiten CPU/IO/Memory bei „metered Service“ richtet, steht mit dem ebenfalls auf Wunsch konfigurierten MySQL Enterprise Monitor ein sehr mächtiges Tool für die Live-Überwachung zur Verfügung. Es bietet einen sofortigen Überblick über den Status der MySQL-Datenbanken und überwacht kontinuierlich alle MySQL-Ser-

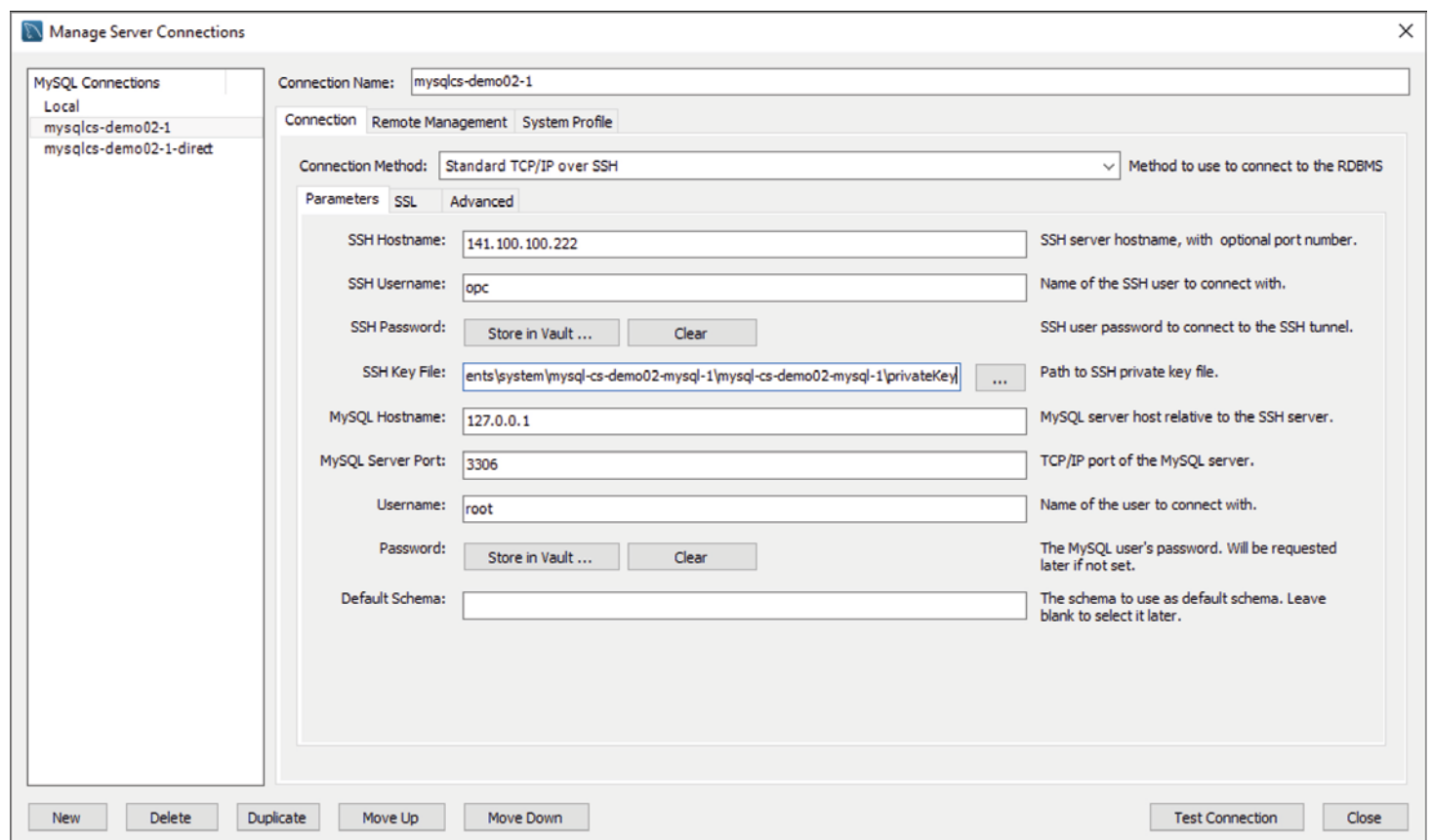


Abbildung 5: Konfiguration des MySQL-CS-Zugriffs in Workbench

ver mit fast 700 MySQL- und Betriebssystem-spezifischen Variablen. Zusätzlich informiert es dank zahlreicher Ratgeber über erkannte Probleme aus mehr als fünfzehn Kategorien (Sicherheit, Datensicherung, Administration, Zugriffsberechtigungen, SQL-Firewall, SQL-Befehle etc.). Über die Funktion der Abfrage-Analyse können SQL-Statements in einem konfigurierbaren Zeitraum eingesehen und ausgewertet werden. Gerade für Performance-kritische Einsätze lassen sich Probleme sehr schnell erkennen und effektiv beheben (siehe Abbildung 4).

Zugriff auf MySQL

Grundsätzlich kann man über die MySQL-Konnektoren/ODBC, über den MySQL-Kommandozeilen-Client oder die MySQL-Workbench auf die MySQL-Instanz zugreifen. Allerdings sind standardmäßig alle Ports für den Zugriff von außen gesperrt, sodass man für den direkten Zugriff auf MySQL zunächst den MySQL-Port 3306 im OPC-Access-Dashboard zulassen muss. Alternativ besteht auch die Möglichkeit, den beschriebenen SSH-Zugriff als Tunnel zu verwenden. Ein kleiner Fallstrick ist bei der Verwendung des beliebten Putty Clients unter Windows zu beachten: Möchte man Putty zum Zugriff auf MySQL CS verwenden, müssen zunächst die von MySQL CS bereitgestellten SSH-Keys in ein Putty-kompatibles Format („Putty Private Key File.ppk“) mittels „PuTTYgen“ konvertiert werden.

Für administrative Aufgaben empfiehlt sich die MySQL-Workbench. Damit kann man MySQL administrieren, aber auch als SQL-Client oder Entwicklungstool mit ER-Diagrammen verwenden. Die MySQL-Workbench bietet ebenfalls die Möglichkeit, auf Basis von Performance-Reports Leistungsdaten, Index-Verwendung und Ressourcen-Belegung auf Benutzer- und Anwendungsebene ohne SQL oder administrative MySQL-Kenntnisse einzusehen.

Die MySQL-Workbench steht für alle gängigen Betriebssysteme zur Verfügung. Sie kann als kostenfreie Version von „<http://www.mysql.com>“ heruntergeladen werden. Alternativ stellt Oracle eine spezielle Version mit erweiterten Funktionen für den MySQL CS unter „<https://support.oracle.com>“ zur Verfügung.

Die MySQL-Workbench unterstützt ebenfalls SSH-Tunneling. Eine Konvertierung der SSH-Schlüssel wie für Putty ist beim Einsatz der MySQL-Workbench nicht notwendig. Das Einrichten zum Zugriff auf MySQL CS über einen SSH-Tunnel vereinfacht sich daher auf einige grundsätzliche Angaben in Verbindung mit dem privaten SSH-Schlüssel (siehe Abbildung 5).

Besondere Bedeutung bekommt die MySQL-Workbench während Datenbank-Migrationen. Der MySQL-Workbench-Migration-Wizard unterstützt und überprüft die einzelnen Migrationsschritte, übernimmt sehr einfach Daten aus einer beliebigen Quelle (MS Access, Postgres, Sybase) und überträgt diese logisch in ein neues Ziel. Noch einfacher können die Daten mit der MySQL-Workbench-Schema-Transfer-Funktion übertragen werden, die allerdings als Quelle und als Ziel eine MySQL-Datenbank voraussetzt.

Der MySQL Cloud Service

In typischen Kunden-Szenarien werden nicht nur einzelne IT Services in der Cloud abgebildet. Vielmehr betreiben Kunden oftmals komplette Lösungen in der Cloud.

Daher verlagern sie nicht nur den Datenbank-Betrieb in die Cloud, sondern ganzheitlich auch die mit den Datenbanken verknüpften Applikations- und Infrastruktur-Umgebungen. Die OPC bietet daher bereits seit Längerem nicht nur einen MySQL-Datenbank-Service, sondern auf Wunsch auch anhängende (Applikations-) Software-Pakete (wie LAMP) im Rahmen der Compute Cloud Services mit an. Die OPC beschränkt sich hierbei nicht nur auf Linux, entsprechende Applikationspakete sind auch in Verbindung mit Oracle Solaris oder Microsoft Windows möglich. Auch die Oracle-eigenen Entwicklungstools (Oracle Java Cloud Service und Application Container Cloud) können so sehr einfach mit dem MySQL CS kombiniert und effektiv betrieben werden.

Fazit

Die Oracle Public Cloud bietet seit Längerem Lösungen für sehr viele Einsatzzwecke und praktisch jede Unterneh-

mensgröße an. Das Portfolio geht von einfachen Einstiegslösungen bis hin zu exklusiv nutzbaren Lösungen auf Basis einer Oracle Exadata. Das ohnehin sehr große OPC-Produktportfolio wird mit MySQL CS erneut um einen wichtigen Baustein erweitert und für Cloud-interessierte Kunden nochmals attraktiver.

Erste Erfahrungen mit MySQL lassen sich über „Free Trials“ unter „<https://cloud.oracle.com>“ sammeln. Oracle stellt hier zeitlich und leistungsmäßig begrenzte Test-Accounts zur Verfügung, um MySQL CS und/oder andere Oracle-Cloud-Angebote einzeln oder im Zusammenspiel ausführlich zu testen.



Carsten Thalheimer
carsten.thalheimer@oracle.com



Mario Beck
mario.beck@oracle.com

OPatch ferngesteuert: Automatisiertes Patchen mit Open Source Puppet

Raphael Daum, DBConcepts GmbH

Das Patchen einer Datenbank gehört zu den regelmäßig wiederkehrenden Aufgaben eines Datenbank-Administrators. Eine innige Beziehung wird aber selten daraus. Auf die Frage, ob das Patchen denn Spaß mache, pendelt die Antwort zwischen „na ja“ und „nein“. Aber wieso ist das so? Kann es nicht auch anders sein? Wenn viele der einzelnen und mühsamen Schritte automatisiert werden könnten, würde Patchen sogar Spaß machen. Dieser Artikel stellt mit dem Open-Source-Configuration-Management-Tool Puppet eine Möglichkeit vor, um das Patchen von Datenbanken quer über alle Oracle-Editionen kostengünstig zu automatisieren, sodass es zu einer entspannten Tätigkeit wird.

Jedes Quartal stellt Oracle neue Patches für die unterstützten Datenbank-Versionen zur Verfügung. Nicht wenige Unternehmen sind auf diesen Zug aufgesprungen und verlangen ein regelmäßiges und zeitnahes Patchen ihrer Datenbank-Systeme. Das ist in den meisten Fällen mit einigem Arbeitsaufwand für die involvierten Abteilungen verbunden. Es beginnt mit der Evaluierung des Ist-Zustands, der Vorbereitung der Systeme und endet mit der Ausarbeitung von detaillierten Wartungsfenstern und Rückstiegs-Szenarien. Durch viel Übung geht manches schneller von der Hand und kleinere Arbeitsschritte lassen sich mit Skripten vereinfachen. Trotzdem ist der Prozess ein langwieriger und birgt jedes Quartal neue Risiken: Welche Sicherheitslücken schließe ich mit dem Patch und welche bisher unbekannt Probleme könnten durch den aktuellen Patch entstehen? Was tue ich, wenn es schief geht?

Wo wollen wir hin? Wie schnell muss es gehen?

Bevor man sich intensiver damit auseinandersetzt, wie das Patchen automatisiert werden kann, muss geklärt sein, wie der aktuelle Ist-Zustand ist und wie der gewünschte Soll-Zustand aussehen soll. Was verhindert, den gewünschten Soll-Zustand zu erreichen? In den meisten Fällen ist es wohl eine Kombination von unterschiedlichen Faktoren: fehlende Zeit, unklare Prioritäten, mangelnde Praxis, fehlende Patch-Strategie, um nur einige zu nennen.

Die Automatisierung des Patch-Vorgangs kann zwar diese grundlegenden Probleme nicht lösen, sehr wohl aber entscheidend zu einer Lösung beitragen. Denn sobald sich das Unternehmen oder die zuständige Abteilung auf eine verbindliche Patching-Strategie geeignet hat, ist aus unternehmenspolitischer Sicht bereits alles

Notwendige getan. Nun beginnt die Phase, in der die technischen Fragen geklärt werden müssen. Während die Patching-Strategie die Frage nach dem Wann beantwortet, gibt die Patch-Automatisierung eine Antwort auf die Frage „Wie patchen?“

Die Qual der Wahl: PSU oder CPU oder SPU?

Bevor wir zur technischen Realisierung kommen, steht noch die Klärung der Frage im Raum, welche Patches die Grundlage für die Automatisierung darstellen können. Oracle bietet eine Fülle verschiedener Patches für Datenbanken an, aber welcher Patch ist der richtige? Critical Patch Update (CPU), Security Patch Update (SPU), Patch Set Update (PSU), Combo Patch (PSU plus OJVM) oder GI-Patch?

Die Erfahrung hat gezeigt, dass ein PSU den meisten Mehrwert und das ge-

ringste Risiko enthält. Aus diesem Grund beruht die hier vorgestellte Lösung auch darauf, ein PSU automatisiert zu installieren. Das hat damit zu tun, dass im PSU sowohl sicherheitsrelevante als auch performancerelevante Bugs geschlossen werden. Zusätzlich listet Oracle drei Vorteile des PSU auf:

- Low-Risk, High-Value Content
- One Integrated, well tested Patch
- Baseline-Version for easier Tracking

Der erste Punkt nimmt Bezug darauf, dass ein PSU keine Konfigurationsänderungen erforderlich macht und das Verhalten der Datenbank nicht verändert, wodurch auch keine Re-Zertifizierung notwendig wird. Im zweiten Punkt wird darauf angespielt, dass bis zu hundert Bugs pro PSU behandelt werden und die Fixes aufeinander abgestimmt und getestet wurden. Punkt drei erleichtert die eindeutige Identifizierung des Patch-Levels der Datenbank. Wird etwa der PSU vom 18. Oktober 2016 eingespielt, ändert sich das Patch-Level der Datenbank auf 12.1.0.2.161018 beziehungsweise 11.2.0.4.161018. Nachdem das aktuelle PSU immer alle vorigen PSUs enthält, ist somit ein damit eindeutiger Patch-Stand gewährleistet.

Spaß am Patchen

Damit wir uns dem Ziel nähern – nämlich das Patchen zu einer entspannten Tätigkeit zu machen –, müssen lediglich zwei Voraussetzungen erfüllt sein: Erstens brauchen wir ein Test-Umgebung und zweitens eine

Automatisierung, die uns die ganze händische und oft fehleranfällige Arbeit bei der Patch-Vorbereitung und dem eigentlichen Patchen abnimmt. Durch die Test-Umgebung lässt sich alles im Vorfeld gefahrlos testen und evaluieren, durch die Automatisierung ist Patch-ApPLY und Patch-Rollback in schnellen Zyklen kein Problem mehr. Automatisieren bringt unter anderem folgende Vorteile mit sich:

- Reproduzierbarkeit
- Die Konfiguration erfolgt an einer Stelle
- Deklaratives Arbeiten (Zielzustand)
- Wiederholte und wiederholbare Ausführung („idempotent“)
- Workflow ist im Code
- Individuelles und personengebundenes Know-how wird in Code übersetzt
- Code ist überwachbar (VCS)
- Sofortige Skalierungsgewinne

Deklarativ statt imperativ

Die Automatisierung von Patch-Vorgängen lässt sich mit sogenannten „Configuration Management Systemen“ implementieren. Deren zentrale Aufgabe ist es, auf einem Zielsystem einen gewünschten (deklarativen) Zustand herzustellen. In unserem Fall verwenden wir so ein System, um auf einem Oracle-Datenbank-Zielsystem einen gewünschten Patch-Level zu realisieren. Als Configuration Management System kommt dabei „Puppet“ zum Einsatz. Es ist Open Source und unterstützt eine Reihe von Plattformen und Architekturen (Linux, Windows, Solaris etc.). Puppet hat sich in den letzten Jahren als äußerst vielseitiges und robustes

Configuration Management System einen Namen gemacht und besticht durch eine rein deklarative Ausrichtung, was in komplexen IT-Umgebungen ein starker Trumpf ist: Wir geben nur den gewünschten Zielzustand eines Systems an und müssen uns nicht darum kümmern, was alles erforderlich ist, um diesen zu erreichen.

Es ist Aufgabe von Puppet, für jedes System den korrekten Weg zu wählen (imperativ). Uns interessiert lediglich der Zielzustand (deklarativ). Um diesen zu erreichen, vergleicht Puppet bei jeder Ausführung den Ist- mit dem Soll-Zustand. Besteht hier eine Differenz, werden die notwendigen Schritte durchgeführt, um das System in den Soll-Zustand zu bringen. Entspricht der Ist- bereits dem Soll-Zustand, passiert nichts; nichts anderes ist mit dem Begriff „idempotent“ gemeint.

Automate, automate, automate!

Die Vorteile einer Open-Source-Lösung liegen auf der Hand: Alle nachfolgend aufgeführten Aufgaben erfordern keine Management-Packs von Oracle. Die Patch-Automatisierung kann mit allen Oracle-Editionen (SE, SE ONE, SE2, EE) umgehen und unterstützt die Datenbank-Versionen 11.2.0.4, 12.1.0.1 und 12.1.0.2. Support für neue Releases kann einfach hinzugefügt werden. Oracle bietet mit dem haus-eigenen „Database Lifecycle Management Pack“ für den Enterprise Manager eine eigene Lösung für das automatisierte Patchen an. Es umfasst wesentlich mehr Funktionalität als das hier vorgestellte Patching, setzt aber auch den Einsatz der kostspieligen Oracle Enterprise Edition samt Management Pack auf allen Datenbank-Zielen voraus. Der Einsatz für die Datenbanken der Standard Edition (SE, SE ONE, SE2) ist lizenzrechtlich nicht erlaubt.

Abbildung 1 skizziert den Patching-Workflow mit Puppet. Überblicksmäßig realisiert die vorgestellte Open-Source-Lösung folgende Aufgaben:

- *Patch-Inventory*
In einem zentralen Patch-Inventory sind alle relevanten Informationen für einen Patch hinterlegt. Das betrifft unter anderem die Anwendbarkeit von Patches für verschiedene Architekturen und Versionen, die notwendigen Pre-Requi-

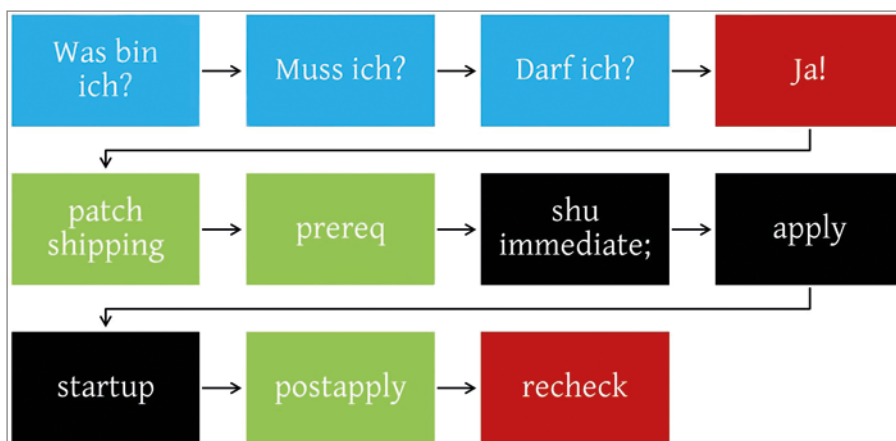


Abbildung 1: Patching-Workflow

```
# 12.1.0.2.161118 (Oct2016)
12.1.0.2::psu::ensure:      present
12.1.0.2::psu::identifier: 12.1.0.2.161118
12.1.0.2::psu::patchid:   24591642
12.1.0.2::psu::patchfile: p24922906_121020_MSWIN-x86-64.zip
12.1.0.2::psu::opatch:    12.1.0.1.2
12.1.0.2::psu::apply:     apply
12.1.0.2::psu::postapply: datapatch.bat -verbose
12.1.0.2::psu::postremove: datapatch.bat -verbose
12.1.0.2::psu::prereq:
  - CheckConflictAgainstOHWithDetail
  - CheckApplicable
```

Listing 1

```
"oradb_homes_c_oracle_product_11.2.0_dbhome_1": {
  "orabase": "C:\Oracle",
  "homepath": "C:\Oracle\product\11.2.0\dbhome_1",
  "homeowner": "ORA_DBA",
  "homegroup": "S-1-5-32-544",
  "homekey": "OraDb11g_home1",
  "opatchversion": "11.2.0.3.15",
  "oracleversion": "11.2.0.4",
  "oracleproduct": "rdbms",
  "psudesc": "No PSU description found",
  "psu": "No PSUs detected"
},

"oradb_homes_c_oracle_product_12.1.0.2_dbhome_1": {
  "orabase": "C:\Oracle",
  "homepath": "C:\Oracle\product\12.1.0.2\dbhome_1",
  "homeowner": "ORA_DBA",
  "homegroup": "S-1-5-32-544",
  "homekey": "OraDB12Home2",
  "opatchversion": "12.1.0.1.3",
  "oracleversion": "12.1.0.2",
  "oracleproduct": "rdbms",
  "psudesc": "WINDOWS DB BUNDLE PATCH 12.1.0.2.161118(64bit):24922906",
  "psu": "24922906"
},
```

Listing 2

rement-Checks, die minimal notwendige OPatch-Version und Schritte zum Installieren und Deinstallieren des Patches.

- **Patch-Storage**
Das Patch-Inventory verweist auf ein zentrales Patch-Storage, auf dem nicht nur die Patches hinterlegt sind, sondern auch weitere Utilities und Skripte, die für das Patchen notwendig sind.
- **Rollout**
Die Patches werden auf die frei definierbaren Zielsysteme ausgerollt und entpackt.
- **Dry-Run**
Ein Dry-Run-Modus ermöglicht es, alle Schritte für das Patchen („apply“ oder

„rollback“) zu simulieren und die Pre-Requirement-Checks auszuführen. Das ist eine Aufgabe, die bereits vor dem eigentlichen Wartungsfenster ausgeführt werden kann und soll. Falls es hier zu Problemen kommt, die sich nicht automatisiert lösen lassen, besteht noch ausreichend Zeit, um sie zu korrigieren. Das könnten etwa vorher installierte Oneoff-Patches sein, die mit dem PSU in Konflikt stehen. Hier hilft dann nur ein Eingriff von Hand. Sollte der Dry-Run erfolgreich gewesen sein, sind bereits viele Stolpersteine für das Wartungsfenster aus dem Weg geräumt.

- **Patch**
Durch das Patch-Shipping sowie die Prüfung der Voraussetzungen und

möglichen Patch-Konflikte wird die Downtime auf die reine Patch-Dauer („patch apply“) reduziert. Patches können automatisiert installiert („apply“) und deinstalliert („rollback“) werden.

- **Kill or Report?**
Sollte das Patchen trotz erfolgreichem Dry-Run scheitern, liegt das meistens an offenen File-Handles und an von anderen Prozessen verwendeten DLLs, was zur Folge hat, dass OPatch die gewünschte Datei nicht aktualisieren kann. Die vorgestellte Patch-Automatisierung verfügt deshalb über die Funktionalität, solche Showstopper ausfindig zu machen und entweder die verantwortlichen Prozesse gleich zu beenden oder lediglich zu berichten.

Komponenten und Konfiguration

Puppet arbeitet mit einer klassischen Client-Server-Architektur: Auf der einen Seite steht ein Puppet-Server (der in unserem Fall auch als Patch-Inventory und Patch-Storage dient), auf dem die gewünschte Ziel-Konfiguration für jedes System hinterlegt ist. Die Konfiguration liegt in einfachen YAML-Files. Auf der anderen Seite läuft am Zielsystem der Puppet-Agent, dessen Aufgabe es ist, regelmäßig den Ist-Zustand des Systems an den Puppet-Server zu schicken und vom Puppet-Server übermittelte Aufträge zur Erreichung des Soll-Zustands umzusetzen. Der Puppet-Agent ist die einzige Software, die am Zielsystem installiert werden muss, um die skizzierte Patch-Automatisierung umzusetzen. Für jeden Patch gibt eine Konfigurationsdatei, in der alle notwendigen Informationen hinterlegt sind (siehe Listing 1).

Am Zielsystem läuft ein Prozess mit dem Namen „factor“, der – wie der Namen bereits erahnen lässt – den aktuellen Ist-Zustand erhebt und an den Puppet-Server schickt. Listing 2 zeigt einen Auszug des Outputs.

Fazit

Die vorgestellte Lösung einer Patch-Automatisierung unternimmt den bescheidenen Versuch, Patch-Abläufe stressfreier

und entspannter zu gestalten, und greift dafür auf die etablierte Open-Source-Software „Puppet“ zurück. Durch die Automatisierung kann eine große Anzahl von Datenbanken schnell und regelmäßig auf das gewünschte Patch-Level gehoben werden.

