

# DOAG SOUG News

## Big Data



### Praxis

Social-Media-Auswertungen  
im Data Warehouse

### Im Interview

Dr.-Ing. Thomas Seeger,  
Deichmann SE



### Virtualisierung

SPARC-Server in  
hyperkonvergenten  
VM-Infrastrukturen



# NOON<sup>2</sup>NOON

## RAC & Dataguard

### Noon:

Nach einer kurzen Einführung in den Tag sowie einem kräftigenden Mittagessen geht es in die erste Runde: Die Grundlagen zu RAC werden vorgestellt. Im Anschluss folgt ein "RAC Attack"-Workshop. Unterstützt durch erfahrene Ninjas installieren die Teilnehmer ihren eigenen RAC mit virtuellen Maschinen (VM) auf ihren Notebooks.

### Mid:

Am Abend steht Networking bei leckerem Essen im Seminarraum auf dem Programm. Probleme werden direkt auf den VMs nachgestellt und geklärt – ein Erfahrungsaustausch, der keine dummen Fragen kennt.

### Noon:

Wie war das mit Dataguard und der Maximum Availability Architecture (MAA)? Darum geht es am Morgen des zweiten Tages. Erneut gibt es eine kurze theoretische Einführung, bevor am eigenen Notebook eine DG-Installation auf zwei VMs durchgeführt wird. Der offizielle Teil endet mit dem Mittagessen.

**GHOTEL hotel & living in Würzburg**  
21. - 22. Januar 2016 von 12 Uhr bis 12 Uhr

**Experten vor Ort:**  
Martin Bach (RAC) und Björn Rost (RAC)



<http://www.doag.org/go/noon2noon>



Wolfgang Taschner  
Chefredakteur  
DOAG/SOUG News

## Aus der DOAG/SOUG News wird das Red Stack Magazin

Liebe Mitglieder, liebe Leserinnen und Leser,

gestatten Sie mir zunächst einen Blick in die Geschichtsbücher: Im Januar 1988 kommen rund 80 Anwender zusammen und stellen die Weichen für die Gründung einer deutschen ORACLE-Anwendergruppe. Einige kümmern sich um die Registrierungsformalien als eingetragener Verein, andere gründen bereits die ersten Arbeitsgruppen. Die AG Newsletter bringt noch im selben Jahr die erste Ausgabe der DOAG News heraus. Darin veröffentlicht die AG Technik eine „Wishlist“ mit neuen SQL-Funktionen an Oracle. Eva Kraut, die erste Vorsitzende der DOAG, stellt die neue Satzung des Vereins vor und Bernd G. Wenzel schildert seine Eindrücke von der International Oracle User Week in Orlando, Florida.

Die Zeitschrift macht Fortschritte und kommt einige Jahre später mit vier Ausgaben pro Jahr heraus. Weitere Meilensteine sind das neue Layout ab dem Jahr 1998 und die Abspaltung der Business-Themen in die DOAG Business News, die zunächst als Sonderheft und später regelmäßig mehrmals jährlich erscheint. Ab 2012 kommt die DOAG News auf sechs Ausgaben im Jahr, es werden Schwerpunktthemen eingeführt und im Jahr 2015 erfolgt der Zusammenschluss mit der Swiss Oracle User Group (SOUG). Die Zeitschrift wird unter dem Namen „DOAG/SOUG News“ auch an die Mitglieder der SOUG verteilt.

Eines ist aber bei allen Ausgaben gleich geblieben: Die Zeitschrift war, ist und bleibt eine Plattform für die Wissensvermittlung und den Erfahrungsaustausch der Anwender aller technischen Oracle-Produkte.

Warum nun schon wieder ein neuer Name für die Zeitschrift? Ab dem kommenden Jahr ist die Austrian Oracle User Group (AOUG) mit an Bord. Ein Titel namens „DOAG/SOUG/AOUG News“ passt nicht wirklich, von daher haben sich die Vorstände der drei Anwendergruppen auf „Red Stack Magazin“ verständigt. Abgesehen vom Titel wird sich inhaltlich nichts ändern. Neu ist allerdings, dass die österreichischen Anwender jetzt ebenfalls von den Artikeln profitieren können.

Ihr

W. Taschner

# MUNIQSOFT

## Support

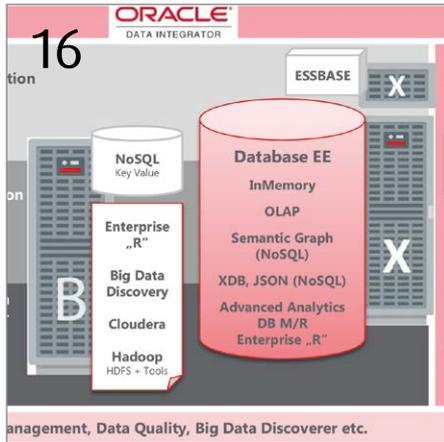
Mit **IQ** vermeiden Sie Notfälle und erhalten Hilfe, wenn's brennt.

+49 (0) 89 6228 6789-0

[www.munisoft.de/support](http://www.munisoft.de/support)

### Unser Kundenservice

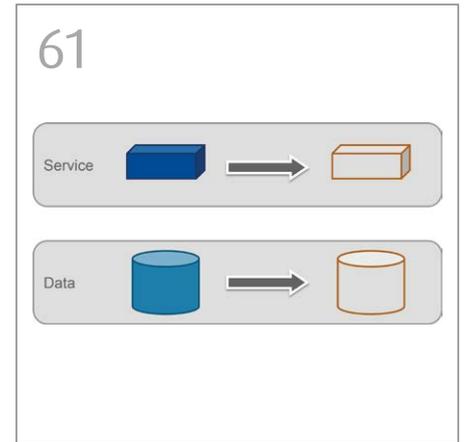
- ▶ Deutschsprachiger Telefonsupport
- ▶ Garantierte Reaktionszeiten
- ▶ 3rd- und 2nd-Level Support
- ▶ Regelmäßige Datenbank-Healthchecks
- ▶ Proaktives Datenbank-Monitoring und Patch-Management
- ▶ Incident- und Problemmanagement



Oracle Data Warehouses und Big Data im Zusammenspiel



Die neueste Version der Oracle-Datenbank unterstützt erstmals nativ das JSON-Format



Die Oracle Database Appliance unterstützt mehrere hochverfügbare Lösungen

## Einleitung

- 3 Editorial
- 5 DOAG/SOUG Timeline
- 8 „Die Datenbank mit der darin integrierten Business-Logik ist ein gutes Fundament für unsere Geschäftsprozesse ...“ Interview mit Dr. Ing. Thomas Seeger

## Big Data

- 11 Chancen und Grenzen: Warum gerade jetzt Big Data  
Alfred Schlaucher
- 16 Oracle Data Warehouses und Big Data im Zusammenspiel  
Peter Welker
- 22 Produkt-Update für die Oracle-Big-Data-Plattform  
Jean-Pierre Dijcks
- 24 Oracle-Datenbank – Schaltzentrale für Big Data, In-Memory und Exadata  
Dr.-Ing. Holger Friedrich
- 30 Social-Media-Auswertungen im Data Warehouse  
Martin Frisch

- 33 Hadoop-Einstiegshürden meistern – mit Business Case, Skill-Aufbau und der richtigen Technologie zu Big Data  
Oliver Herzberg und Slavomir Nagy
- 37 Aufbau eines Semantic-Layers zwischen Datenbank und Hadoop  
Matthias Fuchs
- 41 Native JSON-Unterstützung in Oracle12c  
Carsten Czarski
- 45 Zweiter Frühling für die Analyse unstrukturierter Daten  
Christopher Thomsen

- 50 Die Oracle-Datenbank jenseits von Entity-Relationship-Modellierung – vorhandenes Wissen repräsentieren und neues mit RDF Graph generieren  
Karin Patenge

## Virtualisierung

- 55 SPARC-Server in hyperkonvergenten VM-Infrastrukturen  
Bert Miemietz

## Datenbank

- 61 Data Guard auf der ODA  
David Hueber
- 65 High Performance Spatial Computing mit Oracle 12c  
Eva-Maria Kramer
- 71 Tipps & Tricks aus Gerds Fundgrube Heute: Undo  
Gerd Volberg

## DOAG/SOUG intern

- 70 Neue Mitglieder
- 72 Impressum
- 72 Termine
- 72 Inserentenverzeichnis

# ✦ DOAG/SOUG Timeline

## 7. September 2015

Das Programm-Komitee der Java-Community legt bei einem Treffen in Berlin das Gesamtkonzept für die JavaLand 2016 fest, die vom 8. bis 10. März 2016 im Phantasialand in Brühl stattfinden wird. Es gibt wieder ein sehr interessantes und spannendes Programm, das einen guten Mix aus renommierten Speakern aus dem deutschsprachigen und internationalen Raum enthält und das deutlich mehr Themen als im letzten Jahr abdeckt. Auch die begleitenden Community-Aktivitäten sind noch weiter ausgebaut.



Das Organisationsteam der letztjährigen JavaLand-Konferenz

## 9. September 2015

Dr. Dietmar Neugebauer, Vorstandsvorsitzender der DOAG, und Fried Saacke, DOAG-Vorstand und Geschäftsführer, treffen sich in München mit Michael Beilfuß, Leiter der IDG Communications Verlag AG, um die Zusammenarbeit mit der Zeitschrift „Computerwoche“ zu besprechen.

## 10. September 2015

In Baden kommen die SOUG-Mitglieder zur dritten und letzten SIG des Jahres 2015 zusammen. Die Teilnehmer der Veranstaltung haben die Möglichkeit, sich zu den Themen „Development und Deployment“ sowie „SOA und Mobile Computing“ zu informieren. Nach vielen neuen und interessanten Informationen aus acht Vorträgen werden bei einem abschließenden Aperó, gesponsert von der Firma esentri AG, die Themen diskutiert. Das Netzwerken mit Kollegen kommt dabei natürlich auch zum Zug.



Thomas Kucher von der GreenBee Solutions GmbH bei seinem Vortrag

## 10./11. September 2015

Die DOAG-Vorstandssitzung findet in Karlsruhe statt. Schwerpunkt der ersten gemeinsamen Sitzung nach der Vorstandswahl ist die Umsetzung der strategischen Ziele aus der Delegiertenversammlung für die kommenden Jahre.

## 15./16. September 2015

Oliver Lemm, Competence Center Leiter der MT AG, leitet das Berliner Expertenseminar zum Thema „Professionelle Apex-Entwicklung und Apex 5.0“. Die Teilnehmer können das Gelernte gleich auf ihren mitgebrachten Notebooks umsetzen.

## 17./18. September 2015

Nach der erfolgreichen zweitägigen Big-Data-Veranstaltung mit Hands-on-Workshops im vergangenen Herbst, bei der die Grundlagen und Möglichkeiten der Hadoop-Welt im Vordergrund standen, treffen sich die Big-Data-Interessierten diesmal in Hannover. Nach einer kurzen Einführung in die Grundlagen beleuchten spannende Vorträge sowie drei Hands-on-Workshops die Verbindungsmöglichkeiten zwischen SQL und Big Data aus Sicht der Oracle-Datenbank-Entwickler und Administratoren. Themen sind unter anderem „Einfache Konnektoren“, „Komplexe Integrationswerkzeuge wie ODI“, „Echte Database-Links von Oracle DB zu NoSQL-DBs und Hadoop“, „Big Data SQL“ und „JSON in der Oracle-Datenbank“. Beim Abendevent entwickeln sich in einer lockeren und angenehmen Atmosphäre viele interessante Gespräche.

## 21. September 2015

Der DOAG Cloning Day öffnet in Leipzig seine Türen für alle, die Datenbank-Kopien schnell zur Verfügung stellen oder Produktions-Datenbanken kopieren müssen. Dabei sollen die Ressourcen auf das unbedingt Notwendige beschränkt und trotzdem gegen den gesamten Datenbestand getestet werden. Die teilnehmenden Hersteller bieten dafür Lösungen an, die sie in den Vorträgen präsentieren. Zum Abschluss werden die vorgestellten Verfahren umfassend diskutiert.

## 22. September 2015

Alles Wissenswerte über Datensicherheit und Datenschutz, typische Angriffsszenarien und deren Auswirkungen sowie Forensik gibt es auf dem DOAG Security Day, der als Folgeveranstaltung zum DOAG Cloning Day ebenfalls in Leipzig stattfindet. Die Teilnehmer beider Events treffen sich am Vorabend in gemütlicher Runde in einem Restaurant in der Innenstadt.

### 23./24. September 2015

Als Kombi-Veranstaltung finden in Stuttgart der DOAG Middleware Day und am Folgetag der DOAG OpenStack Day statt. Der erste OpenStack Day führt Oracle-Anwender an das Hype-Thema heran. Die Veranstaltung richtet sich an Architekten oder technische Entscheider, die sich mit OpenStack auseinandersetzen. Zusätzlich zu Grundlagen und Herangehensweise vermittelt ein Live-Workshop erste Praxiserfahrungen.

### 29./30. September 2015

Das Berliner Expertenseminar zum Thema „OBIEE Repository und Reports Master Class“ bietet tiefgehendes OBIEE-Wissen von einem Experten, der seit deutlich mehr als zehn Jahren BI-Projekte durchführt. Gerd Aiglstorfer ist offizieller OBIEE-Expert-Trainer und Sprecher auf weltweit bekannten BI-Konferenzen. Mit dem Seminar erhält jeder, der die Oracle-BI-Werkzeuge einsetzt, Einblick in deren optimale Nutzung und Konfiguration.

### 5. Oktober 2015

In der monatlichen Telko mit Tom Schiersen, EMEA User Groups Relationship Manager von Oracle, geht es um die Zusammenarbeit zwischen Oracle und den europäischen Usergroups sowie um den aktuellen Stand der großen Oracle-Veranstaltungen.

### 6./7. Oktober 2015

Im Berliner Expertenseminar geben Robert Bialek, Oracle Certified Master und Red Hat Certified Engineer, und Mathias Zarick, Oracle Certified Professional DBA 9i, 10g und 11g sowie Oracle Certified Master 11g, tiefe Einblicke zur Hochverfügbarkeit der Oracle-Datenbank. Die Unterlagen werden auf USB-Sticks gespeichert und an die Teilnehmer verteilt.

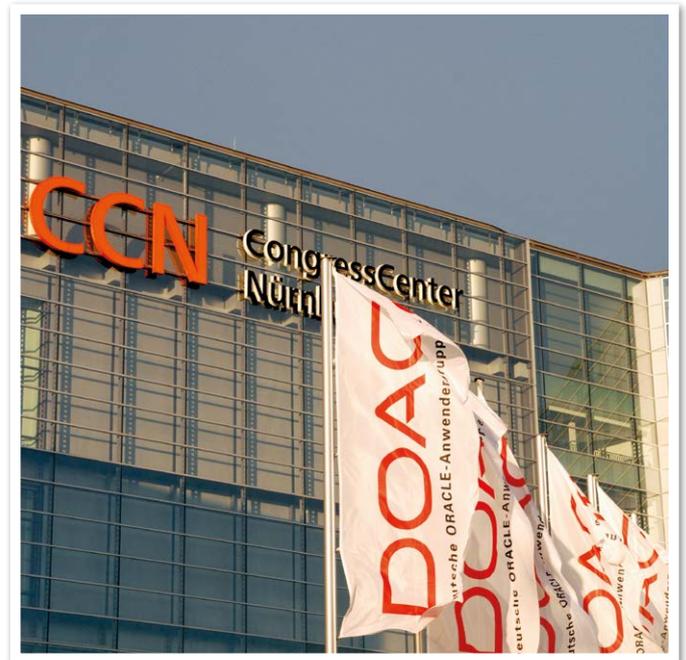
### 8. Oktober 2015

Dr. Dietmar Neugebauer, Vorstandsvorsitzender der DOAG, nimmt am Business-Frühstück der Austrian Oracle User Group (AOUG) über die Rechtssituation der Oracle-Lizenzierung in VMware-Umgebungen teil. Dr. Clemens Appl von der Wiener Wirtschaftsuniversität stellt in einer Stunde die Eckpunkte seiner Arbeit vor. Nach Themen wie TOMA und Lizenzierung (Prozessoren etc.) ging Dr. Appl auf die Systemvereinbarung dauerhaft überlassener Software ein. Er vertritt die Auffassung, dass es nach dem Kauf keine Einschränkung der Nutzung bezüglich der Anzahl der Prozessoren gibt, was auch seine Argumentation zur Verwendung der gekauften Software in virtuellen Umgebungen ist. In der anschließenden Diskussion betonen er und die AOUG nochmals, dass das erstellte Papier kein Garant für einen Erfolg vor Gericht ist. Es bietet aber die Möglichkeit, gegenüber Oracle eine bessere Verhandlungsposition zu haben und bessere und akzeptable Lösungen zu erzielen.



### 9. Oktober 2015

Das Organisations-Team der DOAG kommt mit Vertretern und Dienstleistern des Nürnberg Convention Center zu einem Kick-off-Meeting zusammen, um die Durchführung der DOAG 2015 Konferenz + Ausstellung zu besprechen. Die Teilnehmer sind davon überzeugt, auch in diesem Jahr wieder optimale Rahmenbedingungen für die Jahreskonferenz der DOAG bieten zu können. Erstmals ist auch Oracle mit vertreten, um den integrierten Oracle Cloud Day zu besprechen.



Das Nürnberg Convention Center, Standort der DOAG 2015 Konferenz + Ausstellung

### 13. Oktober 2015

Seine intensive Beschäftigung mit der Oracle-Datenbank hat bislang zu zwei Veröffentlichungen in diesem Bereich geführt: Oracle SQL und Oracle PL/SQL – das umfassende Handbuch. Jürgen Sieben hält das Berliner Expertenseminar über „PL/SQL-Objekte für jedermann“. Der Schwerpunkt liegt auf Beispielen, die die Teilnehmer in ihren Projekten verwenden können. So demonstriert er die grundlegenden Techniken der Objektorientierung am Beispiel eines endlichen Automaten (finite state machine) in der Datenbank, einem Entwurfsmuster, das oft verwendet werden kann und das besonders im Umfeld einer Datenbank spannende Optionen ermöglicht. Das Beispiel dient dazu, Objekte durch einen datengesteuerten Arbeitsablauf zu schleusen.

### 25. Oktober 2015

Die Oracle OpenWorld startet in San Francisco. Wie nicht anders erwartet, beschäftigt sich der Oracle-Chef mehr als drei Viertel seiner Redezeit mit der Cloud, ohne besondere Neuigkeiten zu verkünden. Erst ganz zum Schluss brennt er dann ein Feuerwerk an Neuankündigungen im technischen Umfeld ab. Die Cloud ist und bleibt im Fokus des Oracle-Geschäfts. Erste Neuankündigungen sind zwei Cloud-Services, einer für das Manufacturing und

einer für E-Commerce. Bei Ellisons Rede entsteht der Eindruck, dass Oracle alles daran setzt, um führender Cloud-Anbieter zu werden. Entsprechend holt man gegenüber der Konkurrenz auf. Larry Ellison verkneift sich dabei auch keinen Seitenhieb gegenüber SAP und IBM, die ja bekanntlich beide noch keine eigene Cloud anbieten. Er sieht sich heute im Wettbewerb mit Workday und Salesforce. Für das kommende Jahr hat Oracle im Cloud-Geschäft einen Umsatz von 1,5 Milliarden Dollar eingeplant.



Larry Ellison bei seiner Eröffnungsk keynote

## 25. Oktober 2015

Die Keynote der JavaOne in San Francisco steht unter dem Motto „20 Jahre Java“ und greift technologische Meilensteine der Java-Zeitgeschichte auf. Ohne große Überraschungen werden die Roadmaps für Java SE, Java ME und Java EE aufgezeigt und planmäßig die gesteckten Technologie-Ziele eingehalten. Eines der größten Highlights kommt eher nur am Rande zur Sprache: Der WebLogic Server 12.2.1 steht als Java-EE-7-Implementierung ab sofort zum Download bereit. Georges Saab, Oracle Vice President für die Java Platform Group, startet den Reigen der Keynote-Speaker mit einem Blick auf Java SE. Beim OpenJDK gibt es ein Wachstum von 136 Prozent und rund ein Drittel der neuen OpenJDK-Projekte sind extern orientiert. Dieses Wachstum belegt, dass das Commitment von Oracle nicht nachlässt, sondern Früchte trägt. Im Anschluss daran stellt Shaun Smith, Oracle Product Manager, den neuen Java SE Cloud Service vor. Unter dem Motto „Build, Zip, Deploy“ ist dieser die einfachste Art, um Java-Anwendungen beispielsweise mit Tomcat gebündelt in der Cloud zu nutzen. Bei der Implementierung des Cloud-Service wird zur Virtualisierung Docker benutzt und zum Tooling Flight Recorder for Production Use angeboten. Der Services ist skalierbar und es gibt die Funktion „One Click Java Version Update“. Man muss sich also nicht um die Pflege der Plattform und die Security-Fixes kümmern. Als Highlight stellt Mark Reinhold, Chief Architect Java Platform Group, die wesentlichen Dinge im kommenden JDK 9 vor (Projekt „Jigsaw“). Es sind die Eliminierung von Classpath-Hell und die Modularisierung der Java-Plattform. Ein Vorteil dabei sind Java-Profile (Subsets der Java-Plattform) für unterschiedliche Device-Typen. Er zeigt dazu am Beispiel einer Hadoop-Anwendung wie sich der Classpath-Hell in der Realität manifestiert und wie die Einführung eines

Modul-Systems diese Probleme löst. Durch die Modularisierung der Java-Plattform lassen sich auch passende Profile (Subsets davon) für unterschiedliche Device-Typen definieren. Das nächste Highlight kommt von Brian Goetz, Chief Language Architect Java Platform Group, der andeutet, was nach Java 9 kommen soll: Das Valhalla Project bringt Specialized Generics sowie Value Types und im Panama Project wird es ein Foreign-Function-Interface, Data Layout Control sowie Array 2.0 geben. Eher unspektakulär verkündet Anil Gaur die größte Neuigkeit: Der WebLogic Server 12.2.1 steht ab sofort als Java-EE-7-Implementierung zum Download bereit. Einige Mitbewerber wie IBM und Red Hat haben ebenfalls Java-EE 7-Implementierungen herausgebracht. Zu Java EE 8 nennt er den Status und eine Roadmap: GlassFish 5 wird die Java-EE-8-Referenz-Implementierung sein. Erste Builds sind bereits zum Download bereit.

## 26. Oktober 2015

Oracle-Co-CEO Mark Hurd gibt sich auf der Oracle OpenWord in Bezug auf die Oracle-Strategie zum Thema „Cloud“ absolut überzeugt. Für ihn ist das die Technologie der Zukunft. Die Zahlen sind auf den ersten Blick beeindruckend: Sowohl bei PaaS als auch bei SaaS verzeichnet Oracle momentan eine Steigerung von mehr als 30 Prozent. Doch Mark Hurd reicht das nicht; in zwei Jahren soll die Steigerungsrate bei 100 Prozent liegen. Um dieses Ziel zu untermauern, gibt er fünf Prognosen für die nächsten zehn Jahre ab: 85 Prozent aller Applikationen laufen in der Cloud. Derzeit sind es 24 Prozent. Es wird nur noch zwei große Suite-Anbieter geben, die 80 Prozent des SaaS-Markts bedienen. 100 Prozent der Entwicklungs- und Test-Aktivitäten finden in der Cloud statt. Alle Unternehmensdaten sind in der Cloud gespeichert. Die Enterprise-Cloud ist die sicherste Umgebung für Daten. Der Markt wird zeigen, ob die Rechnung aufgeht.

## 27. Oktober 2015

Noch mehr Produkt-Neuheiten auf der Oracle OpenWord: Die Beta-Version des Oracle Database 12c Release 2 ist ab sofort verfügbar. Entscheidende neue Features sind das Kopieren, Aktualisieren und Verschieben einer Pluggable Database im laufenden Betrieb, die Steigerung der In-Memory-Performance und die bessere Unterstützung von Big Data Analytics. Auch für Entwickler gibt es eine interessante Neuerung: Die Version 12c von Oracle Forms & Reports verspricht mehr Features und höhere Performance. Nach den Vorstellungen von Oracle wandern in den kommenden Jahren zahlreiche IT-Leistungen in die Cloud. Entsprechend hoch ist die Zahl der Neuvorstellungen in Richtung „Cloud-Services“.



Das Freigelände auf der Oracle OpenWord

# „Die Datenbank mit der darin integrierten Business-Logik ist ein gutes Fundament für unsere Geschäftsprozesse ...“

Logistische Herausforderungen im Einzelhandel lassen sich oft nicht mit Standard-Software lösen. Stefan Kinnen, stellvertretender Vorstandsvorsitzender der DOAG, und Wolfgang Taschner, Chefredakteur der DOAG/SOUG News, sprachen darüber mit Dr.-Ing. Thomas Seeger, Leiter der Software-Entwicklung bei der Deichmann SE.



Dr.-Ing. Thomas Seeger (links) im Gespräch mit Stefan Kinnen

*In welchem Geschäftsfeld sind Sie tätig und was sind dabei die besonderen Herausforderungen?*

**Seeger:** Unser Business findet überwiegend im Einzelhandel statt. Die Ware wird weltweit eingekauft, in Südost-Asien konsolidiert und von dort in unsere rund dreitausend Filialen geliefert. Das Warenwirtschaftssystem umfasst neben der Verarbeitung der zahlreichen eingehenden Daten die Verteilung der Waren sowie alle Lösungen für die Zulieferkette, Auftragsschreibung, Kalkulation und die Verwaltung des Hochregal-Lagers. Auf diese Weise erreichen wir einen sehr schnellen Umsatz der Ware.

*Wie ist Ihre IT aufgebaut, um alle diese gestellten Aufgaben zu lösen?*

**Seeger:** Wir unterhalten seit mehr als zwanzig Jahren ein eigenentwickeltes Warenwirtschaftssystem auf Basis von Oracle Forms. Die Geschäftslogik ist in der Oracle-Datenbank-Version 11.2 unter IBM AIX abgebildet; das Warenwirtschaftssystem ist Multi-Mandanten-fähig und es ist mehrsprachig. Für alle anderen erforderlichen Aufgaben wie zum Beispiel Personalverwaltung oder Buchhaltung gibt es bei uns Standard-Anwendungen von SAP.

*Was sind die Besonderheiten in der IT?*

**Seeger:** Als inhabergeführtes Familienunternehmen leben wir viele Traditionen, die sich im Lauf der Jahrzehnte als gut erwiesen haben. So gibt es bei uns ein ganz eigenes, von dem kürzlich verstorbenen Dr. Heinz-Horst Deichmann eingeführtes spezielles System zur Artikelnummerierung, das man sonst auf dem Markt nirgendwo findet. Ein Standard-Software-Hersteller wäre nicht in der Lage, dieses abzubilden. Unser eigenentwickeltes Warenwirtschaftssystem ermöglicht sehr schnelle und kurze Entscheidungswege. Auch Änderungen an den Geschäftsprozessen lassen sich individuell und zielgerichtet umsetzen. Eine der großen Stärken unserer IT besteht darin, dass das Wissen um die Prozesse über Jahrzehnte gewachsen ist.

*Seit wann arbeiten Sie mit der Oracle-Datenbank?*

**Seeger:** Wir haben vor etwa zwanzig Jahren mit den ersten Forms-Anwendungen auf der Datenbank-Version 7 angefangen.

*Wie sind Ihre Erfahrungen mit Forms über diesen großen Zeitraum hinweg?*

**Seeger:** Forms ist extrem stabil und bietet einen Rahmen, in dem wir mit einem großen Entwicklungsteam parallel arbeiten können. Unsere Entwickler sind mitunter sehr überrascht über erstaunliche Verhaltensweisen der Software, doch das haben wir immer wieder in den Griff bekommen. Außerdem mussten wir uns viele Dinge, die wir benötigten, einfach selbst bauen. So ist über die Jahre ein Miteinander der Forms-Standardfunktionen und unserer Individual-Programmierung entstanden.