Im Gegensatz zu anderen Lösungen funktioniert das Patchen mit Puppet unab-

hängig von Datenbank-Version und -Edition und macht es damit zu einer universellen, sehr kostengünstigen Lösung, weil keine zusätzlichen Lizenzen erforderlich sind. Schließlich: Regelmäßiges Patchen spart Nerven (weil das Know-how bereits im Code ist), Zeit (aufgrund der Automatisierung) und Geld (weniger Downtime, weniger Stillstände aufgrund von Bugs).



Raphael Daum
raphael.daum@dbconcepts.at

Entspannt virtualisieren mit dem offenen Oracle-Multipurpose-Hypervisor

Nico Henglmüller und Dr. Thomas Petrik, Sphinx IT Consulting GmbH

Wünschen Sie sich einen stabilen Betrieb Ihrer Virtualisierungslösung sowie Oracle-Datenbanken und Applikationen, und das Ganze in einem sicheren Lizenz-Umfeld? Alles kein Problem, legen Sie die Beine hoch.

Virtualisierungsmethoden sind aus der modernen Betriebsführung nicht mehr wegzudenken. Aktuelle Virtualisierungslösungen können nach ihren Voraussetzungen bei der Installation in zwei Gruppen eingeteilt werden:

- **Bare-Metal**

Diese Art der Hypervisor wird direkt auf die vorhandene Hardware installiert, verwaltet die Ressourcen-Aufteilung und stellt die Treiber für den Hardware-Zugriff selbst bereit. Vertreter dieser Gruppe sind Xen, VMware ESXi, KVM oder Microsoft Hyper-V.

- **Betriebssystem**

Diese Gruppe benötigt ein vorinstalliertes Betriebssystem, da dieses die Treiber für den Hardware-Zugriff bereitstellt. Zu dieser Gattung zählen Oracle VM VirtualBox, VMware Workstation oder Parallels Desktop.

Lösungen, die ein Betriebssystem voraussetzen, werden hauptsächlich auf Workstations eingesetzt, während Bare-Metal-Installationen im Server-basierten Umfeld ihren Einsatz finden. In der Gruppe der Bare-Metal-Installationen befinden sich zwei Open-Source-Virtualisierungslösungen: Xen und KVM. Die Referenzliste von Xen reicht von Amazons Webservices über diverse Google-Produkte bis hin zu den Cloud-Angeboten von Rackspace und Yahoo.

Seit dem Jahr 2007 erweitert Oracle schrittweise die Xen-Implementierung des Xen-Projekts und stellt die in Oracle VM verwendeten Quelltexte öffentlich unter der GNU GPL v2 bereit. Das Resultat dieser Anstrengungen ist der Oracle-VM-Server, der Xen den Schrecken nahm und das Herz der Oracle-Virtualisierungslösung ist. Um diesen Kern baute Oracle eine grafische Oberfläche zur Verwaltung sowie ein API zur Automatisierung, ohne den schlanken Hypervisor um zusätzliche Funktio-

nen aufzublähen. Für den hochausfallsicheren Betrieb der Virtualisierungslösung spielt OCFS2 als Cluster-File-System eine zentrale Rolle. Unter anderem stellt es elementare Funktionen der Virtualisierung, beispielsweise „Copy on write“-Snapshots für Thin Clones oder das dynamische Allokieren von Speicherplatz für virtuelle Festplatten, bereit. Zudem ist der OCFS2-Layer für den vollautomatisierten HA-Betrieb (inklusive Heartbeat) verantwortlich.

Architektur

Die Architektur von Oracle VM besteht aus den folgenden Komponenten (*siehe Abbildung 1*). Der Oracle-VM-Server wird innerhalb weniger Minuten Bare-Metal installiert. Neben der vertikalen Skalierung innerhalb eines Servers bietet ein Oracle-VM-Server-Pool, bestehend aus mehreren Oracle-VM-Servern, die Mög-

lichkeit der horizontalen Skalierung. Beispielsweise ist eine einzige Installation in der Lage, bis zu 384 physische CPU-Kerne und 6 TB RAM zu verwalten. Einer einzelnen virtuellen Maschine können 256 vCPUs (damit sind Threads am Hypervisor-Knoten gemeint) und etwa 2 TB Arbeitsspeicher bereitgestellt werden. Dies alles ermöglicht das Herz der Virtualisierungslösung, der offene Xen Hypervisor, denn der Oracle-VM-Server basiert auf dem Xen-Kernel mit der privilegierten Verwaltungsdomain oder „dom0“, mehreren virtuellen Maschinen oder „domU“ und einem Oracle-VM-Agent innerhalb der „dom0“. Xen übernimmt die Aufgabe des Hypervisors, bietet die Funktionalität der Live-Migration, startet oder stoppt virtuelle Maschinen und verwaltet die Ressourcenzuteilung für diese. Mit Oracle-Ksplice und dem Einzug in den Unbreakable-Enterprise-Kernel ist Oracle VM in der Lage, Kernel-Updates ohne Unterbrechung des Betriebs anzuwenden.

Der Oracle-VM-Manager enthält die gesamte Konfiguration der Topologie der Cluster (Knoten, Pools, VMs, Storage etc.) und speichert diese in einer eigenen MySQL-Datenbank. Mithilfe des Managers ist es möglich, Hunderte bis Tausende virtuelle Maschinen zentral und ohne eine zusätzliche Client-Software zu verwalten. Er besteht aus einem REST- beziehungsweise SOA-API, das die Basis für das Commandline-Interface und die grafische Web-Browser-Oberfläche bildet. Der Manager selbst ist eine proprietäre Java-Applikation, die auf einem WebLogic-Server 12c bereitsteht. Wird nun eine neue virtuelle Maschine erstellt, werden die Meta-Informationen in der Datenbank persistiert, durch den Oracle-VM-Agent auf einem VM-Server angelegt und dort in einer Xen-üblichen Konfigurationsdatei gespeichert. Die Konfiguration des Servers ist in einer lokalen Berkeley-Datenbank gespeichert. Diese Implementierung ermöglicht es dem Oracle-VM-Server, im Falle eines Ausfalls des Manager voll funktionsfähig zu bleiben. Die Kombinationen der VM-Konfigurationsdateien und die Berkeley-Datenbank auf den VM-Servern erlauben es, dass ein Manager jederzeit wiederhergestellt werden kann.

Shared Storage stellt über das Netzwerk Speicherplatz für virtuelle Maschinen an alle VM-Server eines Server-Pools bereit. Prinzipiell kann ein Storage über

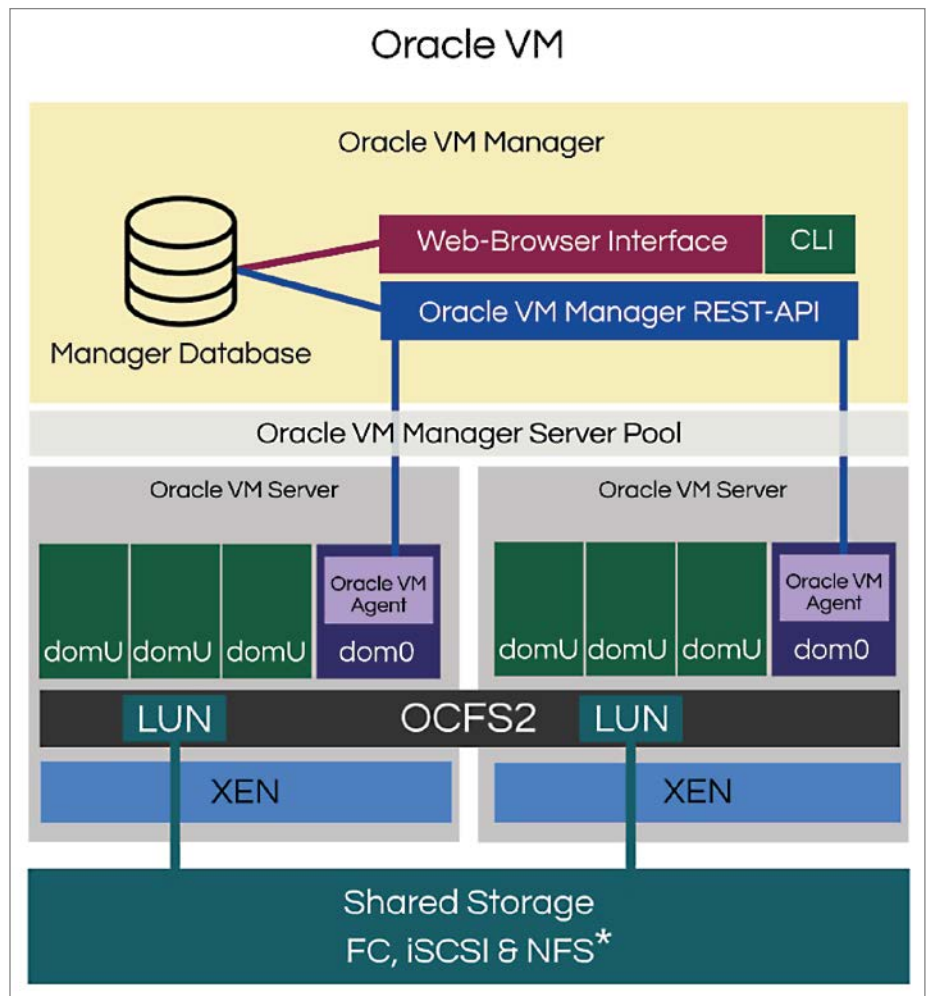


Abbildung 1: Architektur der Oracle VM

* Für die volle HA-Cluster-Funktionalität ist zwingend FC oder iSCSI erforderlich

NFS, Fibre Channel, Fibre Channel over Ethernet oder iSCSI an die VM-Server eines Server-Pools angebunden sein. Oracle selbst testet und zertifiziert Storage-Appliances, um einen reibungslosen Betrieb der Virtualisierungslösung zu garantieren. Durch Storage-Connect-Plug-ins ist die zentrale Verwaltung von Storage-Arrays und darauf betriebenen Storage-Repositories innerhalb des VM-Manager möglich.

Die angesprochene Cluster-Fähigkeit, Hoch-Ausfallsicherheit und die Funktionalität für Thin-Clones wird durch das Oracle-Open-Source-Cluster-File-System OCFS2 bereitgestellt. Storage-Repositories werden aus den LUNs der darunterliegenden Arrays aufgebaut und mit OCFS2 formatiert. In den Repositories sind virtuelle Festplatten, die Meta-Informationen der virtuellen Maschinen, VM-Templates und -Appliances sowie Installationsdateien abgelegt. Ein Storage-Repository muss für alle VM-Server eines

Server-Pools erreichbar und beschreibbar sein. Mit der Copy-On-Write-Funktionalität des File-Systems ist es möglich, innerhalb von Sekunden einen vollwertigen Klon einer virtuellen Maschine zu erstellen. Dieser lässt sich sofort starten; die zunächst nur durch Ref-Links gemeinsamen Blöcke werden während des Betriebs dupliziert.

Open Source und Enterprise ready?

Oracle unterstützt die Entwicklung des Xen-Projekts im Zuge der Open-Source-Strategie und ist eines der zwölf unterstützenden Unternehmen im Advisory Board des Xen-Projekts. Zu den Mitgliedern des Boards zählen Unternehmen wie Intel, AMD und Citrix. Die Quelltexte der von Oracle VM verwendeten Xen-Pakete und OCFS2 sind genauso wie der verwendete Unbreakable-Enterprise-Ker-

nel öffentlich zugänglich und unterliegen der GNU GPL v2. Oracle bietet zu Oracle VM optional kommerziellen Support, der unter anderem Benachrichtigungen bei Security-Updates und eine unlimitierte Anzahl von Service-Anfragen für den gesamten Virtualisierungsstack umfasst. Darüber hinaus garantiert Oracle die Funktion und den Support für zertifizierte Server- und Storage-Systeme. Dies ist durch die Zusammenarbeit mit Hardware-Partnern sichergestellt und wird laufend überarbeitet und aktualisiert. So ist der Support für Oracle-Kunden beginnend beim Shared Storage bis zu den Oracle-Applikationen gewährleistet.

Anspruch von Oracle ist es, mit Oracle VM eine Lücke im eigenen Portfolio zu schließen, um den gesamten Stack von der Storage Appliance über den Hypervisor bis zur Applikation zu unterstützen. Allerdings ist Oracle VM nicht auf Produkte von Oracle oder bestimmte Betriebssysteme beschränkt. Der Fokus der Entwicklung lag von Beginn an auf dem vielseitigen und uneingeschränkten Einsatz des Hypervisors. Diesen „Multipurpose“-Anspruch unterstreicht die Zertifizierung aktueller Microsoft-Windows-basierter Betriebssysteme im Zuge des „Server Virtualization Validation Program“ (SVVP) von Microsoft. Diese Zertifizierung stellt sicher, dass Microsoft-Windows Betriebssysteme auf der zertifizierten Virtualisierungslösung unterstützt und mit dem Partner eine Support-Matrix bereitstellt.

Oracle OpenStack for Oracle Linux

Mit „Oracle OpenStack for Oracle Linux“ setzt Oracle den „Multipurpose“-Ansatz konsequent fort. Ziel dieser Entwicklung ist, dass die Vorteile von OpenStack und des Oracle-applikationszentrierten Ansatzes Einzug in die Virtualisierung halten. In Zukunft könnte der Oracle-VM-Manager durch eine passgenaue Administrationsoberfläche auf Basis des OpenStack-Projekts Horizon abgelöst werden.

Die Oracle-VM-Server finden in der OpenStack-Architektur weiterhin einen sinnvollen Einsatz, da das eingesetzte Xen bereits die notwendige Implementierung der verwendeten Schnittstelle bereitstellt, denn das OpenStack-Projekt

Nova kommuniziert mit den angebotenen Hypervisoren über die „libvirt“-Schnittstelle, die von allen unterstützten Virtualisierungslösungen implementiert wird. So kann auf einem Nova-Knoten KVM und auf einem weiteren ein Oracle-VM-Server als Hypervisor verwendet werden. Auf der diesjährigen Oracle OpenWorld präsentierte Simon Coter, Product Manager Oracle VM, das produktive Minor-Release und gab mit einer Beta-Version einen Ausblick auf die nahe Zukunft der Oracle-OpenStack-Ambitionen.

Enterprise-Features

Oracle VM wurde mit dem Fokus auf den hochverfügbaren und effizienten Betrieb von Enterprise-Anwendungen entwickelt. Darunter befindet sich die Live-Migration von virtuellen Maschinen, Clustering oder Dynamic-Power-Management (DPM), um das Einsparen von Energie in den Vordergrund zu stellen. Dies wird durch das automatisierte Abschalten von nicht ausgelasteten Oracle-VM-Servern erreicht. Die virtuellen Maschinen werden natürlich vor dem Abschalten auf ausgewählte Server innerhalb des Pools live migriert.

Das Pendant dazu ist das Dynamic Resource Scheduling (DRS), das virtuelle Maschinen gleichmäßig auf die Oracle-VM-Server aufteilt und dadurch die Last auf alle Server eines Pools verteilt. Der Auslöser für die Live-Migration der virtuellen Maschinen ist die prozentuale Prozessor-Last auf einem Knoten. Dieser Grenzwert ist frei wählbar und sollte je nach Lastprofil und den Anwendungen in den VMs gewählt werden.

Im Falle von unvorhergesehenen Unterbrechungen im Betrieb eines Oracle-VM-Servers werden die hochverfügbaren virtuellen Maschinen auf einem Server innerhalb des Pools neu gestartet. Da die virtuellen Festplatten und die VM-Konfigurationsdateien aller VMs auf dem Shared-Storage gespeichert sind, ist sichergestellt, dass jeder Server innerhalb des Pools in der Lage ist, die virtuelle Maschine neu zu starten. Oracle VM bietet ebenfalls die Möglichkeit, eine virtuelle Maschine innerhalb von Sekunden zu klonen und einen solchen vollwertigen Klon anschließend zu starten. Dies ist ein Alleinstellungsmerkmal von Oracle VM und beispielsweise in VMware-Umgebungen nicht möglich.

Vorteile für Oracle-Kunden

Da Oracle VM absolut konform zu den Oracle-Hard-Partitioning-Bestimmungen ist, bietet Oracle VM besondere Vorteile bei der Lizenzierung der Oracle-Produkte. Im Gegensatz zu anderen Virtualisierungslösungen ist es nicht notwendig, einen kompletten Cluster zu lizenzieren.

Mit Oracle VM kann man auf Server- oder CPU-Core-Basis lizenzieren und so nur die tatsächlich verwendete Hardware bezahlen. Dafür sind folgende Partitionierungsmöglichkeiten vorgesehen:

- *Server Pinning*
Eine virtuelle Maschine kann derart konfiguriert sein, dass sie nur auf einem Oracle-VM-Server startet. Daher sind nur die Prozessoren des Knotens zu lizenzieren.
- *CPU Pinning*
Oracle VM bietet die Möglichkeit, virtuelle Maschinen auf eine bestimmte Anzahl von CPU-Threads (vCPUs) zu pinnen. In diesem Fall müssen nur die betroffenen Cores lizenziert werden. Es besteht auch die Möglichkeit, mehrere VMs auf dasselbe vCPU-Set zu pinnen und so dieses Subset an Cores als Lizenz-Pool zu verwenden.

Im Oracle-Hard-Partitioning-Dokument wird allerdings auf folgende Bedingungen eingeschränkt:

- Die Live-Migration von virtuellen Maschinen mit aktiviertem Pinning ist untersagt.
- Trotz Pinning bleibt die HA-Funktionalität voll intakt. Das bedeutet, dass die virtuelle Maschine im Falle eines Knoten-Ausfalls auf einem anderen Oracle-VM-Server gestartet wird („cold failover“). Allerdings ist darauf zu achten, dass diese virtuelle Maschine nicht länger als insgesamt zehn Tage pro Jahr auf einem nicht lizenzierten Knoten betrieben wird. Alternativ kann zum Beispiel die gleiche Anzahl von CPU-Kernen oder Sockets für einen zweiten Knoten lizenziert werden. Dadurch ist der dauerhafte, hochausfallsichere Betrieb der Datenbank durch die Virtualisierung auch über die Zehn-Tage-Regelung hinaus sichergestellt.

Beim Betrieb der Oracle-Produkte auf Virtualisierungslösungen geht es aus or-

organisatorischer Sicht stets um Lizenz-Sicherheit, Zertifizierungen und das Supportmodell des gesamten Soft- und Hardware-Stacks. Die Komplexität der Lizenz-Regelungen für Failover, Clustering, Hart Partitioning und Data Recovery erfordern eine umfassende Kenntnis der Lizenz- und Support-Bestimmungen.

Für einen technisch reibungslosen Betrieb und den optimierten Kosteneinsatz ist eine Beratungsleistung dringend anzuraten. In deren Zuge erfolgt eine umfassende Anforderungs-Analyse. Darauf aufbauend werden Architekturen und Lizenz-Optimierungen für die jeweiligen Lösungsansätze erstellt. Diese sollten im Rahmen eines PoC gemeinsam mit den erwarteten Performance-Werten auf Herz und Nieren geprüft werden.

Die dokumentierten Ergebnisse des PoC stellen das Fundament für eine Architektur-Entscheidung dar. Das Ergebnis dieser Vorgehensweise ist ein umfassendes Betriebsführungskonzept, um den ausfallsicheren Betrieb der Virtualisierungslösung, der Applikationen und Datenbanken im Rahmen eines abgesicherten Lizenz-Umfelds sicherzustellen.

Fazit

Mit Oracle VM schließt Oracle die Lücke im Portfolio und stellt eine offene Möglichkeit für den effizienten und hochausfallsicheren Betrieb von Enterprise-Applikationen bereit. Durch die Wahl von Xen

als Hypervisor, die stetige Unterstützung des Xen-Projekts und die Teilnahme im Advisory-Board stärkt Oracle seine Open-Source-Strategie. Die Zukunft des offenen Hypervisors als Kern von Oracles Virtualisierungslösung scheint mit den aktuellen OpenStack-Ambitionen gesichert.

Quellen

- Oracle VM 3, Architecture and Technical Overview:
<http://www.oracle.com/technetwork/server-storage/vm/ovm3-arch-tech-overview-459307.pdf>
- Licensing Data Recovery Environments:
<http://www.oracle.com/us/corporate/pricing/data-recovery-licensing-070587.pdf>
- Database Licensing:
<http://www.oracle.com/us/corporate/pricing/databaselicensing-070584.pdf>
- Oracle OpenStack for Oracle Linux, Installation and User's Guide for Release 2:
http://docs.oracle.com/cd/E64747_01/E64749/E64749.pdf
- Hard Partitioning with Oracle VM Server for x86:
<http://www.oracle.com/technetwork/server-storage/vm/ovm-hardpart-168217.pdf>
- Oracle Linux and Oracle VM Sources:
<https://oss.oracle.com/oraclevm/server>
- Oracle Software Investment Guide:
<http://www.oracle.com/us/corporate/pricing/sig-070616.pdf>

- Oracle VM, Concepts Guide for Release 3.4:
http://docs.oracle.com/cd/E64076_01/E64081/E64081.pdf
- Server/Hardware Partitioning:
<http://www.oracle.com/us/corporate/pricing/partitioning-070609.pdf>



Nico Henglmüller, BSc
nico.henglmueLLer@sphinx.at



Mag. Dr. Thomas Petrik
thomas.petrik@sphinx.at

Oracle schließt 270 Sicherheitslücken

Das Critical Patch Update vom Januar 2017 schließt diverse Sicherheitslücken verteilt über nahezu das gesamte Produkt-Portfolio. Die höchsten Sicherheitsrisiken betreffen Primavera P6 Enterprise Project Portfolio Management. Über zwei Sicherheitslücken

können Angreifer hier Zugang ohne Authentifizierung erlangen. Weitere wichtige Patches für die Produkte Oracle Database Server, Oracle Enterprise Manager Grid Control, Oracle E-Business Suite, Oracle Industry Applications, Oracle Fusion Middle-

ware, Oracle Sun Products, Oracle Java SE, und Oracle MySQL sollten laut Oracle ebenfalls umgehend installiert werden.

Weitere Informationen unter <http://www.oracle.com/technetwork/security-advisory/cpujan2017-2881727.html>



OpenDAX-APIs: Software in Silizium als Turbo für eigene Anwendungen

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

Mit der aktuellen Generation der SPARC-Prozessoren in den T7-, M7- und S7-Systemen führte Oracle unter dem Oberbegriff „Software in Silicon“ eine Reihe von Technologien ein, um mit neuen Hardware-Mechanismen zum einen die Sicherheit und zum anderen die Performance signifikant zu verbessern. Ganze Kategorien von Sicherheitsproblemen, die aus Programmierfehlern resultieren, können durch diese Sicherheitsmechanismen in der Hardware ganz unten im Stack transparent für die höheren Schichten abgefangen werden.

Während andere CPU-Hersteller Grafikunterstützung auf der CPU implementieren, die mittlerweile auch im High Performance Computing sowie einigen weiteren Bereichen genutzt werden, hat Oracle Data Analytics Accelerator (DAX) auf dem Prozessor integriert, die Operationen übernehmen, die in der Datenbank und der Daten-Analyse relevant sind. Nachdem die Nutzung der DAX zunächst nur der Oracle-Datenbank vorbehalten war, die damit In-Memory-Operationen beschleunigt [1], wurde im März 2016 ein API für die DAX offenge-

legt. Damit können Entwickler sich deren Funktionalitäten jetzt für eigene Anwendungen zunutze machen, um deren Performance signifikant zu steigern – oft um eine Größenordnung. Prototypisch wurde dieses Open-DAX-API für Apache Spark genutzt [2], ein Open-Source-Framework zur schnellen Verarbeitung von Big Data mit Fokus auf eine In-Memory-Verarbeitung. Zudem gibt es eine Implementierung des mit Java 8 eingeführten Java-Streams-API, mit dem Java-Anwendungen transparent die DAX nutzen können [3].

Data Analytics Accelerator und DAX-APIs

Der DAX kann die CPU-Kerne insbesondere von analytischen Operationen auf Datenströmen entlasten, etwa wenn große, im Speicher komprimiert vorliegende Datenbestände über Prädikate gefiltert werden. Mit der zunehmenden Verbreitung von Software-Architekturen, die auf eine In-Memory-Verarbeitung ausgelegt sind, werden solche Operationen zu wesentlichen Faktoren für die Performance.

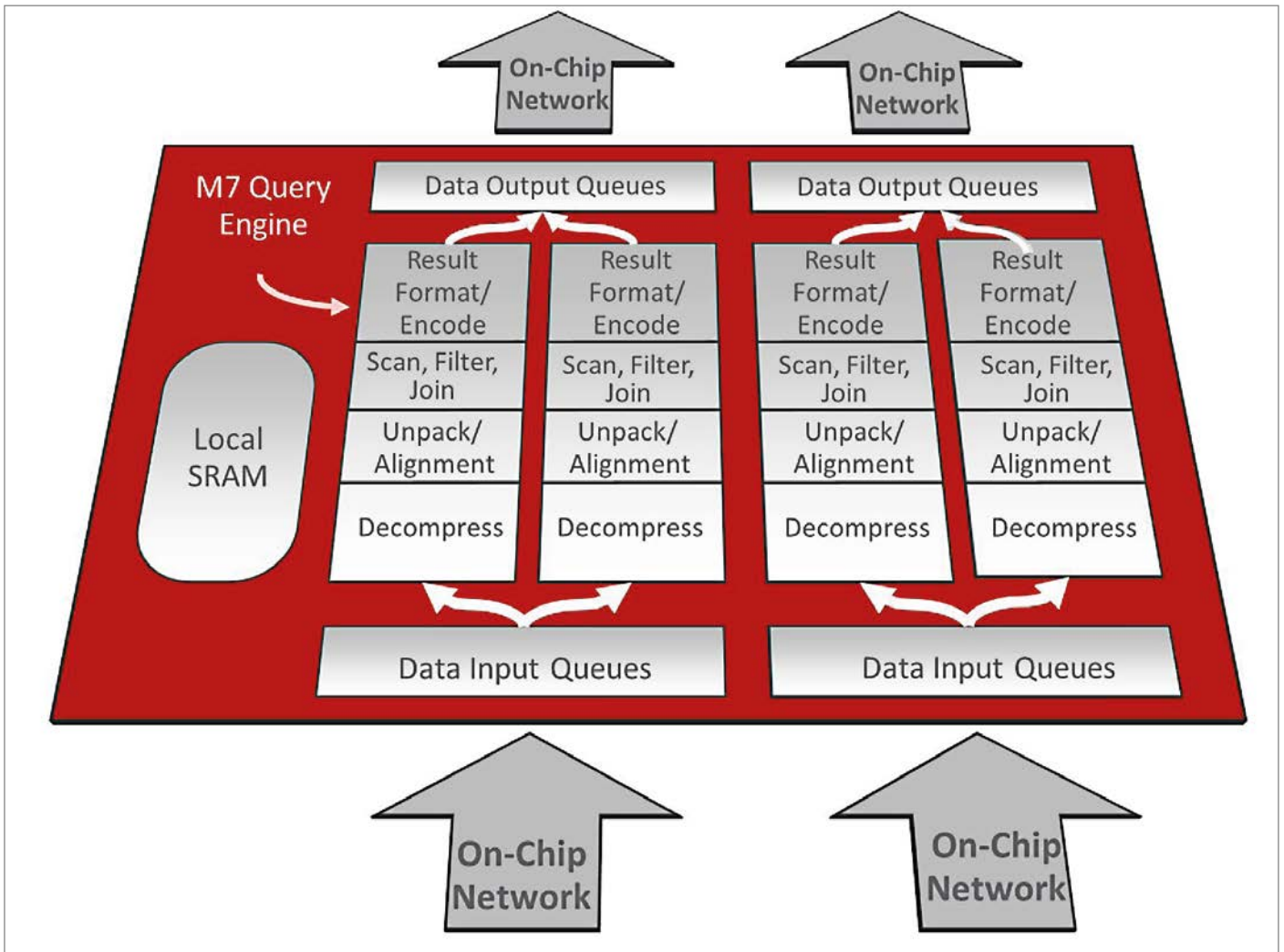


Abbildung 1: Ein DAX mit vier Engines. Der SPARC-M7-Prozessor (in M7- und T7-Servern) verfügt über acht und der S7 (in S7-Servern und im MiniCluster) über vier dieser DAX

Auf dem SPARC-M7-Prozessor finden sich acht und auf dem S7 vier DAX, jeweils mit vier Pipelines, sodass auf dem M7 32 und auf dem S7 16 DAX-Engines verfügbar sind. Sie benötigen weniger als ein Prozent der Chipfläche, aber jede DAX-Engine liefert etwa bei der Dekompression das Leistungsäquivalent zweier General Purpose Cores.

Die DAX liegen an der Schnittstelle des Prozessors zum Hauptspeicher. Das bringt beim Filtern von Daten einen großen Vorteil, weil nur selektierte Daten in die Prozessor-Caches gelangen, was deren Effektivität signifikant verbessert. Zum anderen ist es so möglich, die volle Transfer-Bandbreite zwischen Hauptspeicher und Prozessor zu nutzen, die zudem bei den SPARC-Systemen gut doppelt so hoch ist wie bei x86- oder POWER-Systemen. Ansonsten wird die Verarbeitung in den Prozessorkernen zum limitierenden

Faktor. Läuft die Verarbeitung in den DAX, werden die Kerne kaum belastet und stehen für andere Aufgaben zur Verfügung. Der Leistungsgewinn durch die DAX liegt in vielen Anwendungsbereichen oft bei einer Größenordnung.

Die DAX operieren auf Vektoren mit Elementen fester Länge (Bits oder Bytes), variabler Länge (mit einem Längenfeld im Element oder über einen separaten Hilfsvektor), Lauflängen- (RLE) oder ZIP-kodierten (OZIP) komprimierten Elementen. Eine spezielle Form sind Bit-Vektoren, die unter anderem für die Auswahl von Elementen oder als Ergebnisvektoren von Operationen auf anderen Vektoren genutzt werden. Das API umfasst folgenden Funktionen:

- Zum Filtern von Vektoren, die die Elemente eines Vektors gegen einen Wert

oder einen Wertebereich vergleichen und einen Bitvektor mit dem Ergebnis liefern.

- Zum Transformieren eines Vektors über eine Bitmap in einen Bit-Vektor. Der Wert des i -ten Elements des Eingabevektors wird dabei als Index in die Bitmap verwendet und der Bitwert daraus ergibt den i -ten Eintrag im resultierenden Bitvektor. Diese Funktion ermöglicht unter anderem einen einfachen Test auf Mengen-Zugehörigkeit: Ein Bit in der Bitmap ist gesetzt, wenn der Index in die Bitmap als Wert in der Menge enthalten ist. Damit werden dann im Ergebnisvektor genau die Bits für die Elemente des Eingabevektors gesetzt, deren Werte zur Menge gehören.
- Zum Liefern nicht des Bitvektors, sondern der Indizes der Elemente im Ein-

- gabevektor, die in der Bitmap den Wert 1 ergeben.
- Zum Entpacken gepackter – insbesondere komprimierter – Vektoren in solche mit Elementen fester Länge (einer Zweierpotenz an Bytes). Es gibt auch Funktionen zum Komprimieren, die allerdings rein in Software und nicht im DAX realisiert sind.
- Zum Kopieren der Elemente aus einem Eingabe-Vektor über einen Bit-Vektor in einen Ausgabe-Vektor fester Länge.
- Zum Ausführen von logischen Operationen („and“, „or“, „xor“) auf Bit-Vektoren, die allerdings in der libdax Version 1 in Software implementiert sind.
- Zum Kopieren oder Füllen von Speicherbereichen mit definierten Werten.
- Zum Debugging und Tuning.

Diese Funktionen können entweder synchron aufgerufen werden oder asynchron über einen Post-Poll-Mechanismus, wobei dann auch mehrere Funktionen anstehen können. Dabei werden von den Prozessorkernen Bearbeitungsaufträge über ein speicherbasiertes Protokoll an die DAX gegeben. Jeder Kern kann dabei jeden DAX ansprechen, die Zuordnung von Aufträgen erfolgt automatisch, ein Monitoring des Auslastungsgrades ist möglich [4, 5]. Die Low-Level-DAX-APIs sind über die Shared-Library „libdax.so.1“ und die Definitionen in „dax.h“ implementiert. Die Funktionen sind etwas ausführlicher in [6] beschrieben sowie insbesondere unter „libdax(3LIB)“. Dazu gibt es mit Sprach-Anbindungen für verschiedene Programmiersprachen (C, C++, Java (JNI), Python und SQL) höhere sogenannte „Vector-APIs“, die insbesondere die Abbildung von Datentypen, das Anlegen und Traversieren von und Hantieren mit Vektoren vereinfachen. Daten lassen sich auch aus JSON-formatierten Dateien oder Dateien mit Komma-separierten Werten (CSV) lesen.

Die Implementierung ist noch nicht standardmäßig in Solaris enthalten, wird

aber Entwicklern in der Silicon Developer Cloud zur Verfügung gestellt, in der man sie auch gleich ausprobieren kann, auch wenn man selbst über keinen M7-/T7- oder S7-SPARC-Server verfügt [13]. Dort sind auch einige vollständige Programmbeispiele in den verschiedenen Sprachen zu finden, etwa für das Durchsuchen von Schlüssel-Wert-Paaren oder die Ermittlung von Ausreißern, die um ein Vielfaches größer oder kleiner sind als der Median von zehn Millionen Zahlen [7].

Oracle-SPARC-Software in Silizium als Turbo für Apache Spark

„Apache Spark is a fast and general engine for large-scale data processing“ lautet die Beschreibung, mit der sich dieses umfangreiche, populäre Open-Source-Projekt selbst beschreibt [8]. Etwas salopp kann man Apache Spark als eine Art In-Memory-Hadoop ansehen, das etwa dann vorteilhaft ist, wenn schnellere Antwortzeiten erforderlich sind, als sie Batch-orientierte Hadoop-Jobs erreichen, in denen beim Map-Reduce Platten-orientiert auf Daten im HDFS gearbeitet wird.

Spark unterstützt eine Hauptspeicher-orientierte Verarbeitung auf Basis sogenannter „Resilient Distributed Datasets“ (RDD) zum einen für Zwischenergebnisse mehrstufiger Operationen und zum anderen für unterschiedliche Anfragen auf denselben Daten. RDD können aus Ausgangsdaten erzeugt werden oder durch Transformationen aus existierenden RDD. Sie sind unveränderlich. Es gibt zwei Arten von Operationen auf RDD: Transformationen wie „map“ oder „filter“, die aus einem RDD ein neues erzeugen, sowie Aktionen, die aus dem RDD einen Wert berechnen wie „count“ oder „reduce“. Hinzu kommen eingebaute Funktionen wie das Zählen von Elementen sowie Nutzer-definierte Funktionen.

Spark läuft in einer Master-Worker-Architektur. Für einen Lauf wird auf dem Mas-

ter-Knoten eine neue JVM gestartet, die als Driver fungiert und den Ablauf steuert. Für Tasks, die parallel ablaufen können, werden auf den Worker-Knoten JVMs gestartet, sogenannte „Executor“. Innerhalb eines Executor können mehrere Tasks parallel in verschiedenen Threads laufen, wobei diese Threads auf denselben Daten arbeiten können, während für den Austausch zwischen JVMs ein Datentransfer nötig ist. Anwendungen sind in Java, Scala, Python oder R möglich.

In [2] ist ein Beispiel zur Nutzung der DAX in Java beschrieben: Ein dreidimensionaler Raum ist in 10 x 10 x 10 Zellen aufgeteilt, deren Kantenlängen jeweils ein Zehntel des Wertebereichs der jeweiligen Dimension umfasst. Für eine Million Datenpunkte soll deren Zahl in jeder Zelle ermittelt werden. Durch Nutzung der DAX ist die Laufzeit um einen Faktor sechs gegenüber einer konventionellen Implementierung reduziert. Die Eingabetripel werden zunächst in ein Array eingelesen, in dem jede Spalte den Wert in einer Dimension enthält. Aus jeder dieser Spalten wird RDD erzeugt und daraus ein Vektor für einen DAX, in dem über Scan-Operationen für die jeweils zehn Wertebereiche der Zellen zehn Bit-Vektoren erzeugt werden, wobei ein gesetztes Bit bedeutet, dass in dieser Dimension der Wert des Eingabe-Tripels in den Wertebereich der Zelle fällt. Das ergibt für die drei Dimensionen dreißig Bit-Vektoren. Für jede Zelle werden nun jeweils die drei relevanten Bit-Vektoren durch ein logisches „UND“ kombiniert sowie im resultierenden Bit-Vektor die Zahl der gesetzten Bits und damit die Zahl der Tripel gezählt, die in diese Zelle fallen.

Programmtechnisch werden zunächst für jede Zelle einer Schicht jeweils die der beiden Dimensionen zu einem Bitvektor kombiniert und dieser dann iterativ jeweils mit den Bit-Vektoren der dritten Dimension zum finalen für die jeweilige Zelle. Die gesetzten Bits können dann sehr schnell in Hardware gezählt werden. Der Quellcode dazu ist unter [7] zu finden.

Auf der Oracle OpenWorld 2016 wurden in einer Präsentation [9] erste Ergebnisse der Nutzung von OpenDAX in weiteren Anwendungen vorgestellt: Apache Arrow sowie seitens des ISV ActiveViam (Quartet FS).

Turbo für Java-Streams

Das Streams-API war in Java 8 eingeführt worden. Damit ist es möglich, Operatio-

```
private static int Hotdays_count (ArrayList<Integer> Temperature_list)
{
    return (Temperature_list.parallelStream().filter(t->t>30).count());
}
```

Listing 1

nen auf Daten in einer Art Pipeline zu spezifizieren, ohne explizit über die Daten zu iterieren. Damit wird nicht nur der Code leichter lesbar, er ist insbesondere auch leichter automatisch parallelisierbar. Im folgenden Beispiel wird aus einer Liste mit Temperaturwerten die Zahl der heißen Tage (über 30 Grad) ermittelt (siehe Listing 1).

Hier liegt es nahe, für eine leistungsstarke Implementierung die DAX zu nutzen. In der Software in Silicon Cloud gibt es eine solche Implementierung als Java-Jar-Bibliothek mit einem neuen Package „com.oracle.stream“, das die Schnittstellen des Stream-API implementiert und für Funktionen wie „integer stream filter“, „allMatch“, „anyMatch“, „nonMatch“, „map“, „count“ und „toArray“ den DAX nutzt [3]. Auf Systemen ohne DAX fällt die Bibliothek auf die Standard-Implementierung zurück. Im Beispiel, bei dem Werte in einem Wertebereich gezählt werden, ergibt sich eine Beschleunigung um den Faktor zehn – andere Beispiele liefern Faktoren zwischen drei und zweiundzwanzig.

Fazit

Wer nun selbst mit den OpenDAX-APIs experimentieren will oder sich auch nur die Codebeispiele, Sprachbindungen oder die Dokumentation genauer anschauen möchte, kann dies auch ohne eigenes M7-, T7- oder S7-System nach einer unentgeltli-

chen Registrierung in der Software in Silicon Developer Cloud tun [12].

Literatur

- [1] Franz Haberhauer, Sicherheit und SQL in Silizium, Red Stack Magazin 01/2016, S. 63-67, Februar 2016: https://www.doag.org/formes/pub-files/7832746/docs/Publikationen/DOAGNews/2016/01-2016/2016-01-Red-Stack-Magazin-Franz-Haberhauer_Sicherheit-und-SQL-in-Silizium.pdf
- [2] Suman Somasundar. Apache Spark and SPARC M7 Data Analytics Accelerator, Oktober 2016: <https://community.oracle.com/docs/DOC-994843>
- [3] Karthik Ganesan, Accelerating Java Streams with the SPARC Data Analytics Accelerator, Oktober 2016: <https://community.oracle.com/docs/DOC-1006352>
- [4] Martin Müller, Low-level monitoring of DAX on SPARC M7, Blog, 28. Oktober 2016: https://blogs.oracle.com/martinm/entry/low_level_monitoring_of_dax
- [5] Martin Müller, High Level Monitoring of DAX Activity, Blog, 31. Oktober 2016: https://blogs.oracle.com/martinm/entry/high_level_monitoring_of_dax
- [6] Greg Jumper, Kumaraswamy, Introduction to Stream Processing Using the DAX API, August 2016: <https://community.oracle.com/docs/DOC-998150>
- [7] DAX Sample Programs, Software in Silicon Developer Cloud: <https://swisdev.oracle.com/DAX/DAXsampleprograms.php>
- [8] Apache Spark – Lightning-fast cluster computing: <http://spark.apache.org>
- [9] Amir Javanshir, Sanjayrao Hemade, Allen Whipple, Accelerating Apache Arrow and

QuartetFS ActivePivot with Software in Silicon, Vortrag Oracle OpenWorld 2016 [CON6383], 22. September 2016:

- [https://oracle.rainfocus.com/scripts/catalog/ow16.jsp?search=\[CON6383\]&search.event=oracleopenworld&search.event=javaone, Vortragsfolien: https://static.rainfocus.com/oracle/ow16/session/1464245216959001h4ls/ppt/CON6383-Accelerating%20Apache%20Arrow%20and%20Quartet%20FS%20ActivePivot%20with%20Software%20in%20Silicon.pdf](https://oracle.rainfocus.com/scripts/catalog/ow16.jsp?search=[CON6383]&search.event=oracleopenworld&search.event=javaone, Vortragsfolien: https://static.rainfocus.com/oracle/ow16/session/1464245216959001h4ls/ppt/CON6383-Accelerating%20Apache%20Arrow%20and%20Quartet%20FS%20ActivePivot%20with%20Software%20in%20Silicon.pdf)
- [10] BestPerf Blog, Accelerating Java Streams Performance Using SPARC with DAX, 1.12.2016: https://blogs.oracle.com/BestPerf/entry/20161201_java_streams_and_dax
 - [11] Vijay Tatkar, What Is the SPARC M7 Data Analytics Accelerator?, März 2016, <https://community.oracle.com/docs/DOC-994842>
 - [12] DAX Sample Programs, Software in Silicon Developer Cloud: <https://swisdev.oracle.com/DAX>



Franz Haberhauer
franz.haberhauer@oracle.com

Boomendes Cloud-Geschäft hält den Oracle-Konzernumsatz stabil

Der US-Konzern hat im vergangenen Quartal mit dem Verkauf seiner Cloud-Lösungen erstmals Erlöse von mehr als einer Milliarde US-Dollar erzielt. Dennoch ist der Quartalsgewinn zurückgegangen. Das traditionelle Geschäft mit klassischer Software und Hardware schwächelt weiterhin.

Wie Oracle mitgeteilt hat, schloss das kalifornische Unternehmen das zweite Geschäftsquartal mit einem Gewinn von 2,03 Milliarden US-Dollar ab. Das entspricht im Vergleich zum entsprechenden Vorjahreszeitraum einem Rückgang von acht Prozent. Als Gründe hierfür nennt

Oracle den starken Dollar und einen unvorhergesehenen Verlust auf die ägyptische Währung.

Deutlich schwerer dürften allerdings die weiterhin starken Einbußen in den klassischen Geschäftsbereichen „Software-Lizenzen“ und „Wartung“ wiegen: das Software-Geschäft sank um 4 Prozent auf 6,1 Milliarden US-Dollar, das Geschäft mit den Software-Lizenzen ging um 20 Prozent auf 1,35 Milliarden US-Dollar zurück. Auch im Hardware-Bereich musste das Unternehmen einen Rückgang um 10 Prozent auf 1 Milliarde US-Dollar hinnehmen.

Insgesamt konnte Oracle aufgrund des gut laufenden Cloud-Geschäfts einen leichten Umsatzzuwachs auf 9,07 Milliarden US-Dollar verzeichnen. Die Erlöse im Cloud-Bereich waren insgesamt um 62 Prozent auf 1,05 Milliarden US-Dollar angestiegen, aufgeteilt auf 878 Millionen US-Dollar für Software-as-a-Service und Platform-as-a-Service sowie 175 Millionen US-Dollar für Infrastructure-as-a-Service. Durch die starken Zuwächse blieb der Konzernumsatz damit zwar stabil, Analysten hatten sich aber etwas mehr erwartet, weshalb die Oracle-Aktie um rund zwei Prozent nachgab.



Datenbank-zentrische Anwendungen mit Spring Boot und jOOQ

Michael Simons, ENERKO INFORMATIK GmbH

Wie können Datenbank-zentrische, aber nicht Datenbank-abhängige Anwendungen mit modernen Frameworks entwickelt werden? jOOQ steht für „Java Object Oriented Querying“ und beschreibt ein Framework zum Erstellen von Abfragen – ein „query builder“-Framework. Zusammen mit Spring Boot ist es vielleicht die ideale Möglichkeit, HTTP-APIS für analytische Abfragen zu entwickeln.

Hinweis: Der Quelltext und die vollständige Demo dieses Artikels stehen unter [1] zur Verfügung.

Die ersten Gehversuche des Autors auf der Suche nach einer Alternative zu Oracle Forms 6 endeten vor vielen Jahren sehr schnell in einem ernüchternden

Dschungel aus XML-Konfiguration, kaum zu bändigenden Abhängigkeiten und aufwändigen Deployments. Die erste Java-basierte Web-Anwendung, die er entwickeln durfte, wurde mit einer XML-Datei ähnlich der in *Abbildung 1* konfiguriert. Egal ob Java EE oder Spring-Framework: Bis zum Jahr 2007 kam man um explizite Konfiguration mit XML nicht herum. Zu diesem Zeitpunkt war XML die einzige Möglichkeit, das Spring-Framework vollständig zu konfigurieren.

Der erste Commit im Spring-Boot-Projekt stammt aus dem Jahr 2013. Spring Boot setzt seit der ersten öffentlichen Version konsequent auf intelligente Konventionen und Defaults, Verzicht auf XML-Konfiguration und optimalen Einsatz von Konfiguration durch Java-Code. *Listing 2* nimmt – sofern die Abhängigkeiten im Klassenpfad definiert sind – die gleiche Konfiguration wie die XML-Datei vor.

Viele Open-Source-Komponenten wie Spring, Tomcat, JPA/Hibernate und verschiedenste Datenbanken haben eine extrem hohe Qualität erreicht und die Probleme sind oftmals anders gelagert: Wie lassen sich übliche funktionale und nicht-

```
...<property name="failbackSystemScale" value="false"/>
</bean>
<!-- Configure the JPA Adapter -->
<bean id="jpaDialect" class="org.springframework.orm.jpa.vendor.HibernateJpaDialect"/>
<bean id="jpaVendorAdapter" class="org.springframework.orm.jpa.vendor.HibernateJpaVendorAdapter">
  <property name="database" value="HSQL"/>
  <property name="databasePlatform" value="org.hibernate.dialect.WySQLDialect"/>
  <property name="generateDdl" value="false"/>
  <property name="showSql" value="false"/>
</bean>
<!-- Configure the local Entity Manager Factory -->
<bean id="entityManagerFactory"
  class="org.springframework.orm.jpa.LocalContainerEntityManagerFactoryBean">
  <property name="persistenceUnitName" value="theEntities"/>
  <property name="dataSource" ref="theDataSource"/>
  <property name="jpaDialect" ref="jpaDialect"/>
  <property name="jpaVendorAdapter" ref="jpaVendorAdapter"/>
  <property name="loadTimeWeaver"/>
  <bean class="org.springframework.instrument.classloading.InstrumentationLoadTimeWeaver"/>
</property>
</bean>
<!-- Enable Spring JPA Transactions -->
<bean id="transactionManager" class="org.springframework.orm.jpa.JpaTransactionManager" p:entity-
manager-factory-ref="entityManagerFactory"/>
<tx:annotation-driven transaction-manager="transactionManager"/>
<bean id="exampleBean" class="examples.ExampleBean">
  <!-- setter injection using the nested <ref/> element -->
  <property name="beanOne">
    <ref bean="anotherExampleBean"/>
  </property>
  <!-- setter injection using the neater 'ref' attribute -->
  <property name="beanTwo" ref="yetAnotherBean"/>
  <property name="integerProperty" value="1"/>
</bean>
<bean id="anotherExampleBean" class="examples.AnotherBean"/>
<bean id="yetAnotherBean" class="examples.YetAnotherBean"/>
</beans>
```

A long time ago, in a framework far, far away

Abbildung 1: A long time ago, in a framework far, far away

funktionale Herausforderungen lösen, die einer effizienten Produktiv-Setzung im Weg stehen? Konsistent in Bezug auf Konfiguration, Entwicklung und Deployment? Spring Boot ist angetreten, um genau diese Fragen zu beantworten. Dabei stellt es fertig konfigurierte Instanzen des Spring-Frameworks mit „opinionated Defaults“ bereit. Spring Boot unterstützt dabei nicht nur das Spring-Framework, den Inversion of Control Container, Beans und das MVC-Framework, sondern ein riesiges Ökosystem weiterer Technologien und Konzepte wie Spring Security, Caches, Enterprise Integration Pattern und mehr.

Diese Unterstützung kommt in Form sogenannter „Starter“. Dies sind Standard-Java-Archive, die als Abhängigkeiten im Build-System deklariert sind. Sie bringen automatische Konfiguration sowie benötigte Libraries mit und erlauben eine gezielte Auswahl von unterstützten Technologien, unter anderem auch von jOOQ. Wichtig dabei ist, dass Spring Boot an keiner Stelle einen Code-Generator einsetzt. Andernfalls wäre es schwierig, Konfiguration zu ändern oder zu ersetzen. Falls es notwendig ist, kann übrigens immer noch XML zur Konfiguration genutzt werden.

Alles in allem ermöglicht Spring Boot eine bestmögliche Out-of-the-box-Erfahrung mit dem Spring-Ökosystem. Um konsistente Projektdefinitionen zu gewährleisten, gibt es mit dem Spring Initializr unter [2] eine Web-Anwendung, die es nach Angabe von Namen und Projekt-Koordinaten sowie der Wahl des Build-Systems ermöglicht, fertige Projekte inklusive eines Build-Files und einer minimalen Anwendung herunterzuladen.

Listing 1 und 2 zeigen das Skelett einer „Hallo, Welt“-Anwendung, die eine URL zur Verfügung stellt und den Aufrufer grüßt. Listing 1 ist der Build File des Tools Maven [3] und definiert unter anderem die Liste der Abhängigkeiten, die Sprachversion (Java 8) und die Projekt-Koordinaten. In Listing 2 wird die eigentliche Anwendung definiert, die aus einer Hauptklasse („Application“), annotiert mit „@SpringBootApplication“, sowie einem Rest-Controller („HaloWorldController“) besteht.