*Gibt es besondere Features, die Sie in Forms gerne gehabt hätten?*

**Seeger:** Uns fehlt insbesondere die effiziente Einbindung von Bildern in Bildschirmmasken. Außerdem gibt es keine Funktionen für das Drag-and-Drop und wir vermissen das Definieren von relativen Bildschirmpositionen. Hier mussten wir uns überall individuelle Lösungen erstellen.

### Deichmann SE

Die Deichmann SE mit Stammsitz in Essen (Deutschland) wurde im Jahr 1913 gegründet und ist zu 100 Prozent im Besitz der Gründerfamilie. Das Unternehmen ist Marktführer im europäischen Schuheinzelhandel und beschäftigt weltweit rund 36.000 Mitarbeiter. Unter dem Namen „Deichmann“ gibt es Filialen in Deutschland, Bosnien-Herzegowina, Bulgarien, Dänemark, Großbritannien, Italien, Kroatien, Litauen, Österreich, Polen, Portugal, Rumänien, Schweden, Serbien, der Slowakei, Slowenien, Spanien, Tschechien, der Türkei und Ungarn. Darüber hinaus ist die Gruppe vertreten in der Schweiz (Dosenbach/Ochsner/Ochsner Sport), in den Niederlanden (vanHaren), in den USA (Rack Room Shoes/Off Broadway) und in Deutschland mit Roland. Im Jahr 2014 hat Deichmann die ersten Läden in Russland eröffnet.



*Wie haben Sie vor einigen Jahren die Entscheidung von Oracle aufgenommen, Forms nicht mehr zu unterstützen und weiterzuentwickeln?*

**Seeger:** Im ersten Moment konnte ich das nicht glauben, da Forms sehr weit verbreitet ist. Die Entscheidung hat mich allerdings nicht verängstigt, da wir ja eine gut funktionierende und sehr stabile Forms-Landschaft haben, mit der wir auch weiterhin produktiv arbeiten können. Außerdem hilft uns die mögliche Einbindung von Java-Komponenten entscheidend weiter. Somit können wir unser Warenwirtschaftssystem auch ohne Unterstützung durch Oracle Forms weiterbetreiben. Ähnlich verhält es sich auch mit der Abkündigung von Reports. Hier können wir mit der Freeware-Lösung JasperReport die Oracle Reports ersetzen. Damit sind die Auswirkungen auf unser Unternehmen gering.

*Denken Sie dennoch über eine Migration auf eine neue Technologie nach?*

**Seeger:** Für mich und meine Kollegen sind die vermeintlichen Ersatzprodukte von Oracle wie Apex oder ADF keine ernsthafte Alternative. Sie sind weder vom Umfang noch von der Funktionalität her in der Lage, das zu leisten, was wir heute mit Forms haben. Sowohl für Forms als auch für Designer hätten wir uns eine bessere Migrations- und Zukunftsstrategie gewünscht.

*Haben Sie diese Problematik mit Oracle besprochen?*

**Seeger:** Es hat Gespräche mit Vertretern der Firma Oracle gegeben, allerdings bislang ohne eine befriedigende Perspektive für uns.

*Haben Sie es schon einmal mit Standard-Software versucht?*

**Seeger:** Wir sind schon oft darauf angesprochen worden. Doch aufgrund der Besonderheiten unserer Geschäftsprozesse wurde immer wieder entschieden, bei der Individual-Lösung zu bleiben.



*Können Sie immer noch gute Forms-Entwickler finden?*

**Seeger:** Das ist in der Tat sehr schwierig. Insbesondere junge Entwickler müssen erst besonders motiviert werden, um sich mit Forms zu beschäftigen. Wenn uns das gelingt, sind die Leute durchaus zufrieden mit ihrer Tätigkeit.

*Werden Sie Ihre Datenbank auf die Version 12c migrieren?*

**Seeger:** Wir sammeln bereits erste Erfahrungen mit der neuen Datenbank-Version, aber der Zeitpunkt, an dem wir unser Warenwirtschaftssystem migrieren, steht noch nicht fest.

*Wie zufrieden sind Sie mit dem Oracle-Support?*

**Seeger:** Unsere Datenbank-Administratoren sind im Großen und Ganzen mit dem Oracle-Support zufrieden. Sie wünschen sich allerdings, dass sich die Support-Mitarbeiter insbesondere bei umfänglicheren oder komplexeren Anfragen früher auf unser System aufschalten würden, um die Probleme schneller und besser zu verstehen.

*Wie beurteilen Sie die Lizenzpolitik von Oracle?*

**Seeger:** Inzwischen haben wir an dieser Stelle unseren Frieden gefunden und mit Oracle ein „Unlimited License Agreement“ geschlossen. Somit können wir auch unsere Test-Instanzen aufbauen und damit arbeiten. Zuvor gab es mit einer User-basierenden Lizenzmetrik deutlich negativere Erfahrungen, weil selbst für zeitlich befristete Testprojekte seitens Oracle Lizenzforderungen erhoben wurden, die wir nicht akzeptieren konnten. In einem Fall sind wir dann konkret auf eine MySQL-Datenbank ausgewichen.

*Wie beurteilen Sie die Zukauf-Strategie von Oracle?*

**Seeger:** Ich bin grundsätzlich der Meinung: „Schuster bleib bei deinen Leisten“. Die Kernkompetenz von Oracle liegt aus meiner Sicht in der Datenbank. Wenn neue Firmen aufgekauft werden, um deren Produkte gebündelt an den Mann zu bringen, halte ich das für eine Mogelpackung. Aus meiner Erfahrung vergehen immer einige Jahre, bis die Produkte übernommener Unternehmen vollständig integriert sind.

*Können Sie sich vorstellen, ein Komplettsystem von der Hardware bis zu den Applikationen von einem einzigen Hersteller wie Oracle einzusetzen, Stichwort „Engineered Systems“?*

**Seeger:** Nein, wir haben die Strategie, unser System aus Komponenten unterschiedlicher Hersteller zusammenzustellen.

*Wie denken Sie über die Cloud?*

**Seeger:** Die Cloud ist derzeit für uns kein Thema. Wir werden auch zukünftig unsere Firmendaten auf den eigenen Systemen halten.

*Was ist Ihre Zukunftsvision von der IT?*

**Seeger:** Wir fühlen uns mit dem, was wir hier haben, gut aufgehoben. Die Datenbank mit der darin integrierten Business-Logik ist ein gutes Fundament für unsere Geschäftsprozesse. Selbst wenn eines Tages Oracle Forms nicht mehr zur Verfügung stehen würde, könnten wir mit einem modernen Framework eine Oberfläche bauen, um die Prozesse so, wie wir sie gewohnt sind, abbilden zu können.

*Was erwarten Sie dabei von einem großen IT-Unternehmen wie Oracle?*

**Seeger:** Ich wünsche mir eine moderne Oberflächen-Landschaft, die von der Funktion und vom Umfang her Oracle Forms wirklich ablösen könnte. Sie sollte deutlich offener als Oracle Forms sein, damit man sie leichter mit eigenen Modulen ergänzen kann – auch aus lizenztechnischer Sicht.

*Wie sehen Sie den Stellenwert der DOAG?*

**Seeger:** Die DOAG ist sehr wertvoll, um die Interessen der Anwender gegenüber dem Hersteller zu bündeln. Zudem bieten die zahlreichen Veranstaltungen, insbesondere die Jahreskonferenz, eine hervorragende Möglichkeit zum Erfahrungsaustausch.



**Zur Person: Dr.-Ing. Thomas Seeger**

Dr.-Ing. Thomas Seeger studierte Elektrotechnik an der RWTH Aachen. Nach dem Studium erstellte er als wissenschaftlicher Mitarbeiter der Bergischen Universität Wuppertal seine Promotion über wirtschaftlich optimale Betriebsführung von Energie-Systemen. Anschließend wechselte er zum IT-Beratungsunternehmen SOPTIM AG, wo er acht Jahre lang als Entwickler und Projektleiter in verschiedenen Kundenprojekten eingesetzt war. Im Jahr 1999 begann er seine Tätigkeit bei der Firma Deichmann und baute die neu gegründete Abteilung Software-Entwicklung auf, die das Warenwirtschaftssystem für alle europäischen Firmen der Deichmann-Gruppe entwickelt und betreut.

# Chancen und Grenzen: Warum gerade jetzt Big Data

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

Wer sich heute mit Big Data beschäftigt, sollte sich von all dem Hype freimachen, der rund um dieses Thema gemacht wird. Dann sollte es gelingen, den passenden (nicht nur technischen) Einstieg zu finden, um die Potenziale von Big Data nicht zu verpassen.

Der erste Aspekt ist schnell abgearbeitet: Der Begriff „Big Data“ wird heute oft so schwammig verwendet, dass alles, was mit Software zu lösen ist, als „Big Data Use Case“ gefeiert wird. Produzieren wir nicht schon seit vielen Jahren Software und lösen damit Business-Probleme? Halten wir uns mit diesem Missverständnis nicht länger auf und wenden uns dem wesentlich spannenderen zweiten Aspekt zu. Viele der mit Big Data diskutierten neuen Lösungen enthalten im Kern schon recht alte und bekannte Ideen. Doch um die Dimensionen und das Potenzial von Big-Data-Lösungen schätzen und begreifen zu lernen, sollten wir uns zunächst mit den geänderten gesellschaftlichen und sozialen Rahmenbedingungen und – komplementär dazu – mit den heute existierenden sogenannten „Enabler-Technologien“ befassen. Lebensgestaltung und technischer Fortschritt bedingen sich gegenseitig. Aus dieser Wechselwirkung entstehen immer wieder auch neue Geschäftsideen, ganze Berufszweige, neue Unternehmen oder gar Branchen.

Heute hat sich das Internet zum Kommunikationsmedium schlechthin entwickelt. Es ist es eine riesige Quelle von Fakten und Informationen über Beziehungen, über Trends oder gesellschaftliche und politische Bewegungen geworden. Mehr noch: Was für das Internet Text, Sprache oder Bilder sind, das sind für die Dinge und Gebrauchsgegenstände um uns herum Abermillionen Sensoren und Messstationen. Betroffen sind fast alle Produkte, die Umwelt, der Verkehr, aber auch Produktions- und Handelsprozesse. Das

Gemeinsame ist die Möglichkeit, alles, was passiert, digital zu analysieren, zu bewerten, neue Schlüsse daraus zu ziehen und entsprechend zu handeln.

Das Einbauen eines Sensors bedeutet allerdings lange noch nicht, dass Daten auch ausgewertet werden. Während es meist noch an Ideen dazu fehlt, ist die „Enabler“-Technik zum Auswerten heute im Gegensatz zur Vergangenheit da. Da sind Hadoop, In-Memory, extrem schnelle Hardware und leistungsfähige Entwicklungs- und Abfragesprachen wie Scala, Python oder R. Sensor- und Internet-Technologie demonstrieren exemplarisch die Entstehung des Phänomens der „Big-Data-Daten“. Die Datenmenge ist jedoch nur ein Aspekt. Hinzu kommen die spontane, wenig planbare, fast beiläufige Entstehung der Daten sowie die Format-neutrale Speicherung beliebiger Datenarten, in der Big-Data-Diskussion als die drei „V“s Data Velocity, Data Variety und Data Volume bekannt. Skalierungsverfahren wie Hadoop oder auch In-Memory ermöglichen die Speicherung auch von spontan auftretenden riesigen Datenmengen und auf diesen Daten eine bis dahin nicht geahnte Verarbeitungs- und Analysegeschwindigkeit.

Wir haben also nicht nur zu fast allem, was uns umgibt, Daten, wir können diese Daten auch nutz- und gewinnbringend einsetzen. Wir müssen es auch tun. An diesem Punkt stehen heute viele Unternehmen, denn es gilt Ideen zu finden, um diese neuen Chancen wahrzunehmen und Potenziale auszuschöpfen. Big Data zu nutzen, ist demnach eher die besondere Aufgabe des Erkennens des Potenzials

der neuen technischen Möglichkeiten und die Umsetzung in neue Geschäftsideen. Dazu ist technisches Verständnis gepaart mit Business-Fokus nötig.

## Neue Daten, neue Chancen

Daten sind die besonderen Mittel, aus denen heraus viele neue Ideen rund um Big Data entstehen. Die technische Innovation der letzten Jahre hat den Horizont der nutzbaren Daten erweitert. Wir können diese neuen Daten entsprechend ihrer Verwendung in mehrere Datenarten untergliedern. Die Quelle für diese Kategorisierung in Datenarten stammt aus einer Sammlung mit generischen Big Data Use Cases zu unterschiedlichen Industrien, die man in einem separaten Whitepaper nachlesen kann (siehe [„http://oracledwh.de/downloads/AutoIndex-2.2.4/index.php?dir=Artikel/&file=Big%20Data%20im%20Kontext%20von%20Data%20Warehouse.pdf“](http://oracledwh.de/downloads/AutoIndex-2.2.4/index.php?dir=Artikel/&file=Big%20Data%20im%20Kontext%20von%20Data%20Warehouse.pdf)):

- Daten wie betriebswirtschaftliche Ergebnisse, Verbrauchsdaten, Umsätze, Gewinne, abgesetzte Waren, Menge der Kunden etc. sind klassische Informationen, wie sie in Data-Warehouse-Systemen schon lange vorgehalten und gemessen werden. Das sind überschaubare Datenmengen. Auch in Big-Data-Zeiten nutzen wir diese Daten als Grundlage für viele Entscheidungen. Diese Daten verschwinden nicht, sie werden um die nachfolgenden Datenarten ergänzt.
- Die nächste Gruppe sind Daten, die im Verlauf von Abläufen oder Geschäftsprozessen auftreten, also Log-Informationen oder Sensordaten aus Maschinen.