Dessen Methode „#greeting“ steht unter der URL „/hallo“ zur Verfügung und wertet Parameter aus, die entweder, wie im Beispiel, aus den Request-Query-Parametern oder auch aus URL-Pfaden stammen können. Wird dieses Java-Programm

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://
www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://ma-
ven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>

  <groupId>org.doag</groupId>
  <artifactId>redstack</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <packaging>jar</packaging>

  <name>DOAG Red Stack demo</name>
  <description>Demo project for Spring Boot</description>

  <parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>1.4.2.RELEASE</version>
    <relativePath/> <!-- lookup parent from repository -->
  </parent>

  <properties>
    <java.version>1.8</java.version>
  </properties>

  <dependencies>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-web</artifactId>
    </dependency>
  </dependencies>

  <build>
    <plugins>
      <plugin>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-maven-plugin</artifactId>
      </plugin>
    </plugins>
  </build>
</project>
```

Listing 1: „pom.xml“

```
package org.doag.redstack;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

@SpringBootApplication
public class Application {

    @RestController
    public static class HalloWorldController {

        @GetMapping("/hallo")
        public String greeting(@RequestParam final String name) {
            return "Hallo, " + name + "\n";
        }
    }

    public static void main(String[] args) {
        SpringApplication.run(Application.class, args);
    }
}
```

Listing 2: Application.java „A new hope“

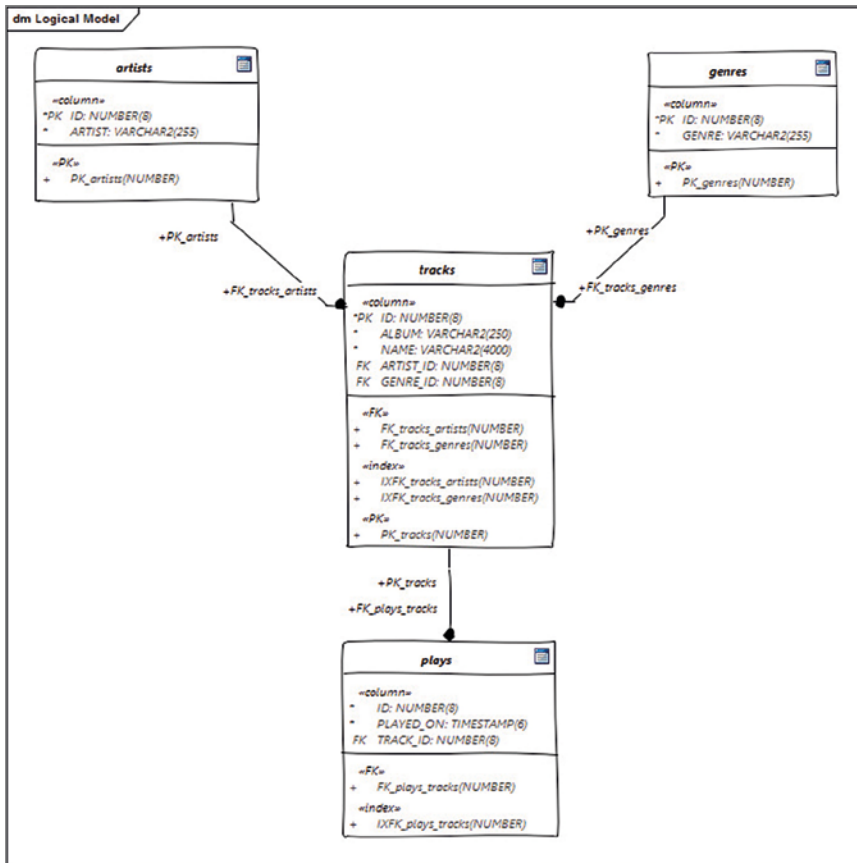


Abbildung 2: Datenmodell der Demo

```

Select *
  from tracks
 where album = 'True Survivor';
    
```

Listing 3: Triviale Abfrage

```

@Entity
@Table(name = "tracks")
public class TrackEntity implements Serializable {

    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    private Integer id;

    @Column
    private String album;

    public static void main(String...a) {
        final EntityManagerFactory factory = Persistence.createEntityManagerFactory("whatever");
        final EntityManager entityManager = factory.createEntityManager();

        List<Track> tracks = entityManager.createQuery("Select t from tracks where album = :album")
            .setParameter("album", "True Survivor")
            .getResultList();
    }
}
    
```

Listing 4: Triviale Abfrage mit Hibernate

gestartet, ergibt der Aufruf von „http://localhost:8080/hallo?name=DOAG“ das Resultat „Hallo, DOAG“.

Dieses Projekt kann man mit Maven bauen. Es entsteht ein ausführbares Java-Archiv vom Typ „.jar“, das alle benötigten Teile des Spring-Frameworks, Spring Boot und einen eingebetteten Application Container (Apache Tomcat) enthält. Das Archiv ist knapp 18 MB groß und kann überall dort gestartet werden, wo Java zur Verfügung steht.

Spring Boot: Fazit

Mit Spring Boot wird das Spring-Framework extrem zugänglich. Sowohl für Entwickler, die sich sehr lange nicht im Java-Umfeld bewegt, zum Beispiel lange im Bereich PL/SQL oder in der Forms-Entwicklung tätig waren, als auch für Menschen, die noch nicht lange im Alltag mit Software-Entwicklung zu tun haben. Da der Fokus nicht mehr auf einer aufwändigen Konfiguration liegt, kann man sich auf interessante Fragen konzentrieren: auf Fachlichkeit, Software-Architektur, Datenstrukturen oder auch darauf, modernes Java oder SQL zu lernen.

Java und Datenbanken: Plain SQL, ORM oder etwas dazwischen?

Gegeben sei ein Datenmodell wie in *Abbildung 2*. Es handelt sich um ein Datenmodell zur Erfassung von Musik. In vier Tabellen werden Titel, Interpreten, Genre und Wiedergaben der Titel erfasst. Wer genau hinschaut, sieht, dass das Datenmodell nicht in dritter Normalform vorliegt, das Album wird bei den Titeln gespeichert. Die Daten aus diesem Modell sollen in einer Anwendung als Charts angezeigt werden.

Eine Abfrage auf die Tabelle der Titel ist trivial (*siehe Listing 3*). Diese Abfrage kann natürlich auch mit einem objektrelationalen Mapping, zum Beispiel mit Hibernate, ausgeführt werden (*siehe Listing 4*). Direkt wird aber deutlich, dass selbst eine einfache Abfrage mehr „Boilerplate“-Code benötigt. Deutlich besser handhabbar wird sie, wenn man Spring Data JPA einsetzt (*siehe Listing 5*). Was aber tun, wenn die Anforderung lautet: Gib mir die Top-40-Titel des letzten Monats, absteigend sortiert nach Anzahl der Wiedergabe inklusive ei-

ner Position sowie der Änderung der Position im Vergleich zur Vorwoche? Mithilfe moderner SQL-Konstrukte ist dieser Geschäftsfall mit einer Oracle-Datenbank und obigem Datenmodell sogar recht leicht zu beantworten (siehe Listing 6).

Diese Abfrage enthält mehrere Common-Table-Expressions sowie Window-Funktionen. Man könnte diese Abfrage nun in Java-Annotationen verstecken oder über eine PL/SQL-Funktion aufrufen: In allen Fällen würde man versuchen, den Programmierer vor dem SQL-Code zu schützen.

jOOQ arbeitet anders. Das Ziel ist nicht, obigen SQL-Code zu verstecken – im Gegenteil. jOOQ ist zuerst ein Code-Generator, der aus dem Datenbank-Schema Zugriffsklassen für eine Domain Specific Language (DSL) generiert. Diese DSL beinhaltet in erster Linie nur die Fachlichkeit, die durch das Datenmodell vorgegeben ist, und keine Architekturvorgabe. Ihr Ziel ist es, SQL zu schreiben. Listing 7 zeigt obige Abfrage typischer mit jOOQ ohne Annotationen.

jOOQ steht für „Java Object Oriented Querying“ und ist laut eigener Aussage ein Framework zur Erzeugung von Abfragen. Es stellt für unterschiedliche Schemas eine Domain-spezifische Sprache (DSL) zur Generierung Datenbank-spezifischer Statements zur Verfügung. jOOQ lässt sich auch ohne Schema nutzen, allerdings ist es dann nur noch in Hinblick auf die Verwendung von SQL-Sprachmitteln, nicht aber in Hinblick auf Tabellen und Spalten typischer.

Das Datenbank-Schema ist also in der Regel die treibende Kraft. jOOQ wird in einem frühen Stadium in den Build-Prozess eingebunden, im Falle von Maven in der „generate-sources“-Phase. Es kann konfiguriert werden, welche Datenbank-Objekte (jOOQ unterstützt Tabellen, Prozeduren, Funktionen, Packages und mehr) betrachtet werden, welches Zielverzeichnis genutzt wird und welche zusätzlichen POJOs und Data-Access-Objekte generiert werden. Während des Build-Prozesses wird das Datenbank-Schema dann gelesen und ein Java-Pendant generiert. Teile davon sieht man in Listing 7. In der „from“-Klausel wird die Konstante „TRACKS“ genutzt, die typischer die gleichnamige Tabelle beschreibt.

Datenbank-Migrationen

Strebt man eine möglichst reibungsfreie Entwicklung an, eventuell noch mit dem

```
public interface TrackRepository extends JpaRepository<TrackEntity, Integer> {
    public List<Track> findAllByAlbum(final String name);

    public static void main(String...a) {
        TrackRepository trackRepository;
        final List<Track> tracks = trackRepository.findAllByAlbum("True Survivor");
    }
}
```

Listing 5: Triviale Abfrage mit Hibernate und Spring Data JPA

```
WITH
    previous_month AS
        (SELECT p.track_id, count(*) as cnt,
            dense_rank() over(order by count(*) desc) as position
         FROM plays p
         WHERE trunc(p.played_on, 'DD') BETWEEN
            date'2016-04-01' and date'2016-04-30' GROUP BY p.track_id),
    current_month AS
        (SELECT p.track_id, count(*) as cnt,
            dense_rank() over(order by count(*) desc) as position
         FROM plays p
         WHERE trunc(p.played_on, 'DD') BETWEEN
            date'2016-05-01' and date'2016-05-31' GROUP BY p.track_id)
SELECT a.artist || ' - ' || t.name || ' (' || t.album || ')' as label,
    current_month.cnt,
    previous_month.position - current_month.position as change
FROM tracks t
JOIN artists a on a.id = t.artist_id
JOIN current_month current_month on current_month.track_id = t.id
LEFT OUTER join previous_month on previous_month.track_id = t.id
ORDER BY current_month.cnt desc, label asc
FETCH FIRST 20 ROWS ONLY;
```

Listing 6: Komplexe SQL-Abfrage

```
this.create
    .with(currentMonth)
    .with(previousMonth)
    .select(label,
        currentMonth.field("cnt"),
        previousMonth.field("position").minus(
            currentMonth.field("position")
        ).as("change")
    )
    .from(TRACKS)
    .join(ARTISTS).onKey()
    .join(currentMonth)
    .on(currentMonth.field("track_id", BigDecimal.class)
        .eq(TRACKS.ID))
    .leftOuterJoin(previousMonth)
    .on(previousMonth.field("track_id", BigDecimal.class)
        .eq(TRACKS.ID))
    .orderBy(currentMonth.field("cnt").desc(), label.asc())
    .limit(n)
    .fetch()
    .formatJSON(response.getOutputStream());
```

Listing 7: Komplexe Abfrage mit jOOQ in Java

Ziel, Continuous Delivery zu erreichen, sind Datenbank-Migrationen essenziell. Als „Datenbank-Migration“ wird der

kontrollierte Prozess bezeichnet, eine bekannte Schema-Version auf eine neue Version zu heben. Dabei können zum Bei-

spiel Tabellen angelegt oder geändert sowie neue Trigger oder Ähnliches eingeführt werden. Spring Boot unterstützt zwei Werkzeuge ohne weitere Eingriffe: Liquibase und Flyway. In der verlinkten Demo wird Flyway eingesetzt [4].

Die grundsätzliche Idee ist folgende: Die Datenbank-Migration wird gegen eine Entwicklungs-Datenbank durchgeführt, die idealerweise jedem Entwickler separat zur Verfügung steht. Dann erst wird die Generierung des jOOQ-Schemas angestoßen und anschließend der Java-Quelltext kompiliert. So ist sichergestellt, dass nur gegen ein aktuelles Schema entwickelt wird. All dies passiert noch im Build-Prozess, also ohne Unterstützung von Spring Boot. Werden nun Feature-Branche gemischt, kann es vorkommen, dass auch Migrationen gemischt werden müssen. Konflikte fallen an diesen Punkt auf. Wird dann die ausgelieferte Anwendung das erste Mal gegen die Produktionsdatenbank gestartet, sorgt Spring Boot dafür, dass die Migration ebenfalls gegen die Produktions-Datenbank gestartet wird. Damit passt das jOOQ-Schema dann auch zur Produktions-Datenbank.

Abbildung analytischer Abfragen auf URLs

Sobald in einem Spring-Boot-Projekt der „spring-boot-starter-jooq“-Starter eingebunden ist, steht jOOQ mit einer Instanz der Klasse „DSLContext“ zur Verfügung. Der Lesbarkeit halber empfiehlt der Hersteller, diese Variable „create“ zu nennen. Entwickelt man für eine Oracle-Datenbank, sind ebenfalls der entsprechende JDBC-Treiber und die kommerzielle Version von jOOQ zur Verfügung zu stellen.

In Listing 8 wird die URL „/{artistIds}/topNAlbums“ auf die Funktion „getTopNAlbums“ abgebildet. Dabei ist das Fragment „/{artistIds}“ ein URL-Parameter, der von Spring Boot typischer auf ein Array von Zahlen („BigDecimal[]“) abgebildet wird. Weitere Parameter stehen in Form von Query-Parametern zur Verfügung, zum Beispiel die Datumparameter „from“ and „to“, die beide typischer optional sind.

Mit diesen Informationen wird dann eine Abfrage erstellt und ausgeführt, um die Wiedergabe aller Alben im gegebenen Zeitraum oder im aktuellen Monat auszuführen. Das Ergebnis dieser Abfrage wird

direkt als JSON in den HTTP-Response geschrieben und kann von Clients weiterverarbeitet werden. Listing 9 zeigt die entspre-

chende Rückgabe in Auszügen. In [5] wird das Thema „HTTP-APIs“ für analytische Abfragen in aller Ausführlichkeit behandelt.

```
@RequestMapping(path =("/{artistIds}/topNAlbums")
public void getTopNAlbums(
    @PathVariable final BigDecimal[] artistIds,
    @RequestParam(defaultValue = "10") final int n,
    @RequestParam
    @DateTimeFormat(iso = ISO.DATE)
    final Optional<LocalDate> from,
    @RequestParam
    @DateTimeFormat(iso = ISO.DATE)
    final Optional<LocalDate> to,
    final HttpServletResponse response
) throws IOException {
    response.setContentType(MediaType.APPLICATION_JSON_UTF8_VALUE);
    this.create
        .select(TRACKS.ALBUM,
            count())
        .from(PLAYS)
        .join(TRACKS).onKey()
        .where(TRACKS.ARTIST_ID.in(artistIds))
        .and(from.map(Date::valueOf)
            .map(PLAYED_ON_TRUNCATED_TO_DAY::greaterOrEqual)
            .orElseGet(DSL::trueCondition)
        )
        .and(to.map(Date::valueOf)
            .map(PLAYED_ON_TRUNCATED_TO_DAY::lessOrEqual)
            .orElseGet(DSL::trueCondition)
        )
        .groupBy(TRACKS.ARTIST_ID, TRACKS.ALBUM)
        .orderBy(count().desc(), TRACKS.ALBUM.asc())
        .limit(n)
        .fetch()
        .formatJSON(response.getOutputStream());
}
```

Listing 8: Nutzung des DSL-Kontexts

```
{
  "fields": [
    {
      "schema": "",
      "table": "TRACKS",
      "name": "ALBUM",
      "type": "VARCHAR"
    },
    {
      "name": "count",
      "type": "INTEGER"
    }
  ],
  "records": [
    [
      "Everything Louder Than Everyone Else",
      151
    ],
    [
      "BBC Live & In-Session",
      68
    ]
  ]
}
```

Listing 9: Curl -X „GET“ „http://127.0.0.1:8080/api/artists/54,86/topNAlbums“

```

define(['ojs/ojcore', 'knockout', 'jquery', 'moment', 'ojs/ojselectcombobox', 'ojs/ojchart', 'ojs/ojdatetimepicker'],
function (oj, ko, $, moment) {
    function artistsContentViewModel() {
        var self = this;
        self.areaSeriesValue = ko.observableArray([]);
        self.areaGroupsValue = ko.observableArray([]);
        var updateCharts = function () {
            // Fill model
        }
    };

    self.optionChanged = function (event, data) {
        updateCharts();
    };
}
return new artistsContentViewModel();
});

```

Listing 10: JavaScript-Modell einer Oracle-JET-Anwendung

```

<h2>Cumulative plays per artist and day</h2>
<div id='chart-container'>
    <div id="lineAreaChart" style="max-width:1024px;width:100%;height:320px;" data-bind="ojComponent: {
        component: 'ojChart',
        type: 'lineWithArea',
        series: areaSeriesValue,
        groups: areaGroupsValue,
        timeAxisType: 'enabled',
        animationOnDisplay: 'on',
        animationOnDataChange: 'on',
        stack: 'on',
        hoverBehavior: 'dim',
        zoomAndScroll: 'live',
        overview: {rendered: 'off'},
        dataCursor: dataCursorValue
    }"></div>
</div>

```

Listing 11: Die HTML-View

Spring Boot und jOOQ: Fazit

jOOQ ist mit Spring Boot quasi trivial verwendbar; sowohl die notwendige Code-Generierung im Build-Prozess als auch das Boot-Strapping des jOOQ-Kontexts passen in das Entwicklungsmodell von Spring Boot. jOOQ zielt nach Ansicht des Autors auf eine SQL-basierte Zwei-Schichten-Architektur. Es ist möglich, das Repository-Muster anzuwenden, aber die Erfahrung spricht dagegen. jOOQ spielt seine Stärken dann aus, wenn es um die Modellierung komplexer Abfragen geht, nicht in möglichst hoher Abstraktion.

Akzeptiert man die zweischichtige Architektur, kann man sehr leicht vorhandenes Wissen im Bereich „Datenbank-Modellierung und Abfrage“ in moderne Software-Entwicklung einbringen. Zudem

kann man im Falle von kommerziellen Datenbanken alle Funktionen nutzen, für die man ja bereits bezahlt hat. Innerhalb einer Spring-Boot-Anwendung lässt sich jOOQ in nahezu beliebigen Cloud-Umgebungen einrichten.

jOOQ passt – wie viele andere Werkzeuge auch – nicht auf jeden Anwendungsfall. Liegt ein Domain-Driven-Design-Ansatz vor, sind komplexe Abfragen nicht maßgeblicher Bestandteil einer Anwendung oder ist der Schreibzugriff wichtiger, sprechen viele gute Gründe für den Einsatz eines ORM wie JPA/Hibernate, gerne auch zusammen mit Spring Data JPA. Ein hier skizzierter Anwendungsfall, quasi eine direkte Abbildung einer HTTP-Abfrage auf eine komplexe Query, lässt sich allerdings hervorragend mit jOOQ abbilden.

jOOQ ist kostenfrei nutzbar mit Open-Source-Datenbanken unter der Apache Software License 2.0. Für den Einsatz mit kommerziellen Datenbanken stehen verschiedene kommerzielle Lizenzen zur Verfügung.

Oracle JET: Die Antwort auf das JavaScript-Frontend-Dilemma

Der Schwerpunkt dieses Artikels liegt nicht auf dem Frontend, aber die besten Daten sind nur schlecht erfassbar ohne ansprechende Präsentation. Viele moderne Anwendungen werden „headless“ betrieben, ohne ein auf dem Server generiertes Web-Frontend. Wird eine UI erstellt, so führt im Jahr 2017 kaum noch

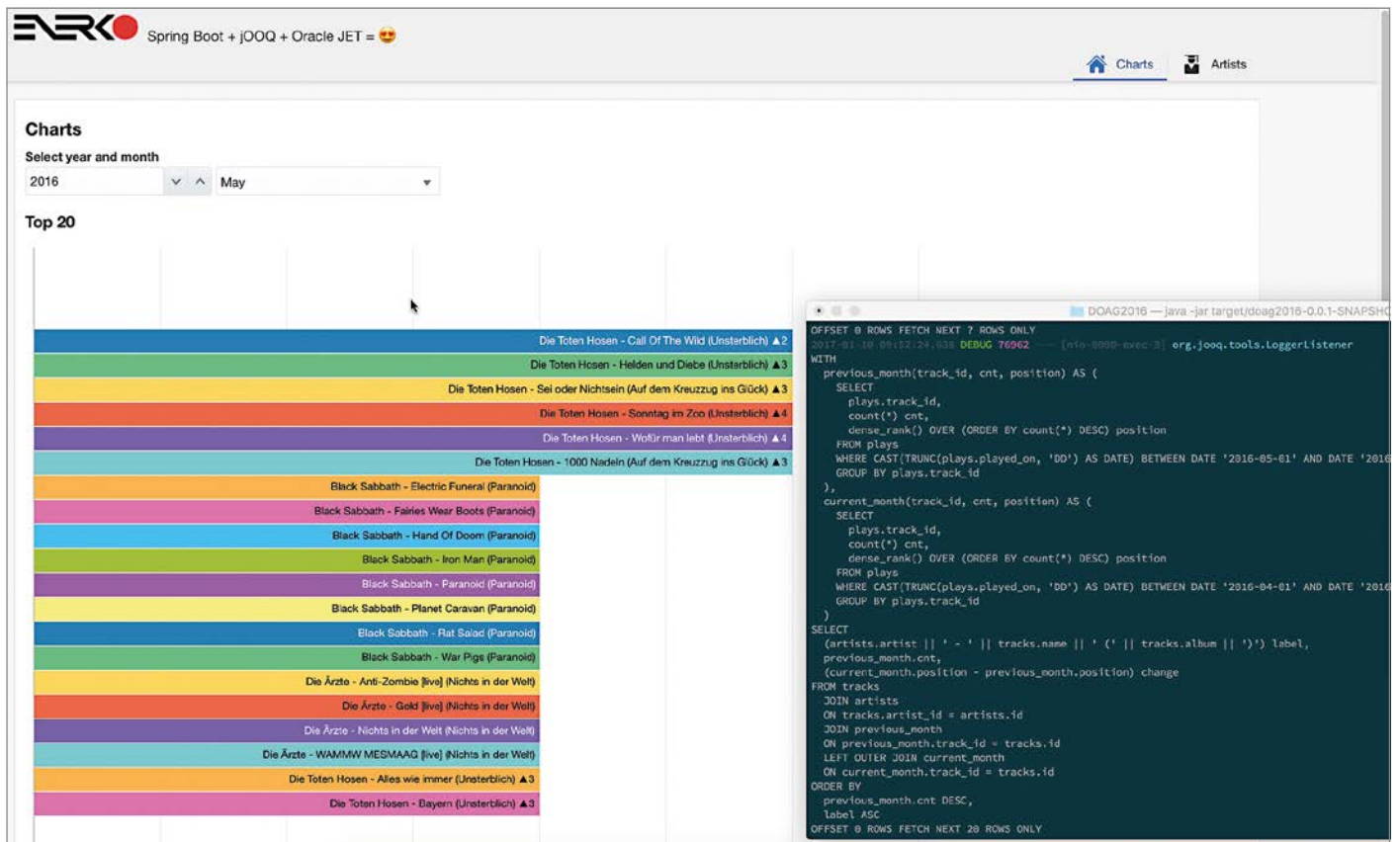


Abbildung 3: Oracle JET mit der Log-Ausgabe von Spring Boot und jOOQ

ein Weg an JavaScript und allem, was dazu gehört, vorbei. Dabei sind weitere Build-Tools und eine erschöpfende Auswahl an Frameworks beteiligt.

Aus diesem Dilemma gibt es mehrere Auswege: Man wählt ein Komponenten-Framework wie eine beliebige JSF-Implementierung, GWT oder Vaadin, das auf Seiten des Servers HTML und JavaScript-Komponenten erzeugt. Solange man keine speziellen Anpassungen benötigt, wird der Entwickler gut bis sehr gut vor UI-JavaScript geschützt. Ganz schnell läuft man aber Gefahr, dass scheinbar einfache Kundenanforderungen sehr in den generierten Client-Code eingreifen, sodass die hohe Abstraktion mehr schadet als nutzt. Einen anderen Ausweg bietet die Zusammenstellung eigener Sammlungen von Komponenten oder der Entwurf eigener Frameworks (vermutlich die Quelle vieler auf „*.js“ endender Frameworks). In beiden Fällen läuft man Gefahr, vom Wartungsaufwand überrannt zu werden.

Oracle JavaScript Extension Toolkit (JET) wählt genau den Ansatz intelligent zusammengestellter Komponenten, um Entwicklern eine Sammlung von Werkzeugen und modulare Anwendungen auf Basis moder-

ner JavaScript-, CSS3- und HTML5-Prinzipien zur Verfügung zu stellen. Es ermöglicht auf Basis bekannter Konzepte und Komponenten die Erstellung von modernen Enterprise-Anwendungen sowohl für den Desktop als auch für mobile Endgeräte.

Oracle JET greift auch die Idee einer „Bill of Materials“ auf: Es nutzt RequireJS für das Management von Modulen und Abhängigkeiten, KnockoutJS zur Implementierung des Model-View-ViewModel-Konzepts (MVVM) sowie jQuery und jQuery-UI für moderne, ansprechende UI-Komponenten. Diese Komponenten wurden mit dem Ziel erstellt, barrierefreien Zugriff zu ermöglichen sowie vollständig internationalisierbar zu sein. *Listing 10 und 11* zeigen das Modell und die View einer Anwendung, die das Ergebnis einer mit jOOQ realisierten SQL-Abfrage anzeigt, die auf eine HTTP-URL abgebildet wurde (*siehe Abbildung 3*).

Das Ziel der Demo war es, die über jOOQ mit SQL selektierten Daten mit Spring Boot einem Frontend zur Verfügung zu stellen und dort darzustellen:

- Als Balken-Diagramm
- Als kumulatives Linien-Diagramm
- Mit ansprechender, zugänglicher Ein-

gabemöglichkeit für Parameter

- Grundsätzlich internationalisierbar

Laut eigener Aussage zielt Oracle mit dem JET-Framework auf „intermediate to advanced JavaScript developers“. Mit grundsätzlich vorhandenem Wissen zu JavaScript, jQuery und der MVVM-Implementierung KnockoutJS gelang es, obige Anforderungen an die Demo einfach umzusetzen, die Aufgabe ließ sich zusammen mit dem Oracle JET Cookbook [12] in wenigen Stunden realisieren.

Oracle JET: Fazit

Die Demo ist eine hübsche Spielerei, beinhaltet aber viele Dinge, die im Geschäftsalltag benötigt werden. Die Technologie ist zugänglich und nach erster Einschätzung auch zukunftssicher, findet man sie doch bereits in Teilen in Oracle Apex wieder.

Abschließendes Fazit

Steht man vor der Entscheidung, neue Produkte auf Basis bestehender, großer Daten-

modelle zu entwickeln und gegebenenfalls Rücksicht auf alte Anwendungen nehmen zu müssen, ist der hier vorgestellte Stack eine Möglichkeit, vorhandenes Wissen über SQL und Datenbanken mit auf eine moderne Plattform zu nehmen und mit einem ansprechenden Frontend zu versehen.

Es entstehen leicht verteilbare Anwendungen beziehungsweise Microservices, die Datenbank-zentrisch, aber nicht Datenbank-abhängig sind. Vorhandene Datenbank-Technik und insbesondere vorhandenes Datenbank-Wissen bleiben erhalten und können genutzt werden.

Soll eine Oberfläche entstehen, ist das verhältnismäßig einfach mit Oracle JET möglich. Da die hier vorgestellte Abbildung analytischer Abfragen auf HTTP-Schnittstellen JSON als Transport-Format nutzt, kann auch nahezu jede beliebige andere Frontend-Technologie des Webs genutzt werden.

Das Unternehmen des Autors setzt seit mehr als einem Jahr die Kombination aus Spring Boot und jOOQ erfolgreich in ihren Produkten ein. Man verzichtet nicht gänzlich auf objektrelationale Abbildun-

gen, setzt aber gezielt SQL-Abfragen in einer Art und Weise ein, die sowohl im Java- als auch im SQL-Umfeld gleichermaßen gut zu verstehen und zu nutzen ist.

Am Ende sind sowohl SQL und objektrelationale Abbildungen als auch jOOQ Experten-Werkzeuge. Ohne entsprechendes Wissen können in der Regel nicht alle Möglichkeiten genutzt oder nicht gut eingesetzt werden. Eine Lektüre der Theorie, der Handbücher und auch einiger sehr lesenswerter Blogs wird daher dringend empfohlen. Nachfolgend sind einige davon genannt.

Quellen und Links

- [1] Vollständige Demo: <https://github.com/michael-simons/DOAG2016>
- [2] Spring Initializr: <http://start.spring.io>
- [3] Maven: <http://maven.apache.org>
- [4] Take control of your development databases evolution: <http://info.michael-simons.eu/2016/10/31/take-control-of-your-development-databases-evolution/>
- [5] An HTTP-API for analytic queries: <http://info.michael-simons.eu/2016/11/02/an-http-api-for-analytic-queries>

- [6] Euregio JUG: <http://www.euregjug.eu>
- [7] ENERKO Informatik: <http://www.enerko-informatik.de>
- [8] info.michael-simons.eu: <http://info.michael-simons.eu>
- [9] Spring Boot: <http://projects.spring.io/spring-boot>
- [10] jOOQ: <http://www.jooq.org>
- [11] Oracle JET: <http://www.oracle.com/webfolder/technetwork/jet/index.html>
- [12] Oracle JET Cookbook: <http://www.oracle.com/webfolder/technetwork/jet/jetCookbook.html>
- [13] Flyway by Boxfuse: <https://flywaydb.org>
- [14] Spring Data JPA: <http://projects.spring.io/spring-data-jpa>
- [15] Modern SQL: <https://modern-sql.com>
- [16] Vlad on Hibernate: <https://vladmihalcea.com/tutorials/hibernate>



Michael Simons
michael@simons.ac

ORACLE Platinum Partner

robotron
datenbank-software

Robotron-Fokustag Datenbanken



Themen

- ▶ Datenbank 12c Release 2 – die neuesten Features
- ▶ Oracle-Lizenzierung leicht gemacht
- ▶ Oracle Database Appliance – das Multitalent
- ▶ Datenbanken in der Oracle Cloud

Termine

- Dresden**
Dienstag, 07.03.2017 10:00 - 15:00 Uhr
- Wil (Schweiz)**
Dienstag, 14.03.2017 13:00 - 18:00 Uhr
- Stuttgart**
Mittwoch, 15.03.2017 10:00 - 15:00 Uhr



Robotron Datenbank-Software GmbH
Wir freuen uns auf Ihren Besuch!

Anmeldung und Informationen unter:
www.robotron.de/fokustag



Cloud@Customer mit der Exadata Cloud Machine

Frank Schneede, ORACLE Deutschland B. V. & Co. KG

Mit der Exadata Database Machine verfügen Kunden bereits seit Jahren über eine mit redundanten Komponenten ausgelegte Hochleistungsplattform für Unternehmens- und Performance-kritische Oracle-Datenbanken. Im Frühjahr 2016 wurde das Exadata-Angebot durch den Exadata Cloud Service erweitert, um Exadata-Technologie in der Oracle Public Cloud zu nutzen. Seit Mitte Dezember 2016 steht nun die dritte Säule der Exadata-Produktfamilie zur Verfügung – die Exadata Cloud Machine. Dieses Modell ist besonders für Kunden geeignet, die eine agile und flexible Cloud-Umgebung für Oracle-Datenbanken nutzen wollen, ohne jedoch die Public Cloud nutzen zu müssen.

Agilität und Flexibilität sind zwei wesentliche Argumente, die dazu beitragen, dass die Akzeptanz unterschiedlicher Cloud-Angebote immer mehr zunimmt. Es gibt jedoch für viele Kunden durchaus valide Gründe, weshalb ein Betrieb von Systemen in der Public Cloud nicht infrage kommt. So können gesetzliche Vorschriften oder unternehmensinterne Compliance-Richtlinien dagegen sprechen, dass Datenbanken in der Public Cloud außerhalb der eigenen Kontrolle betrieben werden.

Neben diesen rechtlichen Rahmenbedingungen kann es jedoch auch technische Gründe geben, die eine Public-Cloud-Strategie verhindern. Um einen nahtlosen und vor allem hochperformanten Zugriff auf die Public Cloud sicherzustellen, ist die Netzwerk-Verbindung zwischen dem eigenen IT-Standort und der Public Cloud essenziell. An vielen Standorten – gerade außerhalb der Ballungsräume – ist eine solche Anbindung entweder nicht verfügbar oder sehr kostenintensiv. Zudem muss sie den Anforderungen an die gewünschten SLAs der gesamten Lösung genügen, was leider häufig außer Acht gelassen wird.

Mit der Exadata Cloud Machine erhält der Kunde ein System, das physikalisch in seinem eigenen Sicherheits- und Zuständigkeitsbereich steht und das er somit unter voller Kontrolle hat. Es wird nach einem verursachungsgerechten Subskriptionsmodell abgerechnet und in einem festgelegten Umfang durch Oracle betrieben. Lastspitzen können, wie in einem Cloud-Ansatz üblich, durch einfaches Hinzubuchen von Rechenleistung abgefangen werden. Die Exadata Cloud Machine ist in die kundeneigene IT-Infrastruktur eingebunden und kann dort zum Beispiel in die vorhandene Backup-Infrastruktur integriert beziehungsweise hochperformant an eigene Applikationsserver angeschlossen werden.

Die Hardware

Die Exadata Cloud Machine basiert auf der aktuellen Exadata-X6-2-Generation und steht in den bekannten festen Ausbaustufen zur Verfügung, also als Eighth, Quarter, Half oder Full Rack. Die Datenbank-Server sind Zwei-Sockel-Systeme und enthalten

zwei aktuelle Twenty-two-Core-Intel-Xeon-E5-2699-v4-Prozessoren (2.2GHz). Der Hauptspeicher im Eighth Rack entspricht mit 256 GB der Standardkonfiguration, die auch das On-Premise-Modell enthält. In den größeren Ausbaustufen sind auf allen verfügbaren Steckplätzen der Datenbank-Server 32-GB-DIMM-Module verbaut, daher ist der Hauptspeicher mit 768 GB pro Datenbank-Server größer bemessen als beim On-Premise-Modell.

Die Konfiguration der Storage-Server entspricht der des On-Premise-Modells „Exadata Storage Server X6-2 High Capacity“, dieser ist mit zwei Ten-Core-Intel-Xeon-E5-2630-v4-Prozessoren (2.2GHz), zwölf 8 TB großen, Helium-befüllten Festplatten und vier der neuen 3.2 TB Flash Accelerator F320 PCIe Cards ausgestattet und verfügt über 128 GB Hauptspeicher. Die Extreme-Flash-Konfiguration steht nicht als Storage in der Exadata Cloud Machine zur Verfügung.

Es gibt kleine Unterschiede zwischen der On-Premise-Konfiguration der Exadata und der Exadata Cloud Machine. So ist die Anzahl der verbauten Storage-Server in

SERVICE METRICS	Eighth	Quarter	Half	Full
Number of Database Servers	2	2	4	8
Number of OCPUs	16 – 68	22 – 84	44 – 168	88 – 336
Total Memory (GB)	480 GB	1,440 GB	2,880 GB	5,760 GB
Number of Storage Servers	3	3	6	12
Total Flash Capacity	19.2 TB	38.4 TB	76.8 TB	153.6 TB
Total Usable Disk Capacity	42.7 TB	85.4 TB	170.9 TB	341.7 TB
Max. Total of all DB Sizes Supported (local backup)	17.1 TB	34.2 TB	68.4 TB	136.7 TB
Max. Total Of All DB Sizes Supported (no local backup)	34.2 TB	68.3 TB	136.7 TB	273.4 TB

Tabelle 1: Übersicht Exadata-Cloud-Machine-Modelle

Quarter, Half und Full Rack um einen Storage-Server gegenüber dem On-Premise-Modell verringert. Da die Exadata Cloud Machine mit OVM aufgesetzt wird, reduziert sich die nutzbare Größe der Hardware ebenfalls ein wenig. *Tabelle 1* gibt einen Überblick über die Konfiguration der Modelle und die nutzbaren Größen.

Der Kunde legt bei der Bestellung des Service fest, wie viele OCPUs er nutzen möchte – danach bemessen sich dann auch die Subskriptionskosten. Eine OCPU entspricht hier einem Prozessorkern, das ist doppelt so viel Rechenleistung, als in Cloud-Subskriptionsmodellen anderer Anbieter mit vCPU (entspricht einem

OCM SERVICE METRICS	OCM 288
Cores	288
Compute Nodes	8
Cores Required For Cloud Software Stack	72
Cores Required For VM Management (2 per compute)	12
Cores available for IaaS / PaaS workload	$288 - (72+12) = 204$
Total Memory	2 TB

Tabelle 2: Spezifikation der OCM 288 für „Cloud Control Plane“

Thread) bezeichnet wird. OCPUs können im Rahmen der in *Tabelle 1* dargestellten Grenzen jederzeit hinzugebucht werden.

Falls die gewählte Konfiguration nicht mehr ausreichend sein sollte, ist natürlich

ein Upgrade der Exadata Cloud Machine möglich, dieses erfolgt jedoch nur in festen Schritten, also von Eighth zu Quarter zu Half und zu Full Rack. Eine elastische Konfiguration ist zum heutigen Zeitpunkt

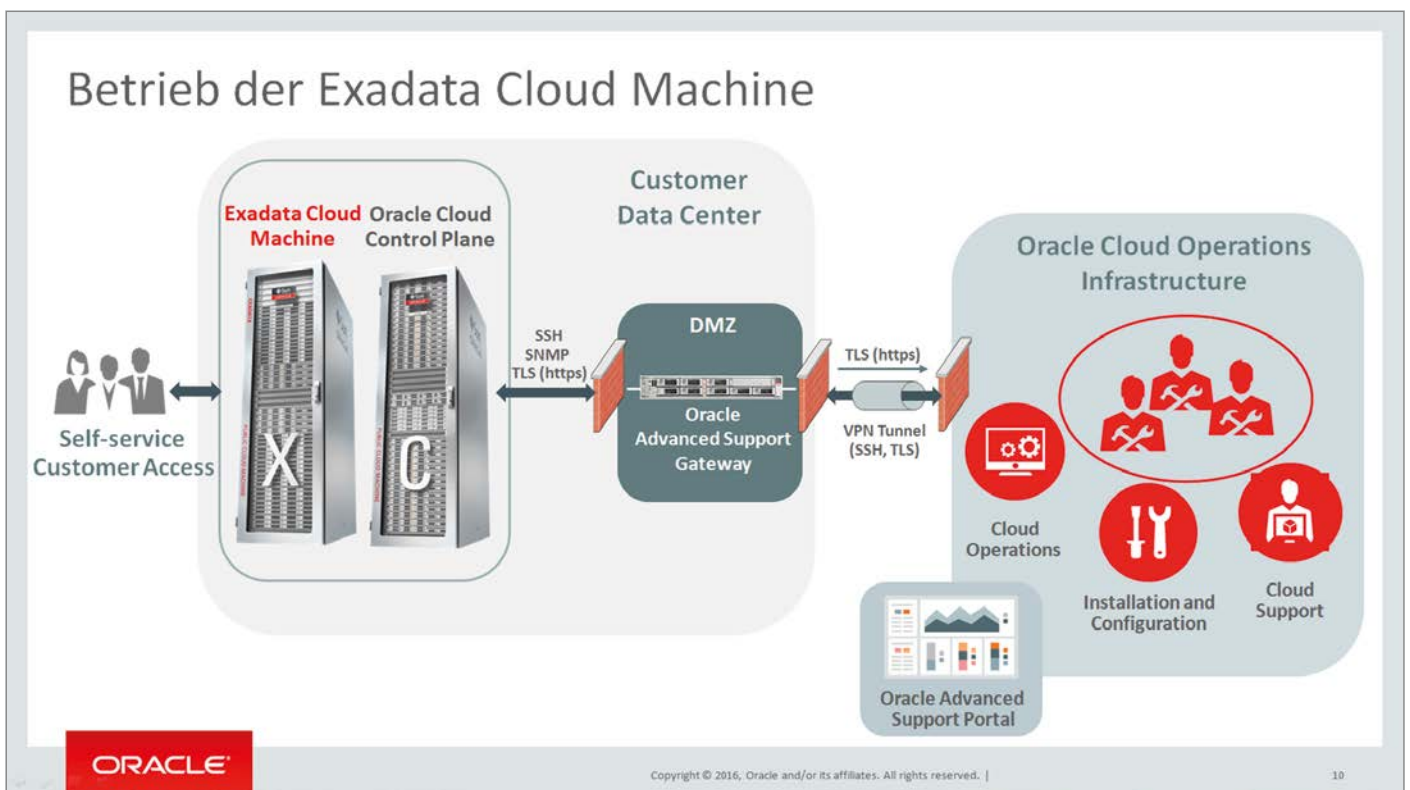


Abbildung 1: Das Betriebsmodell der Exadata Cloud Machine

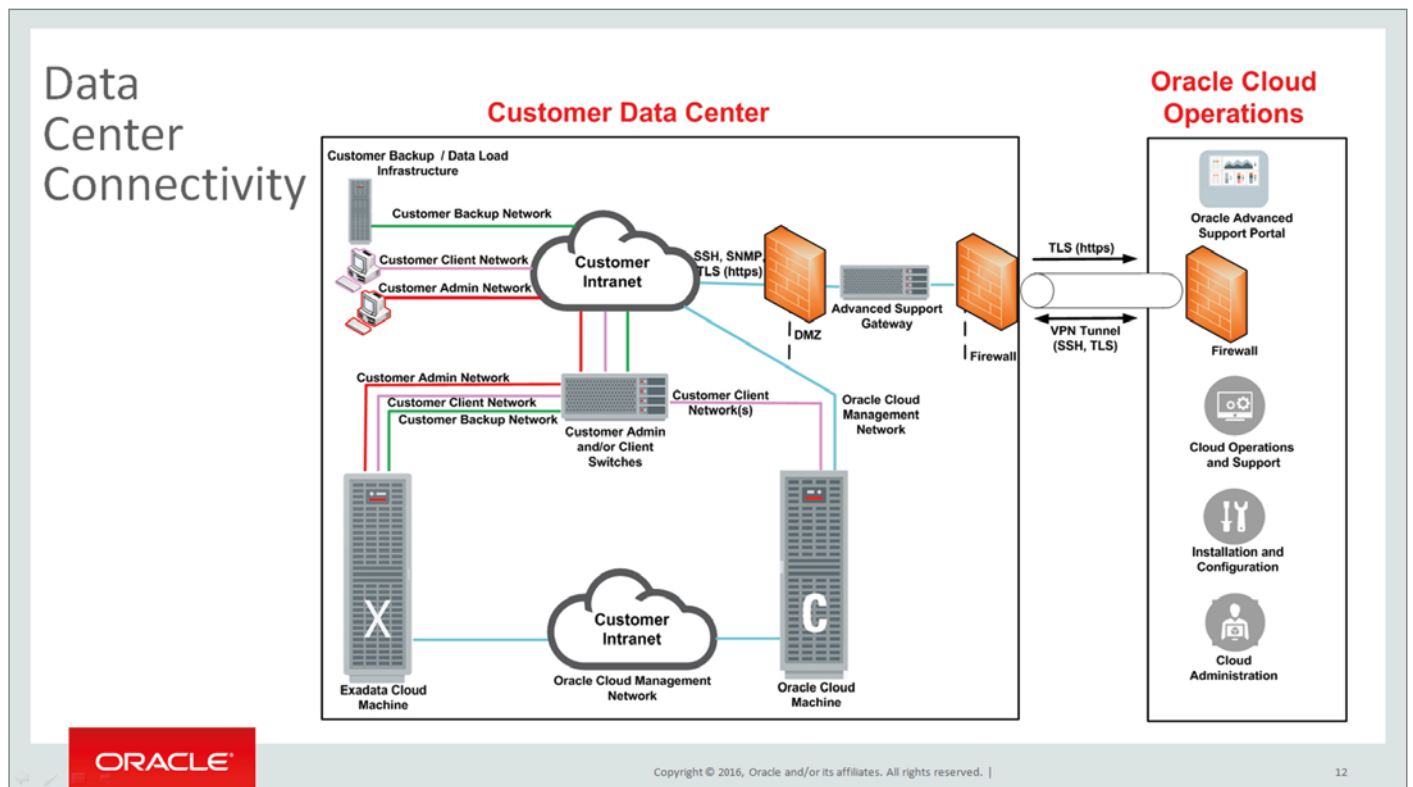


Abbildung 2: Die „Cloud@Customer“-Netzwerk-Topologie

noch nicht vorgesehen. Im Rahmen eines Upgrades werden die fehlenden Hardware-Komponenten nachgerüstet, für das Upgrade auf das Quarter Rack erfolgt also unter anderem auch eine Hauptspeicher-Erweiterung auf 768 GB.

Die Software

Mit der Exadata Cloud Machine wird die aktuelle Software-Generation ausgeliefert. Es werden unterschiedliche Datenbankversionen unterstützt:

- Oracle Database 11.2.0.4 oder 12.1.0.2 mit Grid Infrastructure 12.1.0.2
- Oracle Database 12.2.0.1 mit Grid Infrastructure 12.2.0.1

Auf der Exadata Cloud Machine stehen dem Kunden alle Datenbank-Optionen zur Verfügung, ebenso können alle Exadata-Schlüsselfunktionen und Datenbank-Management-Packs genutzt werden.

Die Maschine wird in der virtualisierten Konfiguration installiert; neben der für die Verwaltung der Maschine notwendigen „dom0“, die ausschließlich in der Zuständigkeit von Oracle liegt, wird eine „domU“ für die zu betreibenden Kunden-Daten-

banken aufgesetzt. Für diese „domU“, die maximal 720 GB pro Datenbank-Server groß werden kann, hat der Kunde vollen Administrator-Zugriff und kann Datenbanken, Agenten oder Skripte nach seinen Anforderungen aufsetzen. Zusätzliche „domU“ können zum jetzigen Zeitpunkt nicht eingerichtet werden.