Es sind Daten mit einem unmittelbaren Zusammenhang zu den verursachenden Aktionen. Man kann direkt Rückschlüsse auf den Verlauf der jeweiligen Aktion schließen. Solche Daten kommen der Vorstellung von Massendaten im Big-Data-Zeitalter sehr nahe.

- Im Gegensatz dazu steht die Gruppe der mittelbaren Daten. Das sind beispielsweise Wetter- und Umweltdaten (Klima) oder Verkehrsmessungen. Solche Daten können einen gewissen Einfluss auf Geschäftsprozesse haben. Vorstellbar ist die Optimierung von Transportrouten per LKW, Bahn oder Schiff. Der Zusammenhang ist jedoch nicht absolut und nur beschränkt vorhanden, aber er ist messbar. Diese Daten sind nicht neu. Sie wurden bislang nur noch zu wenig in Verbindung mit Geschäfts- oder Planungsszenarien gebracht.
- Eine weitere Gruppe sind sogenannte „weiche Daten“ wie Meinungsäußerungen in den sozialen Medien, aber auch Presseartikel und Nachrichten. Hier lassen sich über „Sentiments“, also Textanalysen, Einstellungen und Vorlieben erkennen. Die Nutzung dieser Datenart ist recht neu.
- Eine wichtige neue Gruppe sind die Dokumenten-orientierten Daten, wobei der Begriff „Dokument“ als Platzhalter für alles in Textform Geschriebene, für Bilder, Filme und Tonaufnahmen steht. Wissen über Zusammenhänge entsteht hier durch Vergleiche von vielen ähnlich angelegten Datenobjekten, es ist die Suche nach unbekanntem Auffälligkeiten. Hier hat die Technik in der jüngsten Zeit sehr viele Fortschritte gemacht. Während es heute schon große, mit elektronischen Mitteln verwaltete Medien-Archive gibt, fehlt es immer noch an Anwendungen, die diese Datenart etwa mit den Daten unter Punkt 1 verknüpfen.
- Eine Sonderrolle spielen Bewegungsdaten. Hier ist der Orts- und Zeitbezug der entscheidende Aspekt. Wann treten Ereignisse wo auf? Gibt es Häufungen? Gibt es räumliche oder zeitbezogene Verbindungen, Abhängigkeiten etc.?
- Es fehlt noch die Gruppe der Umfeld-Informationen. Das sind Daten über Wettbewerber oder eine allgemeine Marktlage. Hier handelt es sich ebenfalls um klassische Daten, die immer schon genutzt wurden. Auf solche Informationen

kann man heute durch neuere Techniken wesentlich leichter zugreifen, etwa durch Web-Zugänge auf freie öffentliche Datenquellen oder durch permanentes Monitoring von Web-Auftritten oder Nachrichtenportalen.

Einzelne Datenarten mögen schon seit längerer Zeit in den Unternehmen zur Verfügung stehen. Die mögliche Verknüpfung der Daten untereinander, das nahezu unbegrenzte Fassungsvermögen der Datenspeicher und die Anwendung automatisierter Algorithmen ist das Neue.

### **In-Memory – heute eine der wichtigsten „Enabler“-Techniken für Big Data**

Am Beispiel der In-Memory-Verarbeitung erkennt man gut, in welche Richtung die technische Innovation voranschreitet. Viele der Big-Data-Anwendungsfälle produzieren aus ihrer Natur heraus permanent Einzel-Informationen, die sich in ihrer Summe kaum noch ökonomisch sinnvoll speichern und verwalten lassen. Die Beispiele sind bekannt, ob Sensor- oder Kommunikationsdaten, Bewegungsprofile von vielen Millionen Personen und Fahrzeugen oder Verbrauchsdaten von Energie. Nehmen wir nur Wetterdaten hinzu und schon potenzieren sich die Datenmengen.

Die Hadoop-Technologie mit dem HDFS-Dateisystem ermöglicht dieses massenhafte Speichern von Daten nach heutigen Erfahrungen unbegrenzt. Man nutzt einzelne Knotenrechner (CPU, Memory und Plattenspeicher) zum Speichern von relativ großen Datenblöcken. Wenn die Kapazität der Rechner ausgereizt ist, schaltet man neue Rechner (Knoten) dazu. Um zu wissen, welche Daten (Blöcke) auf welchen Knoten liegen, nutzt man wiederum weitere Rechner. Es entsteht ein riesiges Netzwerk aus Datenknoten und Verwaltungsrechnern. Es ist dieses Konzept, das Skalierung garantiert, während die bislang übliche Aufteilung in SAN und separaten Serverpark beim Kostenvergleich hinterherhinkt.

Das Speichern von Massendaten ist jedoch nur ein erster Schritt. Die Verarbeitung, das Lesen und Analysieren dieser Daten ist der eigentlich wichtige zweite Schritt. Hier gab es gewaltige technische Fortschritte in den letzten Jahren und In-Memory spielt dabei eine zentrale Rolle.

Das ursprüngliche sogenannte „MapReduce-Verfahren“, bei dem Java-Programmcode parallel auf die vielen Datenknoten geschickt wurde („Mapper“-Phase), um die Einzelergebnisse danach über Konsolidierungsprozesse wieder zusammenzufassen („Reducer“-Phase), gelten mittlerweile schon als überholt. Die aktuell viel diskutierte Spark-Technik übernimmt das Skalierungs- und Verteilungsprinzip der vielen Rechner von MapReduce. Spark nutzt dagegen konsequent die Hauptspeicher der jeweiligen Datenknoten und verteilt „In-Memory“ sogenannte „RDD-Objekte“ (Resilient Distributed Dataset) auf die Rechner. Die parallelisierte Verarbeitung schreibt Zwischenergebnisse nicht wieder auf die Datenknoten zurück, wie bei MapReduce, sondern belässt sie als RDD-Bestandteil im Hauptspeicher und reicht sie in Pipeline-Manier an nachfolgende Verarbeitungsschritte weiter. Das Ergebnis sind Performance-Gewinne um Faktoren im Hunderter-Bereich.

### **Hohe Rechengeschwindigkeit auf riesigen Datenmengen**

Ein eindrucksvolles Beispiel für diese extreme Performance-Verbesserung ist die Umstellung der beiden Oracle-R-Mining-Algorithmen „Generalisierte Lineare Modelle“ (GLM) und „Neural Networks“ (MLP) von MapReduce auf Spark. Solche Analysen erfordern hohe Rechenleistungen. In einem Testlauf auf 100 Millionen Sätze konnte man GLM um den Faktor 114 beschleunigen und eine Neural-Network-Analyse auf eine Milliarde Sätze um Faktor 217. Das macht das Ganze natürlich sehr attraktiv, zumal Data Mining durch seinen explorativen Charakter für Big-Data-Analysen genau passt.

Das zweite Beispiel für eine hochperformante In-Memory-Verarbeitung ist das Graphen-Modell, das sich für die Analyse von Zusammenhängen und Beziehungen ideal eignet. Der Anwendungsfall sind Analysen über die Kommunikationsbeziehungen in sozialen Netzwerken. Ein Graph ist, verkürzt gesagt, die Beschreibung der Beziehungen (Kanten) von zwei Objekten (Netzknoten). Einen solchen Graphensatz nennt man Triple. Die drei Aussagen „Meier kennt Müller“, „Müller kennt Schmidt“, „Schmidt kennt Schuster“ ergeben danach ein Graphen-Modell mit der Information „Meier kennt indirekt Schuster“. In sogenannten

```

1: library(twitteR) # Bibliothek für Twitterzugriffe
2: library(tm) # Bibliothek für Textmining
3: tweets <- userTimeline("source", n = nnn) # Lesen von nnn Anzahl Tweets
4: tweets.df <- twListToDF(tweets) # Tweets zum Bearbeiten vorbereiten
5: myCorpus <- Corpus(VectorSource(tweets.df$text)) # In-Memory Objekt "myCorpus" erstellen
6: myCorpus <- tm_map(myCorpus, tolower) # alle Buchstaben klein machen
7: myCorpus <- tm_map(myCorpus, removePunctuation) # alle Punkte, Striche etc. entfernen
8: myCorpus <- tm_map(myCorpus, removeNumbers) # alle Ziffern entfernen
9: myCorpus <- tm_map(myCorpus, stripWhitespace) # Alle Leerzeichen entfernen
10: myCorpus <- tm_map(myCorpus, stopwords("english")) # Entfernen der Stoppwörter, (ich du.)
11: library(SnowballC) # Bibliothek für Grundwort-Bildung
12: myCorpus <- tm_map(myCorpus, stemDocument) # Rückführung aller Wörter auf Grundform
13: tdm <- TermDocumentMatrix(myCorpus) # Matrix mit der Verwendung der Wörter in den Tweets, auf diesem Objekt
wird in der Folge analysiert
14: library(graph) # zusätzliche Netz-Graphik-Bibliothek
15: library(Rgraphviz) # zusätzliche Netz-Graphik-Bibliothek
16: term.freq <- subset(term.freq, term.freq >= 4)
17: plot(tdm, term = freq.terms, corThreshold = 0.05, weighting = T) #Netzgraphik erstellen

```

Listing 1

„Property Graph Data Models“ lassen sich die Beziehungen zusätzlich noch gewichten beziehungsweise attributieren. Leicht vorstellbar, dass solche Netzmodelle gewaltige Ausmaße annehmen können, wenn etwa das Beziehungsgeflecht von Twitter-Nutzern beschrieben wird. Solche Modelle lassen sich in der Oracle-Datenbank, in der Oracle-NoSQL-Datenbank und über HBase in HDFS realisieren.

Oracle liefert heute 35 Algorithmen zur Analyse dieser Netzwerke, etwa zum Herausfinden von Hotspots in Netzen, Finden von Twitters Who-to-follow, Clustern von Netzwerken, Finden von Pfaden und zu vielem mehr. Alle diese Algorithmen nutzen in besonderem Maß die Big-Data-Cluster-Infrastruktur aus, sie laufen auf vielen Knoten parallel und vor allem: Sie laufen ausnahmslos im Hauptspeicher. Um auch besonders große Netze im Hauptspeicher abarbeiten zu können, zerlegt Oracle große Netze in Teilnetze und lädt diese komplett in den Hauptspeicher.

Das erklärt, warum die jüngste Oracle Big Data Appliance Maschine (X5-2) bis zu 13 TB Hauptspeicher pro Rack mitbringt, der von 648 Cores gleichzeitig bearbeitet werden kann. Genau das ist die jüngste Erfahrung der Entwickler der Big Data Appliance: Hauptspeicher und die Anzahl der Rechenkerne werden für Big-Data-Anwendungen immer wichtiger. So ist hohe parallele Rechenleistung durch viele Cores möglich und die zu bearbeitenden Massendaten liegen als Datenobjekt im Hauptspeicher (In-Memory Objekte).

### Kompaktes Prototyping, kompaktes Analysieren und schnelle Ergebnisse

Die Spark-Technologie liefert noch eine zweite spannende Entwicklung: das kompakte und funktionsorientierte Arbeiten. Die ersten MapReduce-Projekte waren meist von der Java-Programmierung geprägt und schreckten Unternehmen damit eher ab. Zu befürchten waren „never-ending“, technisch ausgerichtete Programmierprojekte mit kaum ersetzbaren Experten als Projektmitarbeiter. An schnelles, Business-orientiertes Prototyping oder gar Sandboxing für Fachmitarbeiter war nicht zu denken. Mit neuen Programmierumgebungen und -sprachen wie Scala, Python oder auch R muss man heute zwar immer noch programmieren, die angebotenen Sprachen-Konstrukte, Bibliotheken und darin enthaltenen Funktionen bewegen sich jedoch auf einem höheren Level, es sind speziell vorbereitete Aufrufe. In diesen Sprachen schickt man High-Level-Funktionen gegen In-Memory-Datenobjekte. Die Skript-Erstellung ist effizient und sehr flexibel.

Ein einfaches R-Text-Mining-Beispiel zur Sprach-Analyse von Twitter-Feeds demonstriert diese typisch kompakte Arbeitsweise (siehe Listing 1). Man nutzt wenige knappe Funktionsaufrufe auf ein Datenobjekt im Hauptspeicher. Das Datenobjekt („myCorpus“) ist das Ensemble aller Tweets, die als Einheit dem Mining-Vorgang unterzogen werden. Das Skript sucht diejenigen Wörter, die gemeinsam in den Tweets genutzt vorkommen. Tre-

ten Begriffe besonders häufig mit anderen Begriffen gemeinsam auf, wird dies erkannt und als „besonders intensive Beziehung“ vermerkt. Die Grafik in *Abbildung 1* zeigt dies als fette Beziehungsstriche. Aus der Analyse lassen sich letztlich Themen-Komplexe ableiten, also Diskussionsaspekte in der Twitter-Gruppe.

Das Ergebnis ist eine netzartige Darstellung über gemeinsam in den Feeds genutzte Wörter. Bei großen Netzen erkennt man über „Hotspots“ besondere Themen-Komplexe. Über eine anschließende LDA-Topic-Model-Analyse lassen sich sogar fachliche Diskussions-Schwerpunkte identifizieren.

Ein solches Ergebnis ist in der Kürze wohl kaum mit anderen Mitteln erreichbar. Um die Bedeutung des Beispiels voll zu erfassen, sollte man sich die Datenmenge von Hunderten Millionen von Tweets vorstellen, die im Verlauf eines Tages im englischsprachigen Raum zu finden sind. Das „myCorpus“-Objekt, in dem alle Tweet-Texte vorgehalten sind, liegt komplett im Hauptspeicher. Eine ideale Anwendung für einen Hadoop-Cluster mit riesigem Hauptspeicher und vielen parallel arbeitenden Rechenkernen in den Cluster-Knoten.

### JSON und NoSQL – pragmatischer geht es kaum

Das Text-Mining-Beispiel führt zu einem weiteren wichtigen Big-Data-Aspekt. Die analysierten Tweets liegen zwar im Hauptspeicher, für weitere künftige Analysen muss man sie jedoch dauerhaft speichern. An diesen Massendaten interessieren letzt-

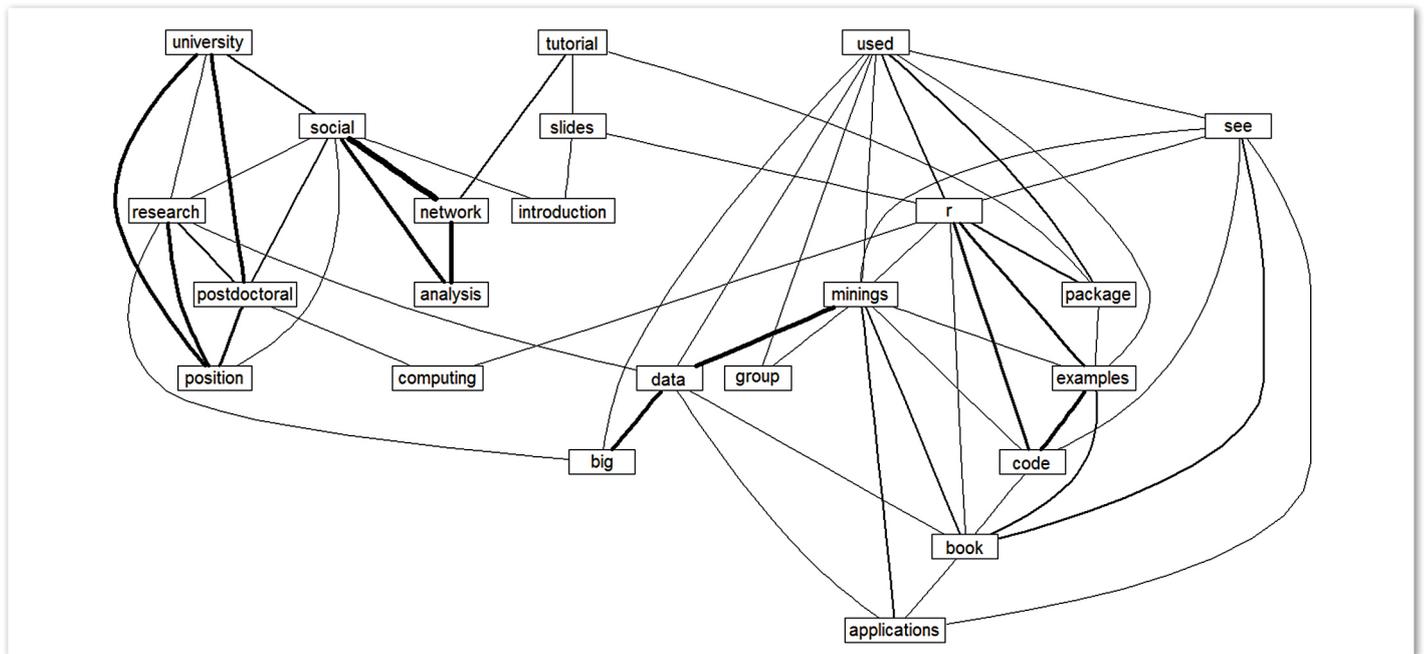


Abbildung 1: Netzgeflecht zur Häufigkeit gemeinsam genutzter Wortpaare

lich nur der Text und wenige Metadaten-Merkmale. Genau für diesen sehr einfachen Zweck eignet sich das zunehmend häufiger genutzte JSON-Format. Jeden Tweet kann man auch als solchen JSON-Satz ablegen (siehe Listing 2).

Das gleiche Skript kann die Tweets auch aus der Oracle-NoSQL-Datenbank (Key Value Store) lesen. Man müsste nur die Zeilen 1 und 3 ändern. Die einzige wichtige Funktion, die die NoSQL-Datenbank zu übernehmen hat, ist schnell und einfach und zig Milliarden Mal einen solchen Twitter-Tweet-Satz abzulegen. Das Strukturieren und Analysieren übernimmt das R-Skript, also die Anwendung. Das hier gezeigte Beispiel funktioniert mit vielen anderen Anwendungsfällen genauso, bei Log-Daten, Sensordaten oder Geopositionen.

### HDFS, NoSQL-Datenbank, RDBMS

Damit sind wir bei den Empfehlungen für die Wahl der Speichertechnik. Über die innere Struktur der oben genannten Tweets muss sich das Speichermedium keine Gedanken machen. Es reicht zu wissen, dass ein Tweet ein Datensatz ist, ideal für eine NoSQL-Datenbank. Diese ist kostengünstiger im Betrieb und bei Schreib- und Lese-Operationen sehr performant. RDBMS-Systeme benötigen wir dagegen für sensible und schützenswerte Daten und, bezogen auf Data-Warehouse-Systeme, für komplexere Abfragen, bei denen das Datenhaltungssystem bei der Struk-

```
{ "id": 347272731663945729,
  "retweetCount": "3",
  "Text": "Introduction to R for Data Mining, a one-hour video by Revolution Analytics http://t.co/fo5KVU11k9",
  "longitude": "9.993682",
  "latitude": "53.551085",
  "statusSource": "web",
  "created": "2015-10-18 21:11:07" }
```

Listing 2

turierung der Daten mithilft. Sie entwickeln ihre Stärken in großen Multi-User-Systemen mit vielen gleichzeitigen Lese- und Schreib-Operationen und auch bei heterogenen Zugriffen durch unterschiedliche Werkzeuge, wie sie im Business-Intelligence-Umfeld üblich sind. SQL steht im Vordergrund.

Reine HDFS-Hadoop-Systeme sind kostengünstige Massen-Schreibsysteme. Große Datenmengen lassen sich in kurzer Zeit in einem günstigen Cluster speichern. Security, parallele Nutzung der Daten und performante Leseoperationen sind weniger gut entwickelt (siehe Abbildung 2).

### Visuelles Analysieren oder muss man doch programmieren?

Programmieren muss nicht unbedingt sein. Von Oracle gibt es heute mit Big Data Discovery ein interaktiv bedienbares Werkzeug, mit dem man Hadoop-Daten auch explorativ bearbeiten kann. Zunächst

verschafft man sich durch „Profiling“-artige Stichproben eine erste Übersicht zu dem angebotenen Datenmaterial. Fehlen zur weiteren Analyse wichtige Merkmale in den Daten, etwa Schlüsselwerte für eine Verknüpfung mit anderen Datenbeständen, so kann man diese interaktiv mit diversen Wizards (ähnlich wie klassisches ETL) ergänzen. Am Ende stehen Grafiken zur visuellen Darstellung von Zusammenhängen.

Allerdings wird auch ein solches Werkzeug immer nur ein Teil in einer größeren Werkzeugsammlung für die Analyse sein. Man muss mit Sprachen wie R direkt auf den Daten arbeiten können, man benötigt Batch-Skripte zum Automatisieren von Arbeitsschritten, man braucht Third-Party-Bibliotheken für Zugriffe auf Remote-Datenbestände etc. Das Text-Mining-Beispiel macht deutlich, wie oft zusätzliche Bibliotheken für spezielle Analyse-Methoden

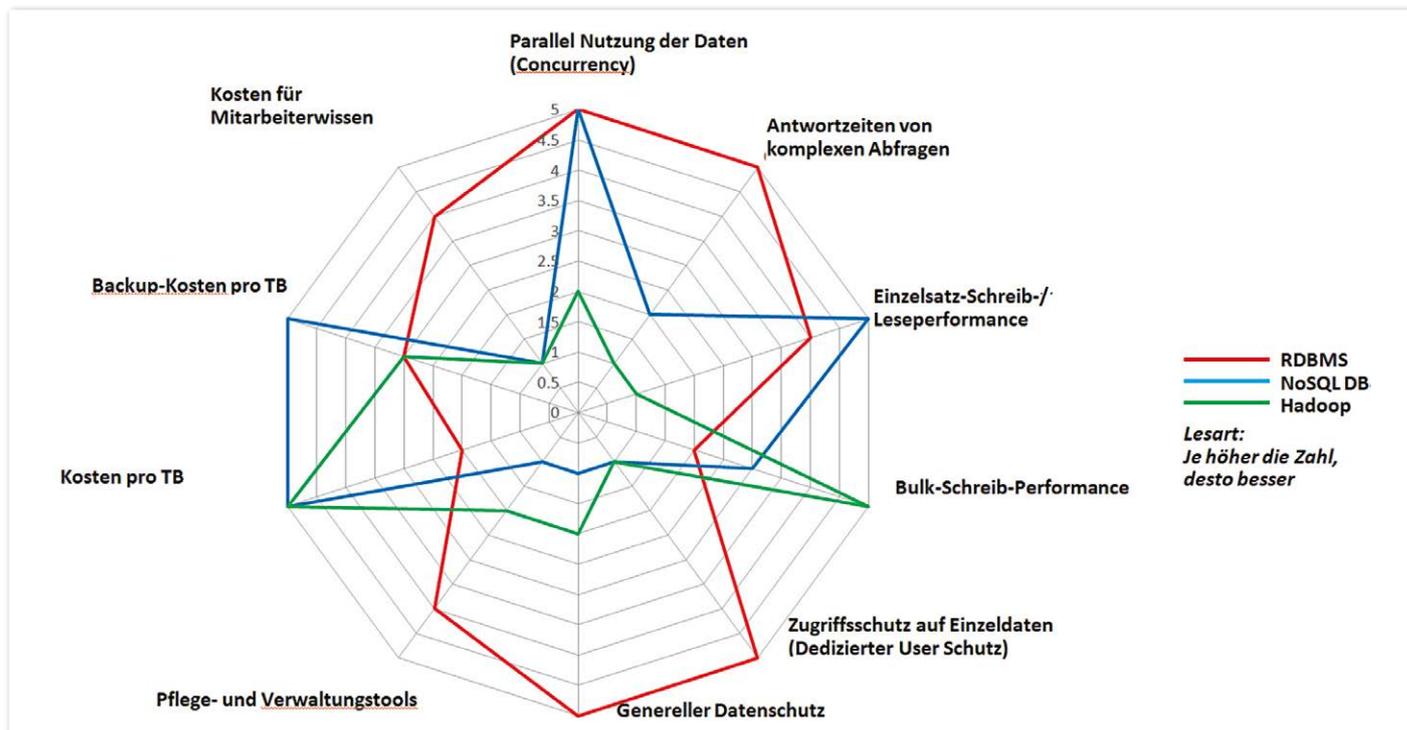


Abbildung 2: Speichersysteme im Vergleich

in den Arbeitsprozess einzubinden sind. Notwendig sind auch Sprachen wie Scala für systemnahe Operationen. Die Datenfachkraft („Data Scientist“) der Zukunft muss sich mit allen Aspekten rund um die Daten beschäftigen. Das sind betriebswirtschaftlicher Kontext, Beziehungen in den Daten, Struktur der Daten, aber auch physikalische Eigenschaften und Art der Speicherung sowie performantes Analysieren. Sich nur auf Business-Aspekte zurückzuziehen und Technik als Blackbox zu betrachten, reicht zur Erfüllung dieser Aufgabenstellungen nicht mehr aus.

**Kritische Aspekte und was nebenbei so alles machbar ist**

Neben all den technischen Möglichkeiten gibt es natürlich auch eine kritische Sicht. Das oben dargestellte Text-Mining-Skript kann man mit wenigen Änderungen auch zum Monitoring von Callcenter-Mitarbeitern in Kundengesprächen nutzen. Gesprächs-Mitschnitte, also Tonaufnahmen, überführt man in geschriebenen Text. Skripte bereinigen sie, wie oben gesehen und vergleichen den Sprachtext mit einer Liste positiver und negativer Wörter. Hinzu kommen Werte zur Stimmhöhe im Verlauf der gesprochenen Sätze und zu weiteren Stimm- und Sprach-Merkmalen (Satz-Analyse im Gegensatz zur Wort-Ana-

lyse). Das Ergebnis ist eine Gesprächsbewertung über ein positiv beziehungsweise negativ geführtes Kundengespräch – ein idealer Input für Mitarbeiter-Bewertungsgespräche. Zukunftsmusik? Leider nein. Wir können davon ausgehen, dass dieses Szenario heute schon Praxis ist, wenn nicht offen, dann zumindest versteckt.

Das Szenario ist nur ein Beispiel aus der Callcenter-Branche. Diese Art des „Monitorings“ kann uns durch Big-Data-Technologie künftig in fast jeder Lebenslage umgeben. Beim Callcenter könnte man noch (etwas hochnäsig) argumentieren: „Warum hat der Mitarbeiter diesen Beruf gewählt?“. Doch die Biodaten-Erfassung der Freizeitsportler ist das freiwillige Nutzen von Sensortechnik. Auch der permanente „Gesunde-Lebensführungs-Check“ durch elektronische Bio-Tracker der Krankenkassen ist freiwillig. Er wird irgendwann durch einen niedrigeren Kassenbeitrag honoriert. Das ist natürlich immer noch freiwillig, aber jeder wird es tun, weil er sparen möchte.

Diesen Anwendungsfällen kann man immer noch einen positiven Aspekt abgewinnen. Kritischer wird es allerdings, wenn Medienunternehmen die „Sentiments“ der Bürger messen und ihnen nur angepasste Unterhaltung und wohlgefällige Informationen als Bild und

Print anbieten. Unangenehme Wahrheiten, Berichte über Missstände, Aufforderungen zum Nachdenken und Verändern wird es in solchen Medien immer weniger geben.

Steht die Technik zur Verfügung, dann wird sie auch eingesetzt, wenn jemand einen ökonomischen Nutzen für sich darin erkennt. Man sollte sich der gesellschaftlichen Dimension von Big Data bewusst sein.