Beim Einsatz größerer Konfigurationen (Half Rack oder Full Rack) ist es zum jetzigen Zeitpunkt nicht möglich, die Datenbank-Server eines Racks aufzuteilen und mit zwei unterschiedlichen Versionen der Grid Infrastructure zu betreiben – der Kunde muss sich also entscheiden, ob er auf seiner Exadata Cloud Machine 11.2 und 12.1 oder eben 12.2 betreiben möchte. Der Cluster wird grundsätzlich über alle verfügbaren Knoten aufgesetzt, also ein 2-Node-RAC im Quarter, ein 4-Node RAC im Half etc.

Es gibt eine klare Trennung der Zuständigkeiten in der Administration der Exadata Cloud Machine: Oracle verwaltet die Exadata-Infrastruktur, also Server, Storage, Storage-Software, Netzwerk, Firmware, Hypervisor etc., während der Kunde all die Komponenten kontrolliert und verwaltet, die direkt mit seinen Anwendungen zusammenhängen, also Database, Grid Infrastructure, VM Guest (DomU) und Betriebssystem. Der Kunde kann seine Daten-

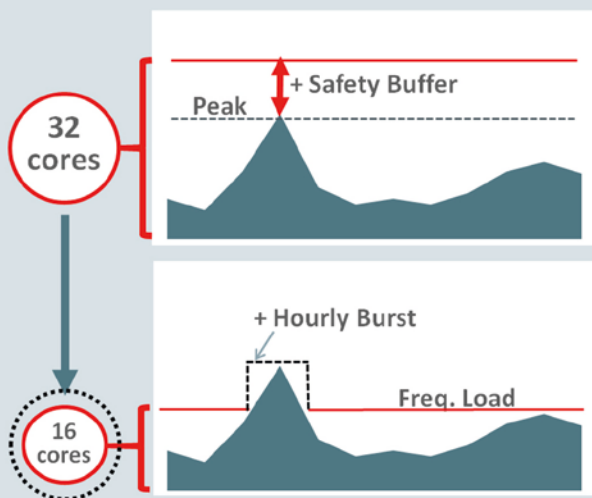
banken nach seinen eigenen Vorgaben einrichten und betreiben:

- Automatisierte Datenbank-Updates und -Patches nach Planung seiner eigenen Wartungsfenster
- Rollierende Ausführung zur Vermeidung von Ausfallzeiten
- Einbinden der Datenbanken in das bestehende (On-Premise)-Backup-Konzept
- Einhalten der Oracle MAA Best Practices nach Bedarf und Anforderungen

Installation und Betrieb

Die Installation der Exadata Cloud Machine wird nach Kundenvorgaben durch den Oracle Advanced Customer Service vorgenommen. Der zugrunde liegende Prozess entspricht im Wesentlichen dem Vorgehen bei allen Engineered Systems. Durch geeignete Tools und Termine vor Ort wird die Installation vorbereitet und nach erfolgter Auslieferung aller Komponenten erfolgt eine weitgehend automatisierte Implementierung der gesamten Lösung inklusive Übergabetests innerhalb weniger Tage. Erst nach erfolgter Abnahme beginnt die Berechnung der gebuchten Cloud Services.

Lastspitzen abfangen mit Online Compute Bursting



On Premise

- Auslegen auf Spitzenlast, zuzüglich Sicherheits-Puffer

ExaCM / ExaCS

- Auslegen auf Standard Last
- Bursting zum Abfangen von ungeplanten Spitzenlasten

ORACLE

Copyright © 2016, Oracle and/or its affiliates. All rights reserved. |

26

Abbildung 3: Online Compute Bursting

Für den Betrieb der Oracle-Cloud-Lösungen ist es essenziell, dass sich das nutzbare Toolset im „Look & Feel“ auf den verschiedenen Umgebungen nicht unterscheidet. Zu diesem Zweck ist ein sogenanntes „Cloud Control Plane“ auf einer separat zu erwerbenden Oracle Cloud Machine (OCM) eingerichtet. Dazu kommt das kleinste verfügbare Modell zum Einsatz, die Oracle Cloud Machine 288. Sie verfügt über acht Compute Server, die mit jeweils zwei Eighteen-Core-Intel-Xeon-E5-2699-v3-Prozessoren (2.3 GHz) sowie einer Hauptspeicher-Ausstattung von 256 GB DDR3 RAM und 800 GB interne SAS-3 SSD bestückt sind. Zusätzlich enthält die OCM ein 160 TB großes NAS-System mit 6,4 TB Read Cache und 800 GB Write Cache für den internen Gebrauch. Die OCM ist wie die Exadata Cloud Machine in das Kunden-Netzwerk integriert.

Der Software-Stack für die Administration mittels Cloud Control Plane läuft auf zwei Compute-Servern mit insgesamt 72 Kernen, der VM-Overhead auf den verbleibenden Compute-Servern verwendet jeweils zwei Kerne, sodass für weitere optionale IaaS- oder PaaS-Dienste auf der Cloud Machine dann noch 204 Kerne zur freien Verfügung stehen (siehe Tabelle 2).

Cloud Control Plane ist eine umfassende Software-Suite, die die gesamte Steu-

erung der Cloud-Umgebung beinhaltet. Angefangen mit dem Bestellprozess, der Verrechnung, dem Kapazitäts-Management und der Bereitstellung von Cloud-Services bis hin zur Verwaltung mehrerer Exadata Cloud Machines innerhalb eines Rechenzentrums-Netzwerks werden alle Funktionen, die der Kunde auch aus dem Public-Cloud-Angebot her kennt, zentral betrieben. Wie auch für den Exadata Cloud Service gibt es für die Administration der Exadata Cloud Machine darüber hinaus ein REST-API, mit dessen Hilfe Administrationsaufgaben skriptgesteuert ausgeführt werden können:

- I/O-Ressourcen verwalten
- Datenbank-Instanzen anlegen oder löschen
- SSH-Zugriff anlegen und anzeigen
- Compute Nodes starten, stoppen oder neu starten
- Informationen zu einzelnen Datenbanken, allen Datenbanken, Compute-Nodes oder dem Status einer Operation

In der online verfügbaren Dokumentation (siehe „<http://docs.oracle.com/cloud/latest/exadatacs/EXARS/index.html>“) ist das REST-API sehr ausführlich anhand von praktischen Beispielen beschrieben. Für den Betrieb der Infrastruktur der Cloud-Machines ist Oracle Cloud Operations zuständig. Der

Zugriff erfolgt über abgesicherte Verbindungen (siehe Abbildung 1).

Zu diesem Zweck ist in der DMZ des Kunden ein Advanced-Support-Gateway installiert, über das der Zugriff von Oracle-Cloud-Operations erfolgt. Man unterscheidet grundsätzlich zwischen zwei Arten von Netzwerk-Verkehr. Externer Netzwerk-Verkehr findet zwischen dem Advanced-Support-Gateway und den Cloud-Operations statt:

- Übermittlung von Konfigurations-, Telemetrie- oder Diagnosedaten über https (outbound)
- Remote Management über TLS VPN (inbound)

Interner Netzwerkverkehr findet zwischen dem Advanced-Support-Gateway und der Oracle Cloud Machine (OCM) sowie der Exadata Cloud Machine (ExaCM) statt:

- Kommunikation vom Agenten in der OCM zum Support Gateway über TLS
- Kommunikation zwischen OCM und Support Gateway über SSH
- Hardware Überwachung der OCM an Support Gateway über SNMP
- Zugriff vom EM-Agenten in der OCM zur ExaCM
- Hardware-Überwachung der ExaCM an das Support-Gateway über SNMP

Abbildung 2 zeigt die vollständige Darstellung der gesamten „Cloud@Customer“-Netzwerk-Topologie.

Durch die Einbindung der Exadata Cloud Machine in die lokale Enterprise-Manager-Cloud-Control-Administrationsoberfläche erhält der Kunde die Möglichkeit, seine gesamte Umgebung, von On-Premise bis in die Public Cloud, über eine Oberfläche zu überwachen und zu steuern. Enterprise Manager Cloud Control sorgt unter anderem dafür, dass Systeme leicht von On-Premise in die Cloud migriert werden können.

Hochverfügbarkeit sowie Backup und Recovery

Für unternehmenskritische Anwendungen und Datenbanken müssen natürlich ein hochverfügbarer Betrieb und eine den Anforderungen an Recovery Time Objective (RTO) und Recovery Point Objective (RPO) entsprechende Backup-und-Recovery-Strategie sichergestellt sein. Die Exadata Cloud Machine bringt (wie das On-Premise-Modell auch) bereits von sich aus wesentliche Funktionen mit; Active/Active-RAC-Konfiguration, ASM-Redundanz und redundant ausgelegte Netzwerk-Komponenten sind feste System-Bestandteile. Eine Absicherung für den Disaster-Fall erfolgt idealerweise über eine Exadata Cloud Machine als Standby, aber es können auch alle anderen Plattform-kompatiblen Lösungen genutzt werden.

Für das Backup-Konzept steht die gesamte Rechenzentrums-Infrastruktur zur Verfügung; natürlich ist es auch möglich, auf der Exadata Cloud Machine ein lokales Backup anzufertigen, was allerdings den für die Datenbanken nutzbaren Platz deutlich reduziert. Es lassen sich eine kundeneigene Lösung wie die Oracle Recovery Appliance, ein NFS-Server (etwa ZFS Backup Appliance) oder auch der Oracle Database Backup Cloud Service nutzen. Auch Lösungen von Dritt-Anbietern, die möglicherweise als Unternehmensstandard vorgeschrieben sind, können problemlos im Sicherungskonzept auf Tape, Disk oder in der Cloud Verwendung finden.

Vorteile und Unterschiede

Die Exadata Cloud Machine bietet im Vergleich zur Exadata Database Machine On-Premise große Vorteile, allerdings sind auch

einige wichtige Unterschiede zu beachten. Die möglichen Konfigurationen der beiden Lösungen unterscheiden sich erheblich, so ist eine auf extreme Rechenleistung ausgelegte, elastische Konfiguration zurzeit in der Cloud nicht abbildbar. In der Cloud stehen lediglich feste Konfigurationen zur Verfügung, die für vier Jahre gemietet werden. Ein Upgrade in der Cloud hat wiederum eine Laufzeit von vier Jahren.

Der wesentliche Unterschied zur On-Premise-Exadata ist ein wirklich bedarfsgerechtes Sizing mit „Capacity on Demand“-Fähigkeiten. Auf der On-Premise-Exadata ist ein Capacity on Demand zwar möglich, jedoch funktioniert das nur in eine Richtung, das heißt, einmal erweiterte Ressourcen und die dafür aufgewendeten Kosten bleiben bestehen. Ein Schrumpfen ist nicht vorgesehen.

Mit der Exadata Cloud Machine oder dem Exadata Cloud Service steht das sogenannte „Online Compute Bursting“ zur Verfügung, das es möglich macht, die Rechenleistung (Anzahl der OCPU) der Exadata auf eine durchschnittliche Standardlast auszuliegen und zu Spitzenlastzeiten online maximal zu verdoppeln (siehe Abbildung 3). Die hinzugebuchten OCPUs werden dann auf Stundenbasis abgerechnet. Sobald keine zusätzliche Rechenleistung mehr erforderlich ist, kann das Bursting wieder abgestellt werden, es fallen keine weiteren Kosten mehr an. Für den Fall, dass ständig mehr Leistung benötigt wird, kann natürlich die standardmäßig genutzte Anzahl von OCPUs jederzeit hochgesetzt werden, diese muss man dann jedoch beibehalten.

Es gelten bei der Konfiguration der Hardware der Exadata Cloud Machine einige Einschränkungen, denn diese geht ja nicht in das Eigentum des Kunden über, sondern wird lediglich durch ihn genutzt. So ist eine Verbindung zwischen On-Premise genutzten Exadata-Systemen zur Exadata Cloud Machine über InfiniBand nicht möglich. Auch das von einigen Kunden durchgeführte Re-Racking, also die Verwendung spezieller eigener Racks, ist nicht erlaubt, ebenso wie das Zusammenfassen der Komponenten, die zur „Cloud@Customer“-Umgebung gehören, in einem Rack.

Softwareseitig werden auf der Exadata Cloud Machine beim Aufsetzen eines Datenbank-Service grundsätzlich alle vorhandenen Knoten genutzt, für einen Single-Instance-Betrieb ist manuelle Nacharbeit notwendig. Eine Einrichtung zusätzlicher virtueller Umgebungen für den Kunden ist

derzeit ebenfalls nicht möglich – daher sind einer gewünschten Konsolidierung auf der Exadata Cloud Machine möglicherweise Grenzen gesetzt.

Fazit

Mit der Exadata Database Machine, dem Exadata Cloud Service und der Exadata Cloud Machine hat der Kunde die Wahlmöglichkeit, wo und wie er seine Performance-kritischen Oracle-Datenbanken betreiben möchte: On-Premise, in der Cloud oder mithilfe von Oracle-Cloud-Operations im eigenen Rechenzentrum als Cloud@Customer. Bei dieser Entscheidung spielen nicht nur Investitionskosten vs. Betriebskosten eine Rolle, sondern ebenfalls regulatorische Vorschriften oder technische Rahmenbedingungen, die unter Umständen einen Gang in die Public Cloud verhindern.

Agile und flexible Cloud-Technologie mit den entsprechenden Werkzeugen sorgen dafür, dass schnell auf geänderte Anforderungen reagiert werden kann und moderne Anwendungen inklusive der Datenbanken in kürzester Zeit dem Anwender bereitstehen. Mithilfe der für den Betrieb der Exadata Cloud Machine zusätzlich notwendigen Oracle Cloud Machine lassen sich nach Bedarf weitere IaaS- und PaaS-Services hinzubuchen, die das Anwendungsspektrum der Cloud@Customer deutlich erweitern.

Weiterführende Informationen

- <https://www.oracle.com/cloud/cloud-at-customer.html>
- <https://www.oracle.com/engineered-systems/exadata/index.html>
- <http://www.oracle.com/technetwork/database/exadata/exacm-ds-3409774.pdf>



Frank Schneede
frank.schneede@oracle.com

Akzeptanz von „My Oracle Support“ steigt, allgemeine Zufriedenheit sinkt

Christian Trieb, DOAG-Vorstand und Leiter Cometece Center Support

Im Rahmen der DOAG 2016 Konferenz hat Christian Trieb, Leiter des DOAG Competence Centers Support, die Ergebnisse der diesjährigen Support-Umfrage präsentiert. Das Serviceportal „My Oracle Support“ schnitt dabei deutlich besser ab als im Jahr 2014: 48 Prozent der Teilnehmer vergaben die Note „gut“. Damit verbesserte sich der „My Oracle Support“ um 7 Prozentpunkte im Vergleich zu 2014. Die allgemeine Zufriedenheit ist dagegen gesunken.

Insgesamt vergaben in diesem Jahr 19 Prozent der 214 Oracle-Support-Kunden aller Unternehmensgrößen und Berufsbilder dem Oracle-Support die Note „Gut“ (2014: 35 Prozent), rund ein Viertel der Befragten bewertete ihn wie schon in der letzten Umfrage als „Befriedigend“ und 56 Prozent bezeichneten ihn als „Nicht ausreichend“ (2014: 38 Prozent). Im Detail ist mehr als die Hälfte der Befragten mit der Qualität

(59 Prozent) und den Prozessen (60 Prozent) weniger zufrieden, die Reaktionszeit des Oracle-Supports bewertete knapp die Hälfte als „Nicht ausreichend“. 52 Prozent ziehen den Support durch einen Drittanbieter in Erwägung (2014: 57 Prozent).

Weiterhin viel Luft nach oben bietet der Bekanntheitsgrad einzelner Support-Tools, wie etwa der „Oracle Proactive Support“. Das Tool ist bei 40 Prozent der

Befragten unbekannt (2014: 65 Prozent) und erreicht damit zwar einen höheren Bekanntheitsgrad als zwei Jahre zuvor, aber 6 Prozent nutzen es (2014: 5 Prozent) und 1 Prozent denkt über dessen Einführung nach (2014: 3 Prozent).

Der angekündigte Oracle-Vertreter Mario Kowalski, Customer Support Country Leader bei Oracle Deutschland, konnte zur Präsentation der Ergebnisse aus

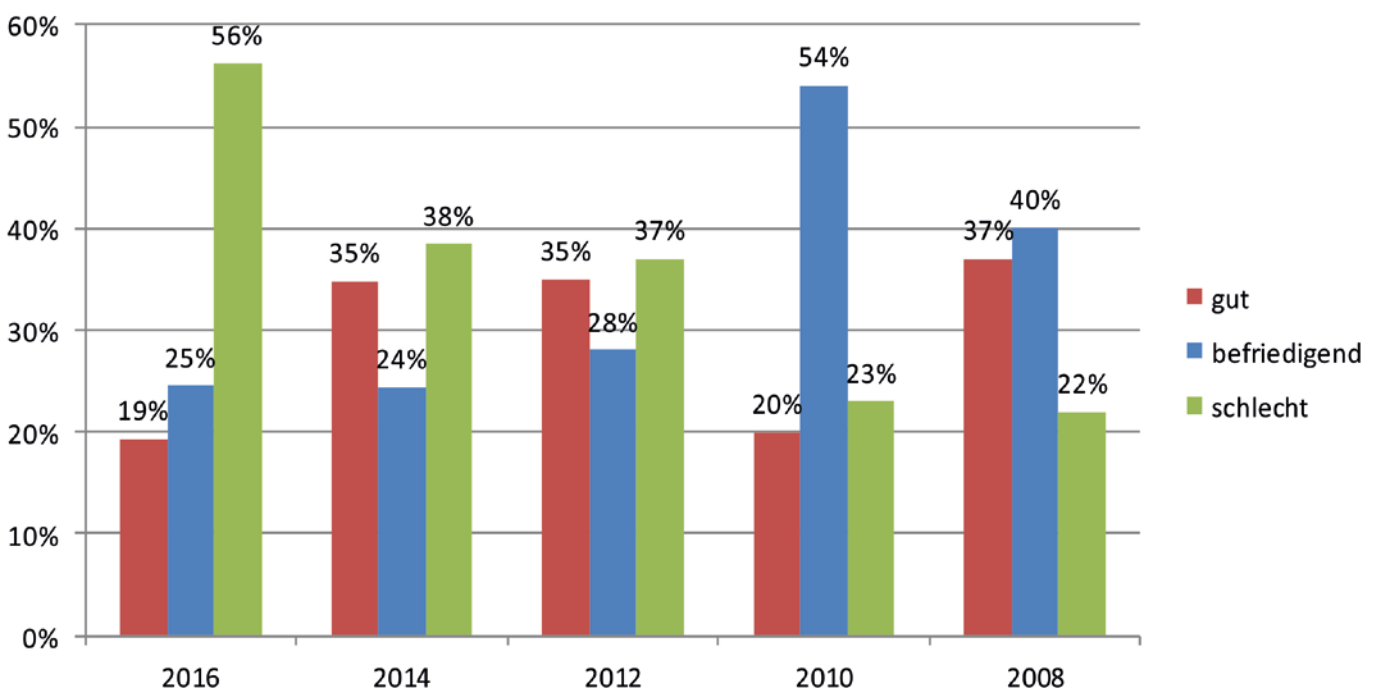


Abbildung 1: Allgemeine Zufriedenheit mit dem Oracle-Support

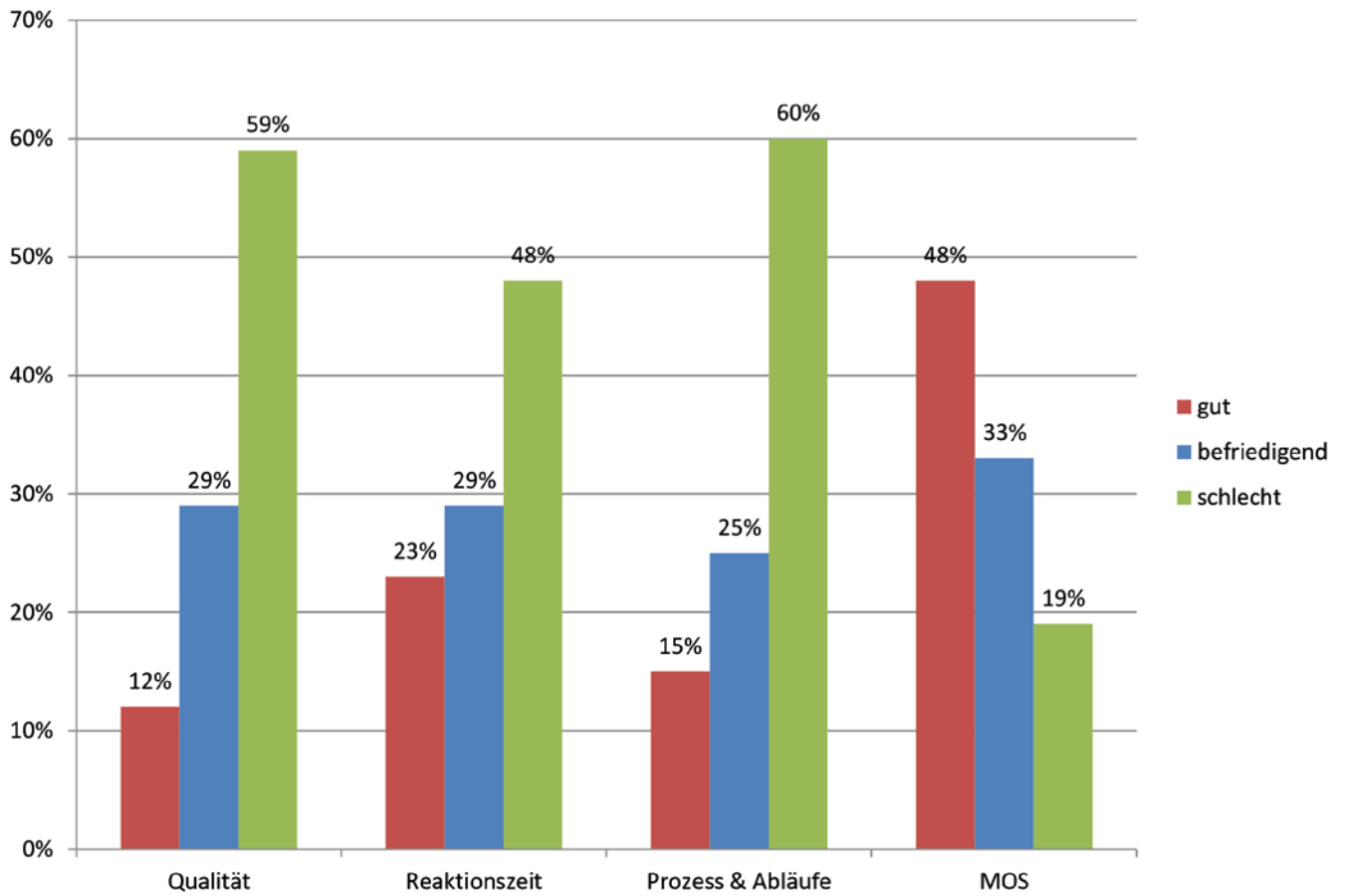


Abbildung 2: Einzelergebnisse Oracle-Support (2016)

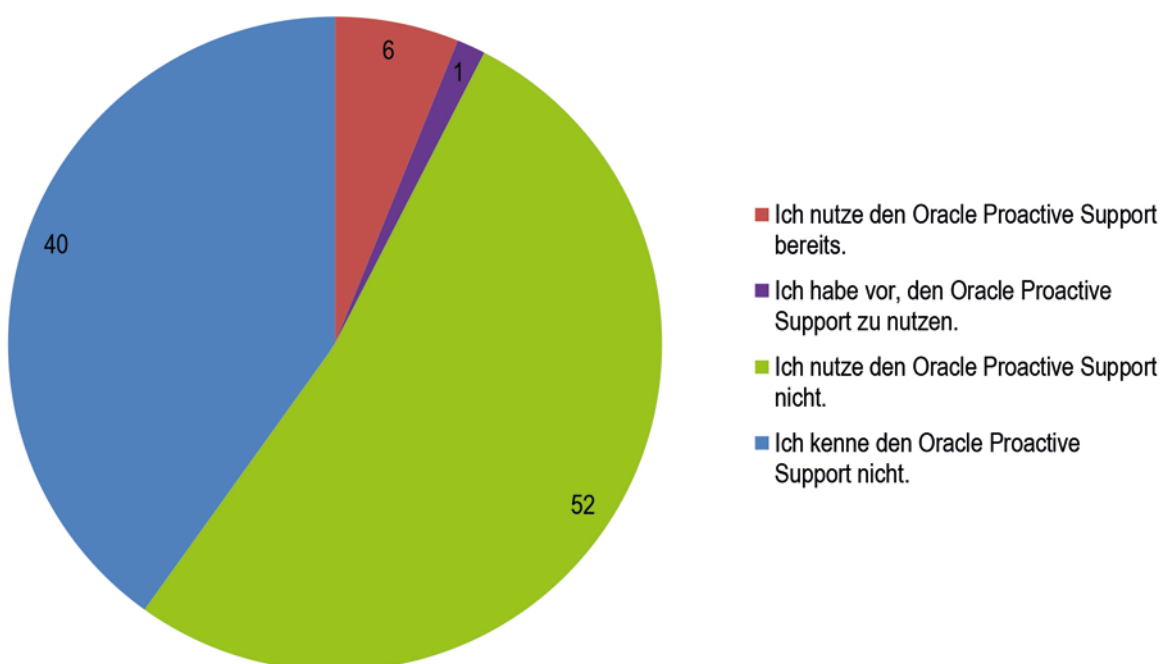


Abbildung 3: Nutzung der von Oracle bereitgestellten proaktiven Tools

Krankheitsgründen kurzfristig nicht teilnehmen, hat aber in einem Artikel Stellung dazu bezogen (siehe Seite unten).

Die DOAG wird auch zukünftig im Dialog mit Oracle versuchen, Verbesserungen beim Oracle-Support zu erreichen. Insbesondere

steht das DOAG Competence Center Support allen DOAG-Mitgliedern zur Unterstützung bei Supportfragen zur Verfügung.

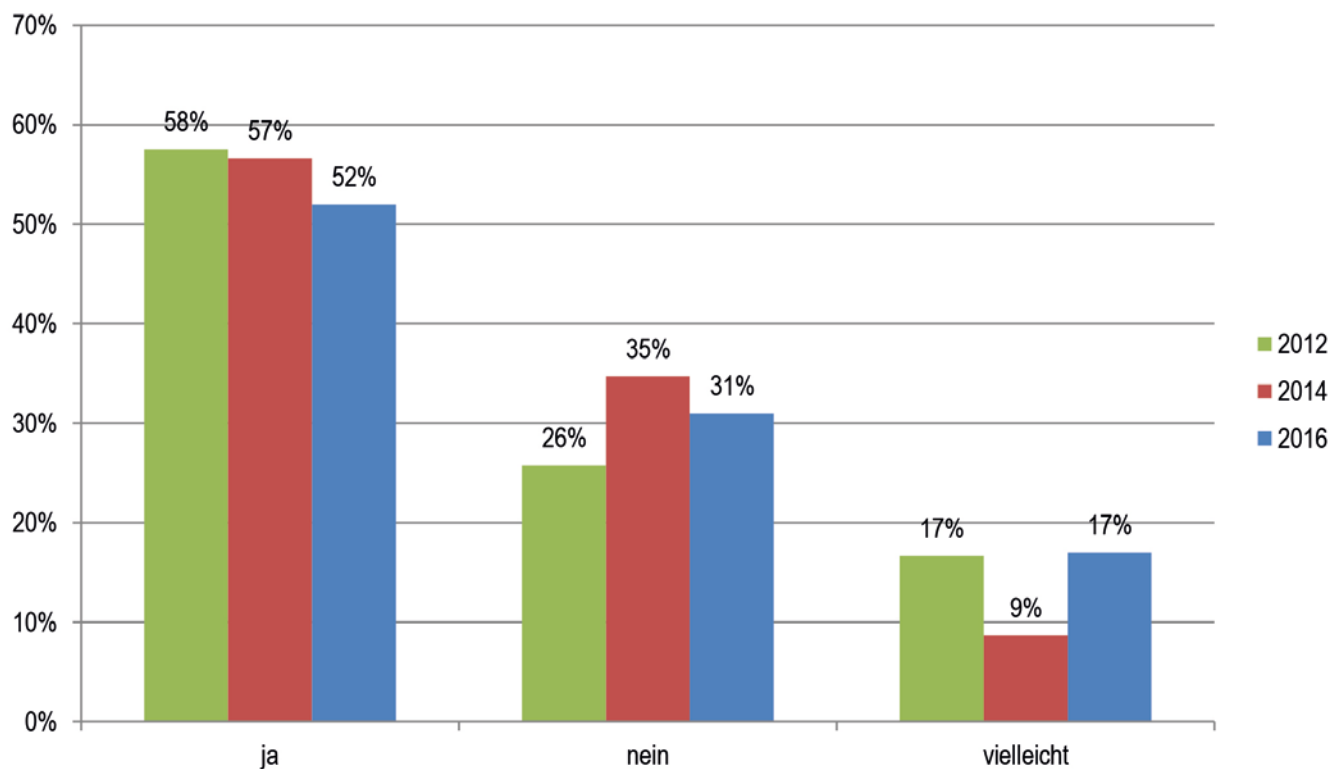


Abbildung 4: Interesse an Support durch Drittanbieter

Die DOAG Support-Umfrage

Mario Kowalski, Customer Support Country Leader Oracle Deutschland

Auf der jüngsten DOAG Konferenz in Nürnberg wurden im Detail die Ergebnisse der diesjährigen Support-Umfrage vorgestellt. An dieser Umfrage haben 214 Personen teilgenommen und ihr Urteil zu den verschiedenen Fragestellungen gegeben. Angeschrieben waren 9.000 Personen aus den 7.500 Mitgliedsorganisationen. Auch wenn Statistiken, zugleich zu einem hoch emotionalen Thema, sicher nicht die vollständige Wahrheit aufzeigen und auch wenn die Anzahl der Rückläufer nur 2,3 Prozent der Befragten ausmacht, nehmen wir jedes einzelne Feedback ernst.

Ich möchte an dieser Stelle, bevor ich auf einige der ganz offensichtlichen und auch von uns erkannten Problembereiche eingehe, einige vergleichbare Ergebnisse aufzeigen, die wir ebenfalls direkt von unseren Kunden erhalten und die über ei-

nen Zeitraum von zwölf Monaten gemessen werden (siehe Abbildung 1).

Während bei der DOAG-Umfrage nur 19 Prozent der 214 Rückläufer sagen, dass sie im Grunde mit dem Service von Oracle zufrieden sind, sind es im weltwei-

ten Durchschnitt und auch in Deutschland 75 Prozent. Auch das My Oracle Support Portal (MOS) erhält bei der DOAG-Umfrage deutlich schlechtere Werte (48 Prozent sind zufrieden) als in der hier aufgezeigten Umfrage (77 Prozent). Die proaktiven Sup-

Like-to-Like Betrachtung German Customers

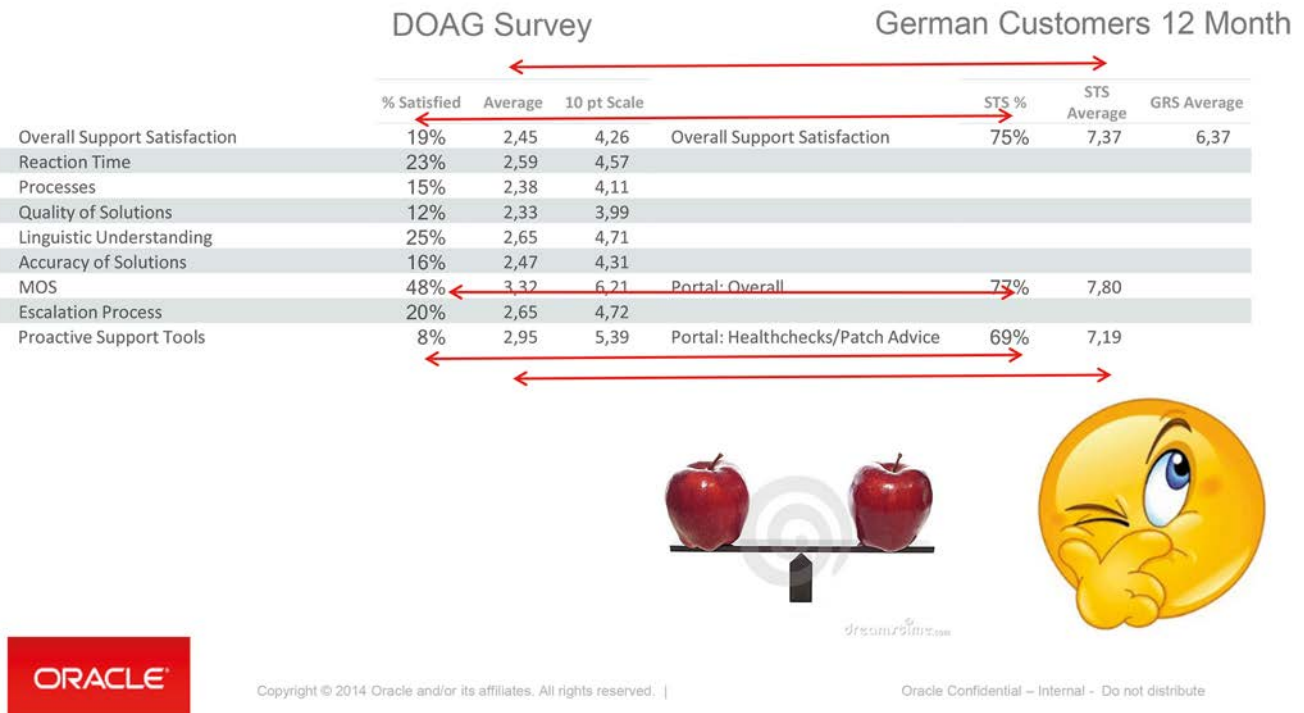


Abbildung 1: Die Ergebnisse von Oracle

port-Tools schneiden bei der DOAG-Umfrage nur mit 8 Prozent als zufriedenstellend ab im Vergleich zu 69 Prozent in der von mir hier vorgestellten Umfrage.

„Traue nie einer Statistik, die Du nicht selber erstellt hast“

Ich möchte nun zunächst versuchen zu ergründen, wie es zu diesen doch massiven Unterschieden kommen kann.

Wer wird befragt? In einer Umfrage, die sich an 9.000 Adressaten richtet, von denen 214 ihre Antworten zurückschicken, muss man zunächst positiv hervorheben, dass immerhin gut 40 Teilnehmer ohne Not sagen, dass sie den Service von Oracle gut finden. 52 Personen konnten sich nicht so recht entscheiden und knapp 120 Personen bringen ihre Unzufriedenheit zum Ausdruck.

Die entspricht auch in etwa dem statistischen Mittel, was vergleichbare Umfragen „Wie zufrieden sind sie mit ...“, unabhängig vom Thema, ergeben. In dem Augenblick, in dem ich die Aufforderung zur Teilnahme erhalte, unterhält sich mein Kopf mit meinem Bauch. Und wenn der Bauch sagt,

„Mensch, da hatten wir doch gerade dieses unsägliche Erlebnis“, dann entscheidet der Kopf, nimmt teil und gibt das Urteil des Bauchs ab. Wenn in dem Augenblick der Umfrage der Bauch signalisiert: „Eigentlich alles ok“, denkt der Kopf:

- a) Hab' ich da überhaupt Lust zu reagieren?
- b) Ok, ich werde reagieren (s.o. „ohne Not“)

Denken Sie auch an das Phänomen bei Wahlen, bei denen die zufriedene Mehr-

heit am schönen sonnigen Sonntag nicht wählen geht und kleine Parteien, die ihre Wählerschaft besser mobilisieren, überdurchschnittlich gut abschneiden.

Unsere Umfragen werden unmittelbar nach Abschluss eines Service-Request an die Kunden gesendet, und zwar unabhängig davon, ob der schnell und gut bearbeitet wurde oder monatelang in der Bearbeitung brauchte. Und dies geschieht jeden Tag, jede Woche, jeden Monat, jedes Jahr. Das hier gezeigte Ergebnis beruht immer auf mehreren Tausend Rückläufern über einen Zwölf-Monats-

	Q4FY14	Q4FY16
Overall Response time met	91 Prozent	92,6 Prozent
Prozent Escalated	5 Prozent	3,4 Prozent
Overall Support Satisfaction	7,17	7,20
Overall SR Satisfaction	7,52	7,48
Timely Resolution	8,10	8,32
Communication Effectiveness	7,93	7,82
Time to Reach Expertise	7,58	7,50

Tabelle 1: Die Werte ab Zeile 3 sind auf einer 10-Punkte-Skala

Zeitraum. Da diese Werte seit Jahren gemessen werden, ist auch ein Blick auf die Entwicklung innerhalb der letzten Jahre interessant. *Table 1* zeigt einige Tendenzen im Vergleichszeitraum Q4FY14 und Q4FY16 (Mai 2014 bis Mai 2016).

Aber auch in dieser Umfrage äußert ein Viertel der Befragten ihre Unzufriedenheit mit dem Oracle Support und das kann mich in meiner Funktion als Country Support Leader nicht zufrieden stimmen.

Service ist kein Sympathie-Träger

Das wird der Service auch nie werden können. Denn ihn benötige ich in den meisten Fällen in einer für mich sehr schwierigen Situation: Meine Datenbanken legen gerade eine Produktion lahm, ganze Abteilungen sind zum Nichtstun gezwungen, der Ausfall der Exadata verhindert einen dringenden Quartalsabschluss. Und dann melde ich über MOS mein Problem und auch nach zwanzig Minuten ist noch nichts geschehen? Wissen Sie, wie lange zwanzig Minuten in dieser Situation sein können? Wenn dann die erste Kontaktaufnahme stattfindet, muss ich auch noch gefühlte einhundert Zusatzfragen beantworten und dann passiert immer noch nichts.

Problem Nr. 1: Communication Effectiveness

Doch es passiert etwas, aber keiner sagt es Ihnen oder hinterlässt einen entsprechenden Vermerk im SR „Ihr SR wird jetzt von mir bearbeitet, ich nehme gerade folgende Analysen vor und werde mich in spätestens zwei Stunden mit einem Vorschlag bei Ihnen melden“. Das würde Ihnen aufzeigen, dass an Ihrem Problem gearbeitet wird. Stattdessen Schweigen.

Problem Nr. 2: Time to Reach Expertise

Wenn es dann noch Ewigkeiten dauert, bis der richtige Experte sich mit Ihrem Problem beschäftigt, dann schweben Sie zu Recht bereits unter der Decke.

Aber was könnte tatsächlich geschehen sein? Ein Mitarbeiter des Product Supports kommt zu dem Ergebnis, dass

er selber den Fehler nicht erkennen kann und auch kein Eintrag aller bekannten Fehler in seiner globalen Wissensdatenbank Ihr Fehlerbild beschreibt. Vielleicht haben Sie selbst auch bei Ihrer SR-Eröffnung bereits die Vermutung geäußert, dass es sich um einen Fehler im Produkt handelt (defect). In diesen Fällen wendet sich der Support zumeist – wenn nicht ganz offensichtlich die Lösung erkennbar ist – an die Product-Development-Kollegen (dies ist der „Bug“: eine Rückfrage ans Development). Dies ist eigentlich auch vernünftig, denn dort ist das Produkt entstanden. Die noch erfahreneren Experten sind also dort. Die Komplexität wird damit natürlich größer und die Bearbeitungszeit verlängert sich nun.

Hatte ich vergessen zu erwähnen, dass nun auch wieder Problem Nr. 1 zum Einsatz kommen könnte? Es wurde nämlich manchmal vergessen, Ihnen das auch mitzuteilen.

Und nun erzähle ich diese wahre Geschichte noch zu Ende.

Am Ende stellt sich heraus, es war gar kein Bug und dieser Request wurde mit „Not a Bug“ geschlossen. Tatsächlich ergab sich während der weiteren Analyse und im anschließenden Lösungsvorschlag „solved by config/set up steps“. Hier sind ganz offensichtlich mehrere Dinge, nicht zuletzt auch bei Oracle, schiefgelaufen. Aber sie passieren.

Die drei „P“ – Products-People-Processes

Schon seit den 1980er-Jahren erhebt die IDC Untersuchungen über die Ausfallgründe in Rechenzentren. Das Ergebnis ändert sich in all den Jahren nur minimal. Zu etwa gleichen Teilen liegen die Ursachen in den Bereichen „Products“, „People“ und „Processes“. Ich kann das bei der Analyse unserer Service Requests bestätigen. Wir alle müssen also daran arbeiten, mögliche Fehlerquellen abzustellen. Es klingt banal, aber vor einem Jahr hatten wir eine wochenlange Eskalation bis ins Engineering und am Ende stellte sich heraus, dass der Kunde nach einem Umzug einige Kabel ausgetauscht hatte und diese für den Ausfall eines Exa-Clusters verantwortlich waren. Aber glauben Sie mir, es gibt auch ausreichend Beispiele, bei denen Oracle sich an die Nase fassen muss. In einem anderen Fall wurde bei der Bearbei-

tung eines Interim-Patches festgestellt, dass die Ursache doch in einem bereits analysierten Bereich des Programm-Codes lag.

Immer wieder gibt es Situationen, bei denen unterschiedliche Release- und Patch-Stände Ursache des Problems sind. Natürlich sind es zu gleichen Teilen auch Produkt-Probleme. Auch Konfigurations-Fehler sind oft Auslöser einer massiven Störung. Aber wer weiß das schon zu Beginn einer Fehler-Analyse so ganz genau?

Service ist also mitnichten eine Einbahnstraße, wir müssen uns immer aus mehreren Blickwinkeln dem Problem nähern – und es ist und bleibt eine gemeinsame Aufgabe. Seien Sie versichert, dass uns daran gelegen ist, die Probleme zu Ihrer Zufriedenheit zu lösen. Im Falle einer Störung ziehen wir gemeinsam an einem Strang. In einer partnerschaftlichen Arbeitsweise schaffen wir es am besten, die Probleme zu beheben. Was leiten wir also nun aus der DOAG-Umfrage und auch unseren eigenen Erkenntnissen ab?

1. Wir nehmen jede Bemerkung ernst und wollen und werden aus ihr lernen.
2. Wir bearbeiten unsere erkannten Baustellen wie Kommunikation und zeitnahes Erreichen der erforderlichen Experten.
3. Wir werden auf den DOAG-Regionalveranstaltungen mit Service-Vertretern Themen adressieren wie Umgang und Eskalationen in MOS, Proaktive Services und Platinum Support (hoffentlich mit mehr Teilnehmern als in der Vergangenheit).
4. Wir werden auch einige zentrale Webcasts anbieten, um uns mit Ihnen zu diesen Themen auszutauschen.

Wir müssen besser werden, jeden Tag, bei jedem Request. Daran arbeiten wir hart und ich wünsche mir: gemeinsam mit Ihnen.



Mario Kowalski
mario.kowalski@oracle.com



In-Memory-Datenbanken: Perfekte Add-on-Technologie für schnellere Analysen

Mathias Golombek, EXASOL AG

Dieser Artikel zeigt Einsatzzwecke und Vorteile der Daten-Analyse mithilfe von In-Memory-Technologie für datengetriebene Unternehmen auf. Darüber hinaus sind beispielhaft die In-Memory-Lösungen von Oracle und EXASOL sowie ihre Vor- und Nachteile für individuelle Unternehmensansprüche gegenübergestellt. Zusammen mit einem Anwenderbeispiel bietet der Artikel besonders Unternehmen eine gute Übersicht darüber, ob und in welcher Art sich der Einsatz einer analytischen In-Memory-Datenbank für sie lohnt.

Die Themen „Big Data“ und „Daten-Analyse“ sind deutlich mehr als der vielzitierte Hype, Big Data ist Realität: Innerhalb von nur 18 bis 24 Monaten verdoppeln sich produzierte Terabyte-Datenmengen in Unternehmen im Durchschnitt. Um von den Vorteilen der Daten-Analyse zu profitieren, sind häufig Unternehmens-, Produkt- oder Kunden-

daten für bestimmte Anwendungen im Subsekunden-Bereich zu analysieren. Für solche komplexen und schnellen Analysen sind In-Memory-Datenbanken eine ideale Lösung. Dieser Artikel geht auf die Besonderheiten analytischer In-Memory-Datenbanken als ergänzenden Performance-Layer für Oracle-Infrastrukturen ein.

Analytik und die Vorteile von In-Memory

Der Einsatz analytischer Datenbanken nimmt zu, denn in den Unternehmen wächst die Nachfrage nach Analytik. Der Bedarf ist insbesondere im Marketing hoch, wenn es um die Steigerung der

Kunden-Profitabilität durch Kunden-Identifizierung in Echtzeit und intelligente Kunden-Interaktionen geht. In der Supply Chain stehen die Prozess-Optimierung durch bessere Planung sowie die Identifizierung und Vermeidung von Risiken im Vordergrund. Auf der Ebene der Unternehmenssteuerung geht es um rechtzeitiges Erkennen von Markt-Trends und Innovationspotenzialen. So durchdringt Analytik alle Unternehmensbereiche.

Analytische Datenbanken bieten zugleich große Vorteile: Sie verbessern die Skalierbarkeit und die Performance von analytischen Datenbank-Abfragen gegenüber traditionellen Datenbanken deutlich. Zusätzlich helfen sie auch, die Betriebskosten zu senken. Das beruht auf der Kombination von bekannten und neuen Technologien wie Spaltenorientierung, Komprimierung, speziellen intelligenten Zugriffsverfahren, massiv paralleler Verarbeitung sowie In-Memory-Technologien. Speziell bei relationalen SQL-Datenbanken mit einer spaltenbasierten In-Memory-Datenverarbeitung lassen sich sehr viele Daten in kurzer Zeit komprimiert hundertmal schneller wegschreiben als in herkömmlichen zeilenorientierten Datenbanken. Der Zugriff auf im Hauptspeicher liegende Daten ist um bis zu tausendmal schneller als der Zugriff auf Daten, die sich auf der Festplatte befinden. In den meisten Fällen sind analytische In-Memory-Datenbanken insgesamt dadurch um den Faktor fünfzig bis hundert schneller und ermöglichen fundierte Echtzeit-Reaktionen basierend auf großen Datenmengen, die zuvor undenkbar waren. So lassen sich Ad-hoc-Auswertungen in Sekundenschnelle auf Knopfdruck bereitstellen.

Datengetriebenen Unternehmen bieten In-Memory-Datenbanken zudem das benötigte Maß an Flexibilität: Die Lösungen sind leicht zu implementieren und skalierbar, was zu einer deutlichen Entlastung der eigenen IT-Ressourcen führt. Die Daten-Analyse vereinfacht sich außerdem erheblich, da strukturierte und unstrukturierte Daten aus verschiedenen Vorsystemen in einer In-Memory-Datenbank direkt ausgewertet werden können – ein vorheriges Aggregieren von Basisdaten ist nicht mehr zwingend erforderlich. Durch die zahlreichen Vorteile von In-Memory eröffnen sich Unternehmen so völlig neue Dimensionen der Daten-Analyse (siehe Abbildung 1).

Zusätzlich zu den klaren Performance-Verbesserungen helfen In-Memory-Daten-

banken auch, Investitionskosten zu minimieren. Big-Data-Analysen erfordern in der Regel keinen kompletten Umbau vorhandener Systeme. Oftmals geht es nur um gezielte Ergänzungen und Erweiterungen. Ausschlaggebende Kriterien für die Kosten der einzusetzenden Analyse-Systeme sind insbesondere die Menge der anfallenden Daten sowie die genutzte Datenbank-Technologie. Setzt ein Unternehmen auf eine analytische In-Memory-Datenbank, fallen die Lizenzkosten häufig nicht auf die gesamten Datenmengen an, sondern reduzieren sich auf den tatsächlich genutzten Arbeitsspeicher oder, wie bei der Oracle-In-Memory-Option, auf die Anzahl der CPUs. Hinzu kommen intelligente Kompressions-Algorithmen und vollautomatisierte Prozesse, die die Investitionskosten einer In-Memory-Datenbank im Vergleich zu konventionellen Datenbanken mehr als halbieren können. Auch Firmen mit beschränktem Budget sind damit in der Lage, durch flexibel skalierbare Software-as-a-Service- oder komplette Cloud-Lösungen Big-Data-Projekte erfolgreich und effizient umzusetzen. Hier empfiehlt sich ein Test-

lauf bei einem zeitlich begrenzten Projekt. Sind die Potenziale erkannt, können Unternehmen die temporären Lösungen Stück für Stück ausbauen und auf andere Bereiche und Projekte ausdehnen.

In-Memory-Lösungen im Vergleich

Seit dem Jahr 2013 bietet Oracle mit seiner Database 12c Enterprise Edition eine eigene In-Memory-Option an. Vor diesem Zeitpunkt mussten sich Nutzer von Oracle extern nach einer In-Memory-Datenbank-Lösung umsehen: Der Katalog-Versender Quelle AG, Anfang der 2000er-Jahre eines der größten Data-Mining-Unternehmen in Europa, verwaltete seine enormen Datenmengen deshalb nicht nur mithilfe eines der größten Oracle-RAC-Systeme seiner Zeit, sondern nutzte ab dem Jahr 2004 zusätzlich die In-Memory-Lösung der EXASOL AG.

Mit dem Release der 12c Enterprise Edition müssen sich Nutzer nicht mehr zwingend nach einer externen In-Memory-Datenbank umsehen. Dennoch kann eine

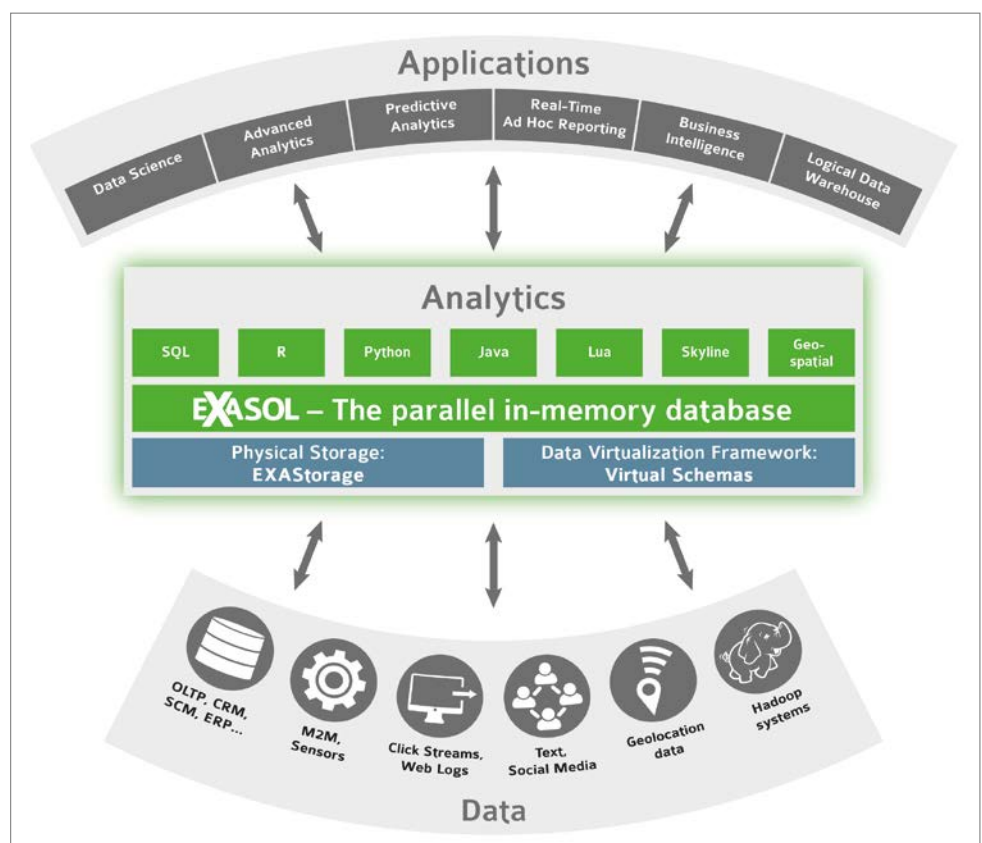


Abbildung 1: Mit einer In-Memory-Datenbank lassen sich strukturierte sowie unstrukturierte Daten aus verschiedenen Vorsystemen ohne vorherige Aggregation direkt analysieren

durchdachte Wahl klaren Mehrwert bieten: Sowohl die speziell für ein Oracle-Umfeld entwickelte Datenbank selbst als auch die auf ein solches Umfeld angepasste Datenbank von EXASOL bieten beispielsweise spezifische Eigenheiten, die unterschiedliche Vor- und Nachteile mit sich bringen. Die Entscheidung, welche Datenbank den eigenen Ansprüchen am besten gerecht wird, sollte daher individuell und gut durchdacht getroffen werden, wie nachfolgend in einem Vergleich zwischen den beiden Lösungen dargestellt ist.

Ein signifikanter Unterschied liegt bereits in der Speicherung der Daten. Bei Oracle 12c werden die Daten erst auf die Festplatte geschrieben und anschließend in den Hauptspeicher geladen. Dieses Vorgehen spart Platz im Hauptspeicher – das ist notwendig, da es zu Query-Abbrüchen kommen kann, falls der Hauptspeicher für die jeweilige Abfrage nicht ausreicht. Jedoch beeinflusst es die Performance der Datenbank: Der Zugriff auf das vergleichsweise langsame Transaktionslog sowie die Praxis, Daten erst auf die Festplatte zu schreiben und anschließend in den Hauptspeicher zu laden, verursachen vermeidbare Wartezeiten. Zusätzlich unterscheiden sich die Daten-Strukturen im Hauptspeicher von denen auf der Festplatte und müssen zwangsläufig konvertiert werden – ein weiterer Faktor, der Performance-Einbußen bei der Oracle-In-Memory-Lösung verursacht.

Bei EXASOL sind alle Daten automatisch im Hauptspeicher abgelegt. Die hierfür am besten geeigneten Kompressions-Algorithmen werden von der Datenbank automatisch gewählt. Das verschafft einen klaren Performance-Vorteil: Geänderte und neue Daten landen nicht zuerst auf der Festplatte, sondern direkt im Hauptspeicher und stehen somit nach einem Commit sofort zur Verfügung. Alle Daten-Objekte, auf die ein Zugriff erfolgt, werden in den Hauptspeicher geladen und bleiben dort für nachfolgende Abfragen bestehen. Trotzdem kommt es hier nicht zu Query-Abbrüchen; sollte der Hauptspeicher für die Abfragen nicht ausreichen, sinkt lediglich die Performance des Systems etwas.

Zusätzliche Geschwindigkeit gewinnt diese Datenbank-Lösung durch die Tatsache, dass die Daten-Strukturen im Hauptspeicher denen auf der Festplatte entsprechen und Daten somit hoch performant durch Memory-Mapping ein- und ausgelad-

ert werden können. Der Prozessor wird hierbei nicht belastet. Dieser Ansatz verschafft mehr Performance – benötigt aber mehr Platz im Hauptspeicher.

Ein weiterer Unterschied ist die Tatsache, dass es 12c-Nutzern ermöglicht wird, die eigenen Daten durch eine Partitionierung in Hot- und Cold-Data zu unterteilen, wobei nur die Hot-Data-Partitionen in den Hauptspeicher geladen werden und der Rest auf der Festplatte verbleibt. Dank dieser Möglichkeit wird kein unnötiger Speicherplatz im Hauptspeicher blockiert. Bei EXASOL hingegen besteht die Möglichkeit zur Partitionierung nicht. Aus diesem Grund kann hier durch den selteneren Zugriff auf Cold-Data wichtigeres Datenmaterial aus dem Arbeitsspeicher verdrängt werden. Es muss so bei der nächsten Abfrage erst wieder eingelagert werden. Dieses Problem hat 12c nicht – aber die Möglichkeit zur Partitionierung bringt auch Nachteile mit sich: Bei Abfragen stehen nämlich nur die Daten-Objekte zur Verfügung, für die explizit festgelegt wurde, dass sie im Hauptspeicher gehalten werden sollen. Alle weiteren Daten müssen bei jeder Abfrage erneut von der Festplatte oder aus dem Cache eingelesen und verarbeitet werden, was die Bearbeitungszeit deutlich erhöht.