Alfred Schlaucher  
alfred.schlaucher@oracle.com

# Oracle Data Warehouses und Big Data im Zusammenspiel

Peter Welker, Trivadis GmbH

Immer häufiger stellt sich die Frage, wie man Daten aus der Oracle-Datenbank und solche aus Big-Data-Plattformen wie Hadoop und NoSQL-Datenbanken verschieben oder gemeinsam auswerten kann. Der Artikel erklärt zahlreiche Möglichkeiten, (nicht) immer nur aus der Oracle-Datenbank-Perspektive: Angefangen mit den Oracle Big Data Connectors über den Oracle Data Integrator, Sqoop und Oracle Heterogenous Gateway bis hin zu Big Data SQL oder Golden Gate gibt es zahlreiche Lösungen. Aber Vorsicht, die Tücke steckt – wie immer – im Detail.

Data Warehouses (DWH) sind vermutlich das prägnanteste Beispiel für Datenbanken, die Daten zahlreicher Vorkomplexe in eine zentrale Datenbasis integrieren. Dafür hat sich in den letzten dreißig Jahren eine Reihe von Methoden und Techniken etabliert, die man heute in den meisten DWHs antrifft. Oft fließen die Daten zentral in eine DWH-Core-Datenbank, auf der sie – üblicherweise einmal täglich – einheitlich aufbereitet abgelegt und über lange Zeit historisch gehalten werden. Darauf setzen dann diverse, Abfrage-optimierte Data Marts auf, die jeweils anforderungs-

gerechte Abfrage-Schnittstellen auf diese Daten zur Verfügung stellen.

In den letzten Jahren häufen sich jedoch die Anforderungen an nichttabellarische Daten, kürzere Aufbereitungszeiten, Self-Service-Anforderungen, die DWH-Daten auch mit fremden Daten gemeinsam nutzen wollen, oder an Datenmengen und Durchsätze, die vor allem zu enormen Datenbank-Lizenz- und -Supportkosten führen, wenn man sie performant in das DWH integrieren möchte. Hier kommen neue Technologien wie Hadoop oder NoSQL-Datenbanken ins Spiel, die es gilt mit den

bestehenden, relationalen DWHs zu vereinen. Die Frage ist: „Wie soll das funktionieren?“ Wie in den letzten Jahren zu sehen, werden diese Technologien die relationalen und multidimensionalen Lösungen nicht einfach ersetzen können. Sie werden aber auch nicht wieder verschwinden. Vielmehr ist das Zusammenspiel der verschiedenen Welten ins Zentrum der Hersteller- und Dienstleisterbestrebungen gerückt. „Big Data Management System“ heißt das Zauberwort dafür bei Oracle.

Im aktuellen „Statement of Direction“ für diese Big-Data-Management-Systeme (siehe

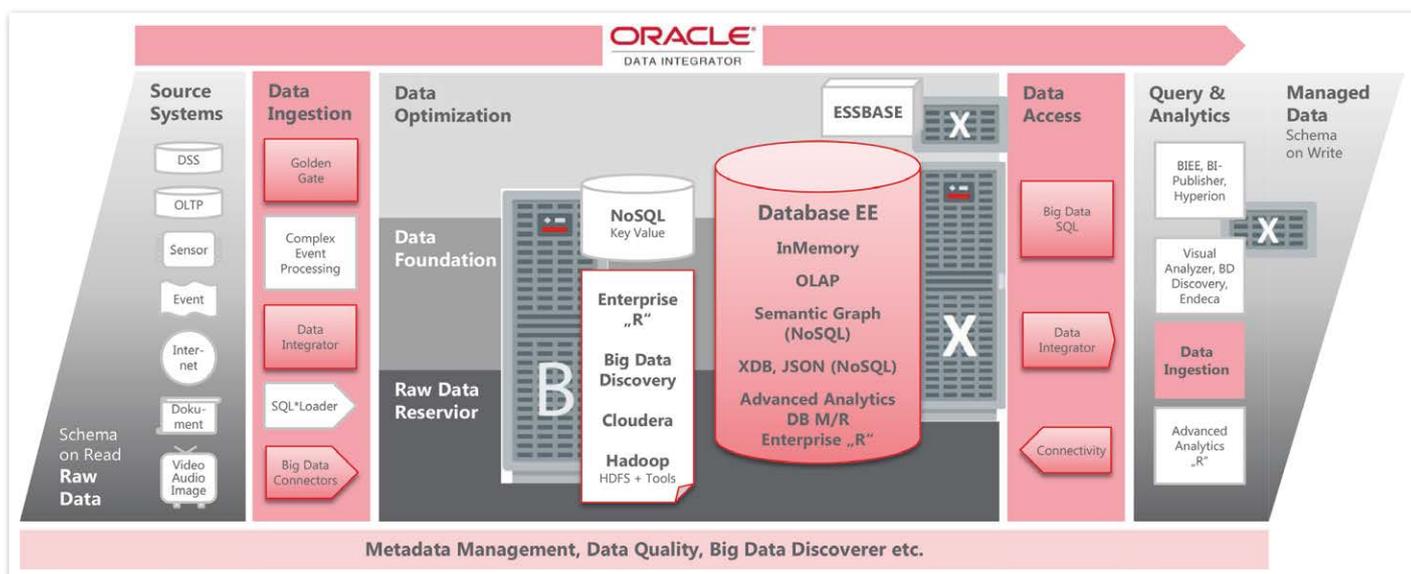


Abbildung 1: Die Oracle „Big Data Management System“-Komponenten – Stand heute – im Überblick

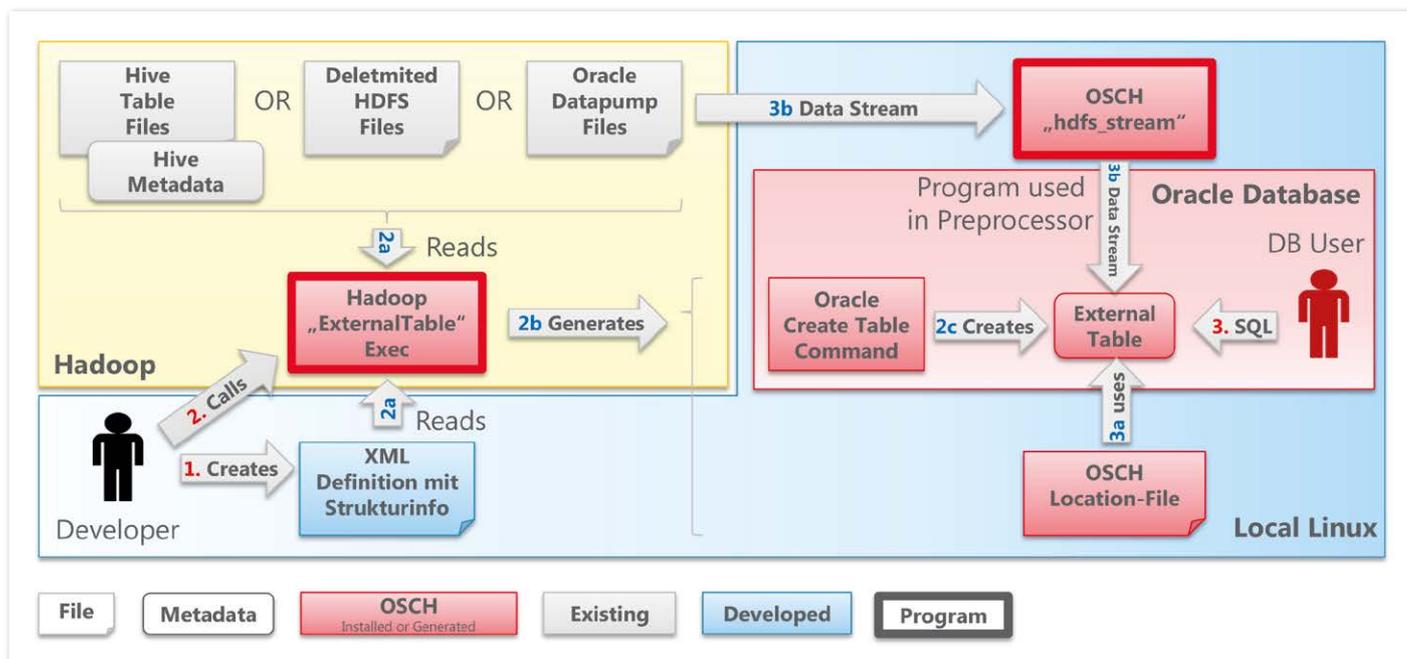


Abbildung 2: Die einzelnen Schritte und Komponenten beim Einsatz von OSCH

„<http://www.oracle.com/technetwork/database/bigdata-appliance/overview/sod-bdms-2015-04-final-2516729.pdf>“ erklärt Oracle eigene Absichten und Visionen für das Zusammenspiel von Big-Data-Technologien und der Datenbank. Langfristig soll alles mehr oder weniger nahtlos zusammenwachsen. Auf dem langen Weg dorthin bilden unter anderem die Big-Data-Konnektoren ODI und Golden Gate den Leim, der beide Welten zusammenhält (siehe Abbildung 1).

Aktuell muss man sich noch selbst darum kümmern, welche Technologie man für welche Anforderung einsetzt. Ideen dafür liefert beispielsweise ein weiteres Oracle Whitepaper (siehe „<http://www.oracle.com/ocom/groups/public/@otn/documents/webcontent/2297765.pdf>“).

In diesem Artikel werden unter anderem die Konnektor-Komponenten in einem Überblick mit Fokus auf die Funktionsweise erklärt. Für mehr Informationen sei auf einschlägige Artikel und Dokumentationen verwiesen. Insbesondere empfehlenswert sind die Hands-on-Übungen zu den „Oracle Big Data Lite“ VMs, die kostenfrei auf der Oracle Website verfügbar sind. Um einen nahtlosen Übergang zu erleichtern, basieren die Beispiele aus diesem Artikel auf den Bit-Data-Lite-Übungen.

### Apache Sqoop

Eingangs sei kurz Sqoop erwähnt. Die Open-Source-Lösung ist das traditionel-

le Hadoop-Werkzeug, um Daten von und nach Hadoop zu bewegen, und darum essenzieller Teil von Hadoop-Distributionen. Es basiert in den neueren Versionen auf einer Client-Server-Architektur. Man kann sich Sqoop wie ein einfaches Daten-Integrationswerkzeug vorstellen. Es nutzt beliebige JDBC-Treiber und direkten Dateizugriff, ist parallelisierbar und via Kommandozeile scriptbar. Es kann in viele ETL-Tools eingebunden werden und wird folgerichtig auch von ODI genutzt, um Daten in die Hadoop-Plattform zu importieren. Behalten wir dies im Hinterkopf und beschäftigen wir uns nun ausführlicher mit den Oracle-eigenen Lösungen.

### Oracle Big Data Connectors

Die lizenzpflichtigen Konnektoren von Oracle bestehen aus diversen Werkzeugen, mit denen Stand heute Daten aus der Big-Data-Welt in der Oracle-Datenbank zugänglich gemacht werden. Für den umgekehrten Weg – also Daten aus der Datenbank beispielsweise in Hadoop einzubinden – hat Oracle bisher lediglich Lösungen ange-

kündigt (Oracle Table Access for Hadoop (OTA4H)), siehe „[https://www.doag.org/konferenz/konferenzplaner/konferenzplaner\\_details.php?id=473721&locS=0&vid=504667](https://www.doag.org/konferenz/konferenzplaner/konferenzplaner_details.php?id=473721&locS=0&vid=504667)“. Die Big Data Connectors bestehen aus vier separaten Werkzeugen:

- Oracle SQL Connector for HDFS (OSCH)
- Oracle Loader for Hadoop (OLH)
- Oracle XQuery for Hadoop (OXH)
- Oracle R Advanced Analytics for Hadoop (ORH)

### Oracle SQL Connector for HDFS

Der Oracle SQL Connector for HDFS (OSCH) erlaubt aus der Datenbank heraus mit der bekannten External-Table-Technik (ET, inklusive Präprozessor) direkte Read-only-Zugriffe auf das Hadoop File System (HDFS). Die unterstützten Formate sind „Oracle Data Pump“, „Delimited Text“ und „Delimited Text aus Apache-Hive-Tabellen“. Letzteres ist etwas erklärungsbedürftig: Apache Hive ist vereinfacht gesagt eine SQL-Schnittstelle auf Dateien in HDFS. Da-

```

hadoop jar \
  $OSCH_HOME/jlib/orahdfs.jar \
  oracle.hadoop.exctab.ExternalTable \
  -conf /home/oracle/movies/moviefact_hdfs.xml \
  -createTable
-- hadoop Befehl
-- OSCH Bibliothek
-- OSCH Java Programm
-- XML mit ET Definition
-- Erzeugt Oracle ET Code
    
```

Listing 1

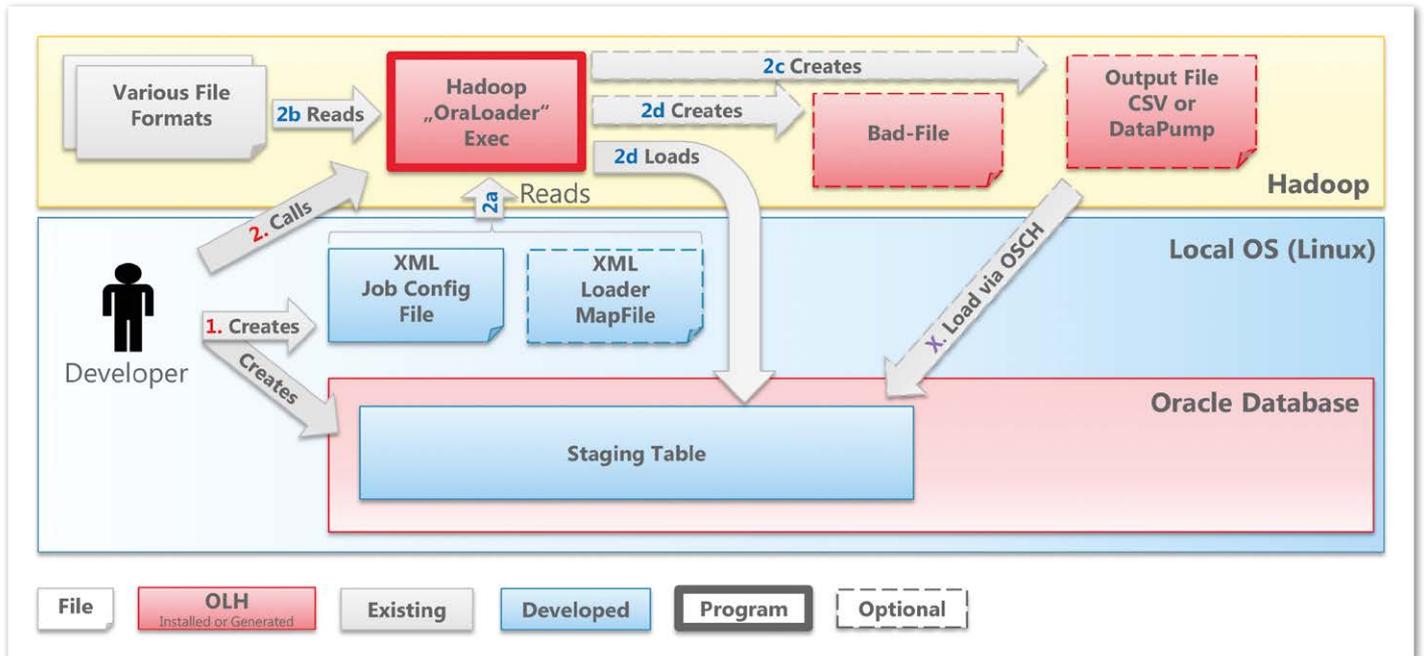


Abbildung 3: Die einzelnen Schritte und Komponenten beim Einsatz von OLH im Online-Modus

bei verwendet Hive ein eigenes Data Dictionary, den sogenannten „HCatalog“. Die Konnektoren können dort hinterlegte Metadaten für das Mapping auf die zugehörigen HDFS-Dateien nutzen. Wichtig: Die SQL-Funktionalität von Hive selbst spielt dann für den Zugriff auf die Daten keine Rolle. Das geschieht direkt über HDFS und Oracle-eigene Komponenten. Leider ist die Verwendung von OSCH nicht ganz so einfach und bedarf einiger Vorbereitung (siehe Abbildung 2).

Im ersten Schritt erzeugt man eine XML-Datei mit Orts- und – soweit nötig – Struktur-Informationen. Diese übergibt man dann in einem zweiten Schritt mittels Hadoop-Client-Aufruf (ein Hadoop Client muss auf dem DB Server installiert sein) an ein Hadoop-MapReduce-Programm, das Oracle mit OSCH mitliefert (siehe Listing 1).

Das Programm erzeugt lediglich einen passenden External-Table-Befehl (auf Wunsch gleich mit Ausführung gegen die Datenbank) und ein sogenanntes „Location File“, das dem OSCH-External-

Table-Präprozessor den Ort der zu lesen den Datei(en) auf dem Hadoop-Filesystem mitteilt. Damit hat man nun eine External Table in der Oracle-Datenbank erstellt, die über einen generischen Streaming-Präprozessor auf eine – oder mehrere gleichformatige – Dateien in HDFS zugreift. Dieser Ansatz unterstützt – genau wie bei einer herkömmlichen External Table vom Typ „ORACLE\_LOADER“ – paralleles Lesen via SQL auf mehrere Dateien, Formatmasken, Datenkonvertierung etc.

Genauso herkömmlich ist allerdings auch der Zugriff auf die Daten: Es werden alle in die Oracle-Datenbank gezogen. Push-down von Filtern aus „WHERE“-Klauseln oder andere Mechanismen, um den Datenstrom zwischen Hadoop und der Datenbank einzuschränken, gibt es nicht. Es ist also keine gute Idee, auf diese Art in Terabyte-großen Hadoop-Dateien x-mal in der Stunde nach wenigen Datensätzen zu suchen. Gut geeignet ist es aber, um eine batchorientierte Leseschnittstelle auf kleinere oder mittelgroße Hadoop-basierte Daten zu implementieren.

### Oracle Loader for Hadoop (OLH)

Dieses Werkzeug nutzt einen etwas anderen Weg, um Daten aus einem Hadoop-Cluster in die Oracle-Datenbank zu importieren. Im Mittelpunkt steht dabei vor allem der reine Durchsatz. OLH dürfte derzeit die schnellste Möglichkeit sein, um Daten aus HDFS in eine Tabelle in der Oracle-Datenbank zu laden. Unterstützt werden die bereits in OSCH genannten Datenformate sowie das AVRO-Format. Zudem gibt es eine (eingeschränkte) Möglichkeit, Daten aus der Oracle-NoSQL-Datenbank zu importieren und beliebige weitere Formate mittels Java-Programmierung (Erweiterung der „MapReduce InputFormat“-Klasse) einzubinden (siehe Abbildung 3).

Der Einsatz ist auch hier mit ein wenig Arbeit verbunden. Zunächst erstellt man ein Job-Config-XML mit allen nötigen Loader-Definitionen wie Dateinamen, Parallelisierungsgrad, Name der Zieltabelle, Infos zur JDBC-Connection zur Datenbank etc. Wenn die Zieltabelle für den Ladevorgang der Quelldatei entspricht und weitere Bedingungen wie ein einheitliches Datumsformat und Spaltennamen passen, genügt das, um einen Online-Mode-Ladevorgang auszuführen. Wenn es komplizierter wird, muss man noch eine weitere XML-Datei mit Mapping-Informationen erstellen. Im nächsten Schritt wird der Loader gestartet und mit den XML-Dateien versorgt (siehe Listing 2).

```
hadoop jar \-- hadoop Befehl
$OLH_HOME/jlib/oraloader.jar \
oracle.hadoop.loader.OraLoader \
-conf /home/oracle/movies/jobconfig.xml
-- Tool aus folgender Bibliothek
-- Name des Tools (Java Programm)
-- XML mit Loaderdefinition
```

Listing 2

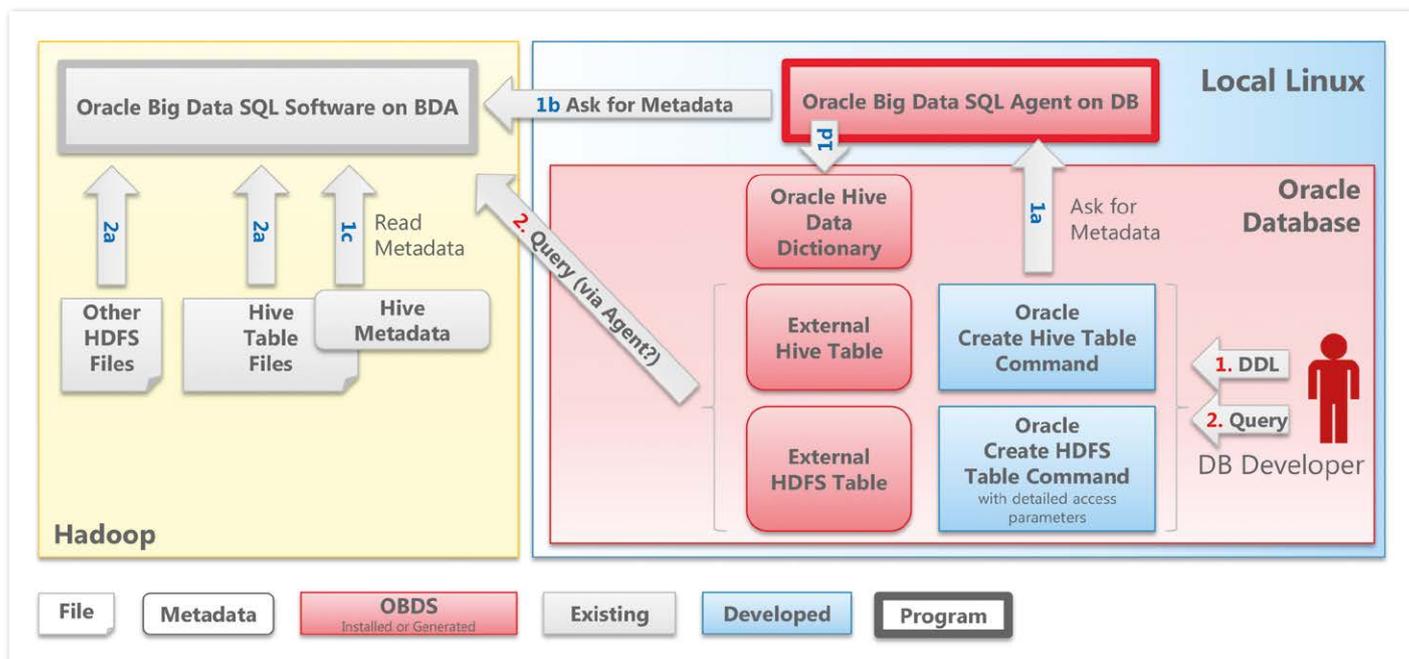


Abbildung 4: Die einzelnen Schritte und Komponenten beim Einsatz von Big Data SQL

Was nun passiert, ist einem klassischen „SQL\*Loader“-Vorgang recht ähnlich: Das „OraLoader“-MapReduce-Programm liest die Dateien, konvertiert sie in ein passendes Ladeformat und lädt sie in eine Tabelle in der Datenbank. Bei Bedarf wird höchste Performance durch Parallelisierung dadurch erreicht, dass jeweils eine Datei in eine Partition einer Zieldatenbank via Direct Insert (nur OCI-Treiber) geladen wird. Das dafür vorab nötige Sortieren und Gruppieren von Daten auf Hadoop erledigt OLH dabei selbstständig. Genau wie beim SQL\*Loader können auch hier Bad Files geschrieben werden. In einem weiteren Modus, dem sogenannten „Offline-Mode“, können die für das Laden vorbereiteten Daten auch in Dateien im Datapump-Format geschrieben und dann später via OSCH geladen werden.

### Oracle Big Data SQL

Betrachten wir die Möglichkeiten, die man beispielsweise mit Datenbank-Links hat, wirkt der Oracle Big Data Connector (OSCH) ungewöhnlich sperrig. Einerseits ist er auf registrierte, einzelne Tabellen beschränkt – klar, bei Zugriffen über External Tables. Insbesondere bietet er aber keinerlei Optimierung für verteilte Abfragen. Die Hadoop-basierten Dateien werden bei jeder Abfrage immer vollständig auf die Oracle-Datenbank gestreamt und erst dort in die übliche Abfrage-Opti-

mierung eingebunden. Drittens ist es mit OSCH ziemlich aufwändig, die External Tables zu erstellen.

Die bessere Oracle-Lösung dafür lautet „Big Data SQL“. Mit diesem Mechanismus können – ähnlich wie in den Exadata Cell Nodes – Smart Scans innerhalb von Hadoop ausgeführt werden. Damit sind auch die Übergabe von Filtern nach Hadoop und die Einschränkung der zurückgegebenen Spalten (Projektion) oder auch effiziente Joins via Bloom-Filter möglich. Oracle benötigt dafür allerdings eigene Komponenten auf dem Hadoop-Cluster und einen speziellen SQL-Agent, der für die Datenbank beispielsweise den Austausch der Hadoop-Metadaten sicherstellt. Dazu kommt der Vorteil, dass das Erstellen von External Tables mit diesem Werkzeug nicht komplizierter sein muss als das Erzeugen normaler External Tables auf normale Dateien im direkten Zugriff der Datenbank.

Klingt im Ansatz also sehr gut, hat aber – neben den Lizenzkosten – ein besonders

unangenehmes Manko: Big Data SQL ist aktuell nur auf der Kombination Exadata/Big Data Appliance verfügbar. In der Hoffnung auf baldige Verfügbarkeit des Features auch für anderen Konstellationen – das Statement-of-Direction mag man in diese Richtung interpretieren können – betrachten wir die Lösung im Folgenden dennoch etwas genauer (siehe Abbildung 4).

Nachdem Big Data SQL auf DB-Seite konfiguriert wurde – und damit die Hadoop-Locations in Oracle-Directories gemappt sind – ist es recht einfach, HDFS-basierte Dateien als External Tables mit nur einem Befehl einzubinden. Das Beispiel in Listing 3 ermöglicht das Lesen einer Log-Datei mit genau einer Spalte im JSON-Format).

Wichtig dabei sind zwei neue External-Table-Typen namens „ORACLE\_HDFS“ und „ORACLE\_HIVE“. Erstere für den direkten Zugriff auf alle möglichen Dateien in HDFS, die zweite für den Zugriff auf

```
CREATE TABLE movielog
  (click VARCHAR2(4000))          -- Für ganze Applog Zeile
  ORGANIZATION EXTERNAL          -- External Table
  (TYPE ORACLE_HDFS              -- Neuer Typ ORACLE_HDFS
  DEFAULT DIRECTORY DEFAULT_DIR
  LOCATION ('/user/oracle/dat/') -- Dateien auf Hadoop Cluster
  );
```

Listing 3

```

CREATE TABLE movieapp_log (
    custid INTEGER,      movieid    INTEGER ,      genreid INTEGER ,
    time   VARCHAR2(20), recommended VARCHAR2(4),  activity NUMBER,
    rating INTEGER,     price     NUMBER )      ORGANIZATION EXTERNAL
(TYPE ORACLE_HIVE
ACCESS PARAMETERS (
    com.oracle.bigdata.tablename: log_db.applog
    com.oracle.bigdata.colmap: {"col": "CUSTID", field: "cust_id"}
    ... )
);

```

Listing 4

Hive-Tables direkt via HCatalog-Metadaten. *Listing 4* zeigt ein Beispiel dafür.

Nun sind deutlich intelligentere Zugriffe auf Hadoop-basierte Daten möglich und auch einer Einbindung in verteilte Datenbank-Abfragen steht nichts mehr im Wege – außer natürlich den üblichen, schon bei Database-Links bekannten Distributed-Query-Problemen.