Die beiden Systeme unterscheiden sich unter anderem auch in ihren Lizenzierungsmodellen: Die Lizenzierung bei EXASOL ist ausschließlich RAM-basiert, während Oracle pro CPU berechnet. Oracle 12c ist eine sehr flexible Lösung, die gut an individuelle Bedürfnisse angepasst werden kann, dafür aber mit klarem manuellen Konfigurationsaufwand verbunden ist. Die Flexibilität, aber auch der manuelle Konfigurationsaufwand der Oracle-Datenbank sind relativ hoch und einige Fragen müssen wiederholt geklärt werden:

- Wie viel RAM soll/darf verwendet werden?
- Welche Tabellen und welche Spalten sollen im Speicher gehalten werden?
- Wie sollen diese komprimiert werden?
- Wann sollen die Daten in den Speicher geladen werden? („Lazy loading“ bei Zugriff oder Laden beim Starten der Datenbank?)

Anwender können und müssen hier folglich viele Dinge selbst entscheiden und einstellen.

Bei EXASOL hingegen ist eine dezidierte Optimierung auf ein spezifisches Anwendungsszenario nicht möglich. Dafür entfällt aber auch der andauernde manuelle Konfigurationsaufwand: Dank der umfangreichen Oracle-Kompatibilität, dem nativen OCI-Support (Oracle Call Interface) und der Tatsache, dass die SQL-Abfragen beider Datenbanken syntaktisch nahezu „1:1“ übereinstimmen, kann die Lösung einfach, schnell und mit geringem Aufwand in eine bestehende Oracle-Infrastruktur integriert werden und bearbeitet Anfragen danach automatisch und quasi ohne manuellen Aufwand.

Welche Datenbank ist die bessere Alternative?

Die Einführung von Oracle 12c entbindet Oracle-Nutzer nicht zwingend von einer intelligenten Wahl der zu nutzenden In-Memory-Lösung. Wie am Beispiel von EXASOL beschrieben, bieten auch andere In-Memory-Lösungen anwendungsspezifische Vorteile und können, wie in diesem Fall, zum Beispiel komplementär als analytischer Performance Layer („Accelerator“) oberhalb von Oracle verwendet werden und somit eine größtmögliche Geschwindigkeit innerhalb des Oracle-Umfelds garantieren oder alternativ für bestimmte Anwendungen als paralleles System zu einer Oracle-Umgebung eingesetzt werden (*siehe Abbildung 2*).

Verschiedene In-Memory-Datenbanken bieten verschiedene anwendungsspezifische Vorteile und weisen gleichzeitig unterschiedliche Nachteile auf. So ist, wie auch John Appleby vom SAP-Beratungsunternehmen bluefin zu berichten weiß (*siehe „<http://www.bluefinsolutions.com/blogs/john-appleby/october-2014/oracle-database-in-memory-faq>“*), die In-Memory-Option von Oracle eher für einfachere Abfragen ausgelegt und kann bei komplexen analytischen Abfragen mit Performance-Einbußen zu kämpfen haben. Tatsächlich kann die Performance hier so weit abfallen, dass sie schlechter ist als bei Oracle-Systemen ohne In-Memory-Technologie. Zudem ist der manuelle Konfigurationsaufwand der Oracle-Lösung beträchtlich. Dafür besticht die Datenbank mit ihrer Flexibilität und Anpassungsfähigkeit an individuelle Bedürfnisse sowie mit der relativ geringen Hauptspeicherplatz-Belastung. Die Datenbank von EXASOL hingegen benötigt mehr

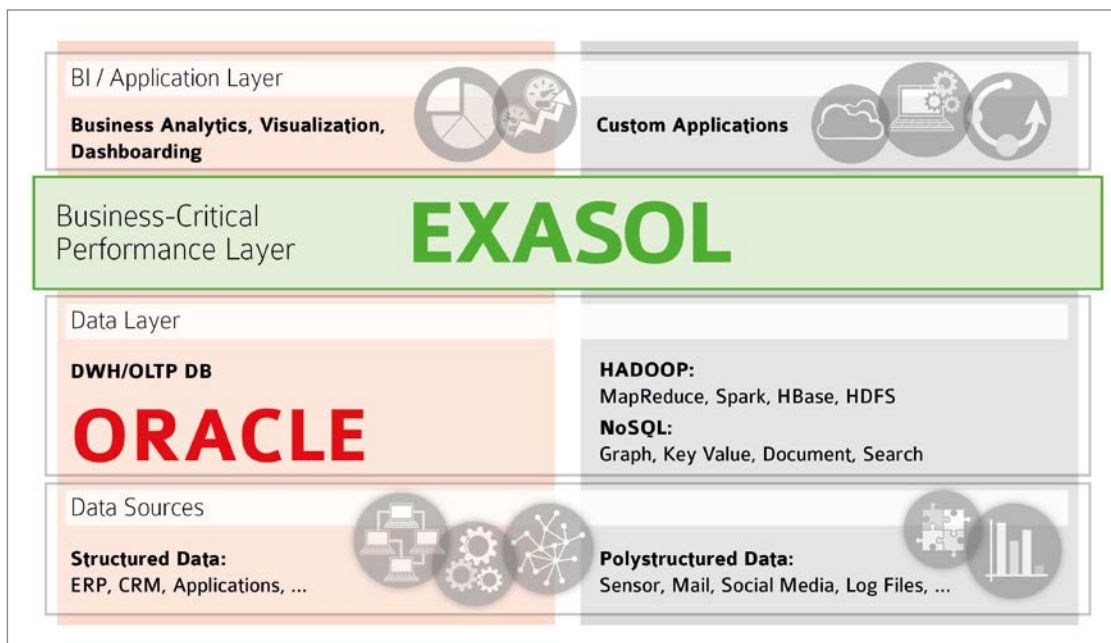


Abbildung 2: EXASOL im Einsatz als Add-on-Technologie in einer Oracle-Umgebung zur Beschleunigung der Daten-Analyse

Platz im Hauptspeicher und weist eine geringere Flexibilität auf als Oracle 12c. Dafür ist sie einfach implementiert und arbeitet fast ausschließlich selbstständig – der manuelle Konfigurationsaufwand entfällt hier. Außerdem beeindruckt sie durch enorme Performance und ist explizit für komplexe, analytische Abfragen ausgelegt.

Welche In-Memory-Lösung die richtige für das eigene Unternehmen ist, muss jeder selbst entscheiden – doch diese Entscheidung sollte durchdacht sein und am besten über eine Evaluierung der Bestandslage im Vorfeld einer geplanten Erweiterung oder Umstellung getroffen werden. Ein sogenannter „Proof of Concept“ gibt Aufschluss über das Potenzial der In-Memory-Lösung im Unternehmen und plant das Projekt zudem personell und monetär ein. Die richtige Wahl der In-Memory-Datenbank, basierend auf den Bedürfnissen des Unternehmens und auf Informationen über die Vor- und Nachteile verschiedener Lösungen, hilft Unternehmen, sich den größten Mehrwert zu sichern, den Analytik und In-Memory zu bieten haben.

Anwenderbeispiel: Accarda AG

Ein Anwender, der sich entschlossen hat, Oracle und EXASOL zu kombinieren, ist der Schweizer Finanzdienstleister Accarda. Um Umsatzverluste im Einzelhandel zu vermeiden, entwickelte Accarda in Zusam-

menarbeit mit der Manor AG das „Accarda Loss Prevention System“, kurz ALPS. Diese Lösung ermittelt auffällige Trends, Unregelmäßigkeiten und Abweichungen in den Bezahlprozessen, um fehlerhafte Abrechnungen zu identifizieren. Hierzu müssen Ad-hoc-Analysen auf enormen Datenmengen ausgeführt werden. Aus diesem Grund benötigt ALPS eine hochleistungsfähige Datenbank im Hintergrund und setzt seit rund zwei Jahren auf die analytische In-Memory-Lösung aus Nürnberg.

Die Datenbank dient dem ALPS, um den Prozess flexibel und automatisch zu gestalten. Zu Beginn des Prozesses liefern Kunden ihre Kassendaten an. Accarda bereitet die Daten auf und analysiert sie hinsichtlich Strukturbrüchen, definierter Kennzahlen, Unregelmäßigkeiten oder Schwellenwertüberschreitungen. Registriert ALPS auffällige Kassentransaktionen oder Abläufe, so werden diese automatisch auf einem Report ausgewiesen und die zuständigen Stellen benachrichtigt. Die Reporte stehen täglich zur Verfügung. Zudem können Kunden vertiefende Ad-hoc-Analysen durchführen, um konkrete Nachforschungen anzustoßen oder Maßnahmen zur Prävention von Diebstahl einzuleiten. Damit diese Prozesse schnell ablaufen, bedarf es einer skalierbaren Lösung, die sämtliche Analysen auf Ebene der Einzeltransaktionen unterstützt und Daten „on-the-fly“ analysiert. So profitiert der Finanzdienstleister in erster Linie davon, dass Daten-Aggrega-

tionen überflüssig werden beziehungsweise die Materialisierung der Daten entfällt. Lade- und Analysezeiten wurden erheblich beschleunigt. Die Handelspartner des Unternehmens profitieren wiederum von ständiger Verfügbarkeit und Analysen, die bis auf die Ebene der Einzeltransaktionen reichen. Laut Stefan Schurgast, dem Business Development und Innovation Manager der Accarda AG, war die Oracle-Kompatibilität von EXASOL das Hauptkriterium für die Auswahl dieser Datenbank. Sie unterstützt zudem Standard-SQL und läuft auf Standard-Hardware. Daher ist es möglich, auch nach der Integration bereits bestehende ETL-Tools weiterhin zu nutzen. Außerdem erklärt Schurgast, hat ihn auch die Vision mit den vier Eckpfeilern Clustering, MPP, spaltenorientierte Speicherung und intelligente Kompressionsalgorithmen sowie der In-Memory-Komponente überzeugt.



Mathias Golombek
mathias.golombek@exasol.com



Oracle-Forms-Trends von der DOAG 2016 Konferenz und wohin die Reise geht

Frank Hoffmann, Cologne Data GmbH

Oracle hat mit der Forms-Version 12c vor einem Jahr die Basis für den unterstützten Betrieb von Forms bis mindestens zum Jahr 2025 gelegt. Mit der Verzögerung von einem Jahr ist diese Version nun auch bei den deutschen Forms-Kunden angekommen und bei der großen Mehrheit als zukünftige Version gesetzt. Bei einer spontanen Umfrage auf der DOAG 2016 Konferenz gaben etwa neunzig Prozent der Forms-Kunden an, Anfang 2017 auf Forms 12c wechseln zu wollen.

Mit den Versionen 12.2.1.1 und 12.2.2.1 sind alle aktuellen bekannten Bugs behoben und die Caching-Funktion für FSAL auf vielfachen Kundenwunsch realisiert (browserfreie Aufruf-Technologie). Die Kombination von FSAL und Caching kann Forms-Aufrufe auf drei bis fünf Sekunden minimieren.

Oracle Forms steht nun langfristig vor einem großen Technologie-Umbruch. Die Applet-Technologie, die bereits im Jahr 2000 mit Forms 6i eingeführt wurde, kommt für eine Nachfolge-Version basierend auf Java 10 nicht mehr infrage. Java 9 wird die letz-

te Applet-unterstützte Version sein. Die aktuelle Forms-Version läuft noch unter Java 9 – wird es aber unter Java 10 nicht mehr tun.

Mit Java 10 (ab etwa 2025 bis 2030) wird das Oracle-Forms-Entwicklerteam also eine neue Client-Technologie einführen müssen. Gut, dass dabei eng mit dem Java- und dem WebLogic-Team zusammengearbeitet wird. Aktuell ist die neue Version noch nicht in Planung – sie wird jedoch weiter eine Server-Komponente auf Basis des WebLogic-Servers haben sowie eine neue Anzeige-Technologie wie HTML5 oder SVG. Damit wird sich auch

das „Look and Feel“ grundlegend ändern und zwingend ein neuer Builder eingeführt werden müssen. Applikationen werden dann in einem neuen, zeitgemäßen Gewand erscheinen.

Mit dem Umstieg auf Java 10 wird dann die Geschichte von Forms noch mindestens weitere zwanzig Jahre geschrieben und Oracle Forms wird seinen fünfzigsten Geburtstag feiern. Die Frage, die sich die Forms-Kunden heute stellen, ist also nicht mehr Migration von Forms nach „xxx“, sondern das Upgrade nach Forms 12c. Die DOAG unterstützt diesen Trend

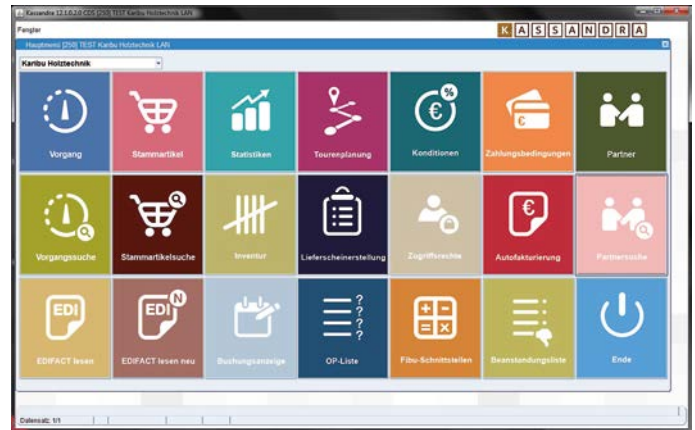
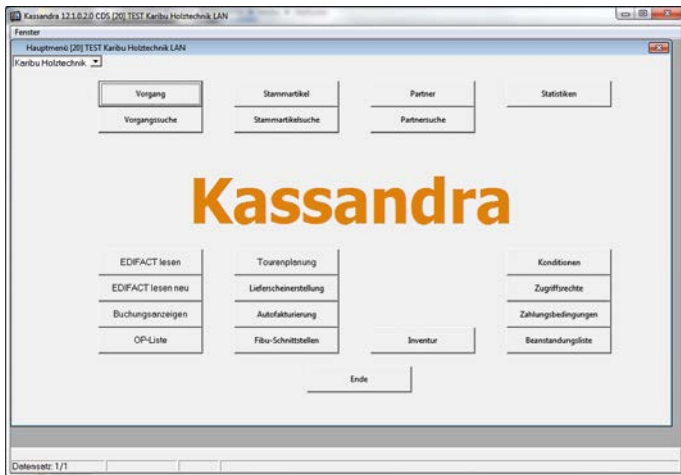


Abbildung 1: Beispiel einer Modernisierung von Forms 6i (links) auf Forms 12c (rechts)

mit einer neuen Forms-Gruppe und weiteren Aktivitäten wie einem Forms Day und einem Demoserver.

Wichtig ist nun, die alte Forms-Software auf den Stand der Version 12c zu bringen und mit Modernisierungen möglichst eng an der Kerntechnologie zu bleiben. Mit Forms 12c und der integrierten „webutil“-Funktionalität lassen sich fast alle Anforderungen mit PL/SQL-Logiken realisieren. Modernisierungen müssen genau auf Nachhaltigkeit geprüft werden, um zukünftige Migrationen kostengünstig gestalten zu können. *Abbildung 1*

zeigt das Beispiel einer Modernisierung der Einstiegsmaske mit Grafikgestaltung.

Fazit

Die mehr als 100.000 Forms-Anwendungen weltweit können in den kommenden Jahren sicher weiter genutzt werden und der Autor stimmt Robert Johansson von Softbase (DK) in seiner Aussage auf der DOAG 2016 Konferenz zu: „Für Oracle-basierte Unternehmens-Desktop-Applikationen ist Forms die erste Wahl.“



Frank Hoffmann
frank.hoffmann@cologne-data.de

Termine

März

07.03.2017 - 08.03.2017
Berliner Expertenseminar mit Alexander Kornbrust: Härten von Oracle 12c Datenbanken (Erstellung Policies, Überprüfung) | Berlin
Cornel Albert
expertenseminare@doag.org

08.03.2017
Regionaltreffen Berlin/Brandenburg
Michel Keemers
regio-bb@doag.org

Bis zum 10.03.2017
AOUG Anwenderkonferenz, CALL FOR PAPERS
Wien
www.aoug.at

10.03.2017
DOAG Vorstandssitzung | Berlin
Hannover
office@doag.org

10.03.2017
DOAG Webinar: Oracle 12c In-Memory Option Architektur
Christian Pfundtner

13.03.2017
Regionaltreffen München/Südbayern
Franz Hüll, Andreas Ströbel
regio-muenchen@doag.org

13.03.2017
Regionaltreffen Osnabrück/Bielefeld/Münster
Andreas Kother, Klaus Günther
regio-osnabrueck@doag.org

14.03.2017 - 15.03.2017
DOAG Middleware Days
DOAG Geschäftsstelle
office@doag.org

14.03.2017 - 15.03.2017
Oracle Data Vision 2017
Düsseldorf
office@doag.org

16.03.2017
Regionaltreffen Stuttgart
Jens-Uwe Petersen, Anja Stollberg
regio-stuttgart@doag.org

22.03.2017
DOAG Database Day 2017
Dr. Matthias Mann
matthias.mann_v-tservices@de.ibm.com

23.03.2017
Regionaltreffen Dresden/Sachsen
Helmut Marten
regio-sachsen@doag.org

Wir begrüßen unsere neuen Mitglieder

Persönliche Mitglieder

- Achim Bloch-Späth
- Peter Schneider
- Raimund Kolbe
- Gabriela Munoz
- Oliver Szymanski
- Chukwu Kalu
- Sabine Hunsicker
- Thies Wellpott
- Markus Hohloch
- Ozren Pestic

Firmenmitglieder DOAG

- RS Media Marketing Services GmbH, Torsten Strychik
- Unternehmensberatung Haberstock GmbH, Markus Haberstock

Impressum

Red Stack Magazin wird gemeinsam herausgegeben von den Oracle-Anwendergruppen DOAG Deutsche ORACLE-Anwendergruppe e.V. (Deutschland, Tempelhofer Weg 64, 12347 Berlin, www.doag.org), AOUG Austrian Oracle User Group (Österreich, Lassallestraße 7a, 1020 Wien, www.aoug.at) und SOUG Swiss Oracle User Group (Schweiz, Dornacherstraße 192, 4053 Basel, www.soug.ch).

Red Stack Magazin ist das User-Magazin rund um die Produkte der Oracle Corp., USA, im Raum Deutschland, Österreich und Schweiz. Es ist unabhängig von Oracle und vertritt weder direkt noch indirekt deren wirtschaftliche Interessen. Vielmehr vertritt es die Interessen der Anwender an den Themen rund um die Oracle-Produkte, fördert den Wissensaustausch zwischen den Lesern und informiert über neue Produkte und Technologien.

Red Stack Magazin wird verlegt von der DOAG Dienstleistungen GmbH, Tempelhofer Weg 64, 12347 Berlin, Deutschland, gesetzlich vertreten durch den Geschäftsführer Fried Saacke, deren Unternehmensgegenstand Vereinsmanagement, Veranstaltungsorganisation und Publishing ist.

Die DOAG Deutsche ORACLE-Anwendergruppe e.V. hält 100 Prozent der Stammeinlage der DOAG Dienstleistungen GmbH. Die DOAG Deutsche ORACLE-Anwendergruppe e.V. wird gesetzlich durch den Vorstand vertreten; Vorsitzender: Stefan Kinnen. Die DOAG Deutsche ORACLE-Anwendergruppe e.V. informiert kompetent über alle Oracle-Themen, setzt sich für die Interessen der Mitglieder ein und führen einen konstruktiv-kritischen Dialog mit Oracle.

Redaktion:

Sitz: DOAG Dienstleistungen GmbH
(Anschrift s.o.)
Chefredakteur (ViSdP): Wolfgang Taschner
Kontakt: redaktion@doag.org
Weitere Redakteure (in alphabetischer Reihenfolge): Gaetano Bisaz, Mylène Diacquenod, Marina Fischer, Klaus-Michael Hatzinger, Sebastian Höing, Fried Saacke

Titel, Gestaltung und Satz:

Alexander Kermas, DOAG Dienstleistungen GmbH (Anschrift s.o.)

Fotonachweis:

Titel: © rawpixel/123rf.com
Foto S. 12: © lculig/123rf.com
Foto S. 14: © <https://svn.apache.org/repos/asf/cassandra/logo/cassandra.svg>
Foto S. 19: © Rattanapon Muanpimthong/123rf.com
Foto S. 23: © attaphong/123rf.com
Foto S. 27: © Andriy Popov/123rf.com
Foto S. 46: © Tawatchai Khid-arn/123rf.com
Foto S. 68: © peterhowell/istockphoto.com
Foto S. 72: © Dusanka Visnjican/123rf.com

Anzeigen:

Simone Fischer, DOAG Dienstleistungen GmbH (verantwortlich, Anschrift s.o.)
Kontakt: anzeigen@doag.org
Mediadaten und Preise unter: www.doag.org/go/mediadaten

Druck:

adame Advertising and Media GmbH,
www.adame.de

Alle Rechte vorbehalten. Jegliche Vervielfältigung oder Weiterverbreitung in jedem Medium als Ganzes oder in Teilen bedarf der schriftlichen Zustimmung des Verlags. Die Informationen und Angaben in dieser Publikation wurden nach bestem Wissen und Gewissen recherchiert. Die Nutzung dieser Informationen und Angaben geschieht allein auf eigene Verantwortung. Eine Haftung für die Richtigkeit der Informationen und Angaben, insbesondere für die Anwendbarkeit im Einzelfall, wird nicht übernommen. Meinungen stellen die Ansichten der jeweiligen Autoren dar und geben nicht notwendigerweise die Ansicht der Herausgeber wieder.

Inserentenverzeichnis

dbi services ag www.dbi-services.com	S. 25	E-3 Magazin www.e-3.de	S. 31	Robotron Datenbank-Software GmbH S. 57 www.robotron.de
DOAG e.V. www.doag.org	U 2, U 3	MuniQsoft GmbH www.muniqsoft.de	S. 3	Trivadis GmbH www.trivadis.com U 4



JavaLand

28.-30. März 2017 in Brühl



Erfolg durch Know-how.

Trivadis Trainings | 2017



■ Oracle Cloud Computing, Migration auf Oracle 12C, Fusion Middleware + SOA, Business Intelligence, High Availability, Security... das alles und noch viel mehr bieten Ihnen die neuen Trivadis Trainings. Wir verwandeln Wissen in Können. Für Ihren Erfolg. In unseren modernen Trainingszentren in Ihrer Nähe oder als E-Learning im Virtual Classroom. Mit Grundlagen für Ihren Nachwuchs und gezielten Weiterbildungsangeboten für Ihre Spezialisten.



Jetzt den neuen Trainingskatalog downloaden:
trivadis.com/de/trainingskatalog

Oder direkt anmelden:
trivadis.com/de/training



BASEL ■ BERN ■ BRUGG ■ DÜSSELDORF ■ FRANKFURT A.M. ■ FREIBURG I.B.R. ■ GENÈVE
HAMBURG ■ KOPENHAGEN ■ LAUSANNE ■ MÜNCHEN ■ STUTTGART ■ WIEN ■ ZÜRICH

trivadis
makes IT easier. ■ ■ ■