### Oracle Database Gateway

Was bisher auffällt, ist die praktisch völlige Absenz von Hadoop-eigenen Abfrage-Mechanismen und Tools, außer HCatalog für die Metadaten. Bei allen betrachteten Tools nutzt Oracle fast ausschließlich eigene Programme für den Zugriff auf die Daten. Damit sind ein sauberes Typen-Mapping und eine zentrale Kontrolle über die Kompatibilität möglich. Eine andere Möglichkeit – und die ist interessanterweise sogar weitgehend kostenfrei, wenn man von den Lizenzen für ODBC-Treiber absieht – ist der Einsatz des „Generic ODBC Gateway“. Früher auch bekannt als „Heterogenous Services“, bietet Oracle schon lange die Möglichkeit, via Datenbank-Links auch auf andere Datenbanken als auf die von Oracle zuzugreifen. Die darunter versammelten Lösungen sind fast alle kostenpflichtig – bis auf die Variante für „GENERIC ODBC“, bei der beliebige ODBC-Treiber für den Zugriff auf beliebige Datenquellen eingesetzt werden können (sogar auf Microsoft Excel).

Um es klar zu sagen: Man wird sich einige Limitierungen bei Kompatibilität und Performance einhandeln. Dennoch ist dieser Ansatz – gerade für erste Schritte, Evaluationen oder bei nur vereinzelt, selektiven Zugriffen auf die Hadoop-Welt – durchaus einen Blick wert. Folgende Schritte sind auf dem Datenbankserver

(und/oder einem extra Gateway Server) auszuführen:

1. Passenden ODBC-Treiber installieren und konfigurieren
2. ODBC-Gateway installieren
3. „listener.ora“ und „tnsnames.ora“ anpassen
4. In „ORACLE\_HOME\hs\admin“ eine neue Datei erzeugen
5. Public-Datenbank-Link erzeugen

Gehen wir von einem „HIVE ODBC“-Treiber aus. In diesem Fall wird die SQL-Engine von Hive selbst in die Abfragen eingebunden. Man hat also nach der Einrichtung des Datenbank-Links freien Zugriff auf alle Hive-Tabellen, auf die man als Hive-User Zugriff hat – nicht anders als bei normalen Datenbank-Links – und nutzt die Rechenleistung des Hadoop-Clusters via Hive. Sogar die Metadaten kann man direkt über das Oracle Data Dictionary nutzen. Befehle wie die in *Listing 5* funktionieren also oft problemlos, während die Last der Zugriffe und der Ort der Metadaten außerhalb der Oracle-Datenbank verortet sind.

Nicht jede Hadoop-Datenquelle oder jeder ODBC-Treiber funktioniert allerdings gleich gut. Mit der In-Memory-SQL-Lösung für Hadoop „Impala“ von Cloudera und dem dazugehörigen ODBC-Treiber sind beispielsweise heute nur wenige Metadaten verfügbar, während mit Apache Hive und den dafür verfügbaren Treibern aktuell schon sehr viele Metadaten-

Informationen ermittelt werden können. Hier heißt es: Gut evaluieren.

### Oracle Data Integrator

Für den ODI gibt es eine Big-Data-Option (separate Lizenz), die für Mappings nativen Code für die Hadoop-Werkzeuge „Pig“ (eine Scripting Engine, die ihrerseits wieder MapReduce-Code generiert), „Spark“ (eine modernere und schnellere MapReduce-Alternative) und „Oozie“ (das zentrale Workflow-Tool in der Hadoop-Welt) generiert.

Darüber hinaus erlaubt es die Option, entweder den traditionellen ODI-Agent oder Apache Oozie als Orchestrierungs-Engine für ODI einzusetzen. Sie bringt einen schnellen WebLogic Hive JDBC Driver und zahlreiche Direct Load KMs (LKM = Loading Knowledge Modules, kombiniert mit anderen KMs in einem Mapping einsetzbar) etwa für Sqoop mit.

Nutzt man die Oracle Big Data Connectors, erhält man damit auch die Lizenz für Oracle Data Integrator Application Adapter for Hadoop, mit dem die Konnektoren über entsprechende Knowledge-Module in ODI eingebunden werden können. Gleiches gilt für Big Data SQL. Auch hier ist eine Integration in ODI verfügbar – wenn man die entsprechenden Systeme und Lizenzen hat.

### Oracle Golden Gate

Oracle Golden Gate for Big Data unterstützt Golden-Gate-Replikation von allen bekannten Golden-Gate-Quellen (Oracle,

```

SELECT COUNT(*) FROM mytab@hive WHERE cust_cat = 95;
SELECT table_name FROM all_tables@hive;

```

Listing 5

DB2, SQL Server, MySQL, Informix, Sybase ASE, SQL/MX, JMS etc.) in die Big-Data-Welt – allerdings nicht in die Gegenrichtung. Mit Change-Data beliebert werden Apache Flume, HDFS direkt, Hive und die Hadoop-eigene NoSQL-Datenbank HBase. Über die ebenfalls in der Lizenz enthaltene „Golden Gate for Java“-Lösung ist auch die Integration der Ziele Oracle-NoSQL-Datenbank, Apache Kafka, Apache Storm und Apache Spark möglich.

### Fazit

Die Zusammenarbeit mit Big-Data-Plattformen steht im Zentrum der heutigen Oracle-Big-Data-Aktivitäten. Die entsprechenden Oracle Big Data Connectors (OLH, OSCH, OXH etc.) sind dafür schon länger verfügbar, allerdings nur für bestimmte Anforderungen geeignet und auch nicht unbedingt einfach zu benutzen. Oracle Big Data SQL vereinfacht und beschleunigt die Konnektivität deutlich,

ist äußerst vielversprechend – aber lizenzmäßig extrem limitiert und daher leider für die meisten Kunden heute irrelevant.

Nicht von Oracle propagiert, aber dennoch interessant ist Oracle Generic ODBC Database Gateway als kostengünstige und einfache Option für Big-Data-Plattform-Verbindungen ohne besondere Ansprüche an Durchsatz und Kompatibilität.

ODI bietet in den neuen Versionen ebenfalls zahlreiche KMs rund um Hadoop & Co. Mit der Big-Data-Option ist sogar die Nutzung von Hadoop als Daten-Integrationsplattform greifbar geworden, weil Mappings auch direkt auf Hadoop-Distributionen ausführbaren Code generieren, der früher der relationalen Welt vorbehalten war. Auch Golden Gate mit Hadoop & Co. als Replikationsziel erweitert sein Repertoire in Richtung Big-Data-Technologien.

Aus Sicht eines Data-Warehouse-Entwicklers offenbaren sich mit der Annähe-

rung beider Welten neue Möglichkeiten für daten- und kostengerechtes Processing, Data Lifecycle Management (etwa Archive auf Hadoop), Self Service BI und vieles mehr. Nicht zuletzt blicken wir gespannt in die Zukunft und warten auf Lösungsansätze rund um die Oracle-Vision eines Big-Data-Management-Systems.



Peter Welker

[peter.welker@trivadis.com](mailto:peter.welker@trivadis.com)

## Ein großer Schritt für die Oracle-Lizenzierung in virtuellen Umgebungen

In einem gemeinsamen Gespräch der DOAG, SOUG (Swiss Oracle User Group) und AOUG (Austrian Oracle User Group) am 18. November 2015 auf der DOAG 2015 Konferenz + Ausstellung mit Andrew Mendelsohn, Executive Vice President Database Server Technologies bei Oracle, wurde deutlich, dass es im Bereich der lange diskutierten Lizenzierungsproblematik hoffungsvolle Fortschritte gibt.

Bisher ist es je nach eingesetzter Hypervisor-Version vorgeschrieben, den Cluster, alle von der Management-Konsole erreichbaren Host oder die gesamte virtuelle Infrastruktur zu lizenzieren. Bei VMware wäre das bis einschließlich Version 5.0 der Cluster, ab Version 5.1 das vCenter und ab Version 6.0 die gesamte virtuelle Infrastruktur. Laut Mendelsohn gibt es Anzeichen dafür, dass die Segmentierung der Hypervisor- oder VM-Umgebung mit VLAN-Technologie zunehmend akzeptiert wird.

Eine Segmentierung der virtuellen Netze in logische Gruppen hat den Vorteil der

Trennung der Umgebungen bei gleichzeitiger Beibehaltung von Datenkommunikation. Das VLAN kann somit als Begrenzung des lizenzpflichtigen Bereichs angenommen werden, so dass nur die Host-Server innerhalb des VLANs lizenziert werden müssen, da es mit den derzeit bekannten Hypervisor-Technologien nicht möglich ist, virtuelle Maschinen im Betrieb über die Grenze des VLANs hinaus zu verschieben.

Die VLAN-Lösung gilt sinngemäß für alle Hypervisor-Anbieter, die das Verschieben virtueller Maschinen im Betrieb über Clustergrenzen hinaus möglich machen. Der-

zeit muss dies auf dem Wege eines Einzel-Approvals durch Oracle genehmigt werden. Dies ist noch nicht die von den Usergroups und von vielen Kunden und Partnern erwartete allgemeingültige Lösung.

Die DOAG, SOUG und AOUG begrüßen diesen wichtigen Schritt aber ausdrücklich und ermutigen ihre Mitglieder, von dieser Möglichkeit Gebrauch zu machen. Allerdings raten sie dazu, das Verhalten der virtuellen Maschinen mit System Logs zu dokumentieren und zu archivieren. Die Usergroups wünschen sich jedoch noch immer eine einheitliche und allgemeingültige Regelung.

# Produkt-Update für die Oracle-Big-Data-Plattform

Jean-Pierre Dijcks, Oracle Corp.

Rund um das Thema „Big Data“ war es und ist es wohl nie langweilig bei Oracle. Nun gibt es aus dem Development wieder eine Reihe von spannenden neuen Features, Produkten und Services rund um das Big-Data-Management-System und seine Komponenten zu vermelden. Hier ein kurzes Update. Zur DOAG Konferenz im November werden sicherlich noch weitere vorgestellt.

Bezüglich Big Data strebt Oracle eine ganzheitliche Lösung an. So integriert man Hadoop-Technologie, NoSQL- und relationale Datenbank in einem zusammenhängenden System. Zusätzlich kann man diesen Verbund auch als Public Cloud nutzen.

## Big-Data-Cloud-Service

Die Big-Data-Cloud steht jetzt offiziell zur Verfügung, Sie ist besonders interessant, weil man jetzt das komplette Big-Data-Management-System sowohl On-Premise als auch in der Cloud nutzen kann – und zwar ohne funktionalen Unterschied. Unternehmen können also das Big-Data-Management-System in der Cloud unter den gleichen technischen Rahmenbedingungen nutzen, als wenn sie die Lösung im eigenen Rechenzentrum stehen hätten (siehe *Abbildung 1*). Darüber hinaus sind weitere Services wie die Big-Data-Discovery-Cloud und der Big-Data-Preparation-Cloud-Service im Angebot.

## Big Data Appliance

Die Big Data Appliance wurde bereits zu Beginn des Jahres unter dem Namen „X5-2“ ausgebaut. Die Kapazität der Platten ist auf 8 TB gestiegen. Die Maschine verfügt also jetzt über 1,7 PB Rohdaten-Kapazität. *Abbildung 2* zeigt eine kurze Übersicht über die Erweiterung.

Noch eindrucksvoller als der riesige Speicher-Ausbau ist die CPU-Erweiterung auf die neueste Intel-Xeon-Prozessor-Generation. Heutige Big-Data-Anwendungen

haben häufig sehr große Anforderungen: MapReduce für ETL, SQL für Ad-hoc-Abfragen und Spark für spezielle, hochperformante Rechenoperationen verlangen nach enorm viel CPU-Rechen-Power und das vor allem auf gewaltigen Datenmengen. Die 658-CPU-Kerne der neuen Intel-Prozessoren ermöglichen dies.

Auf der Software-Seite ist es jetzt möglich, von der Big Data Appliance aus auf Daten in der Oracle-Datenbank zuzugreifen („Oracle Table Access for Hadoop“). Damit können beispielsweise Hive-Anwendungen Joins zwischen Daten auf der BDA und der Oracle-Datenbank bilden, ohne dass die Datenbank-Daten zunächst auf die BDA kopiert werden müssen. Das erleichtert viele Abläufe auf der BDA enorm. Zudem gibt es jetzt eine Reihe weiterer Automatismen, mit denen sich Daten sehr leicht von der Datenbank in das Hadoop-System auf der BDA kopieren lassen. Neue Interfaces für den SQL Developer unterstützen diese Vorgänge grafisch.

## Big Data SQL

Big Data SQL in der Version 2 liefert mit dem schon von der Exadata her bekannten Storage Index ein wichtigstes neues Feature. Der automatisch durch das System im Hauptspeicher gebildete Index dient primär der I/O-Minimierung. Im Gegensatz zur Exadata wird auf der Big Data Appliance ein Index pro 256 MB Block und auf 32 Spalten gebildet. Er ist für die Oracle-Datenbank und für das HDFS völlig transparent und wird automatisch ungültig, wenn sich an den betroffenen Daten etwas ändert (siehe *Abbildung 3*).

Der Storage Index ist an die jeweiligen Spalten geknüpft, die der Benutzer über eine External Table vom Typ „ORACLE\_HIVE“ oder „ORACLE\_HDFS“ abfragt. Wird eine Spalte in einer Abfrage angefasst, schaltet sich der Index sofort ein.

## Big Data Spatial and Graph

Big Data Spatial und Graph steht seit geraumer Zeit auch für die Hadoop-Umgebung zur Verfügung. Eine weitere neue Kompo-



Abbildung 1: Oracle-Big-Data-Cloud-Service: Private und Public Cloud

nente hier ist „Multimedia Analytics“. Damit lassen sich Filme oder Bildarchive scannen und Suchmuster beziehungsweise bestimmte Bildinhalte erkennen. Gesichtserkennung ist beispielsweise eine Lösung, die out of the box damit machbar ist. Multimedia-Analysen nutzen in optimaler Weise die extreme Rechenleistung eines Hadoop-Clusters aus. Bilder können in nahezu unbegrenzter Menge vorliegen und sie werden in einer akzeptablen Zeit analysiert.

Diejenigen, die bislang noch nicht mit Spatial- oder Graph-Daten in Berührung kamen, sollten unbedingt einen Blick auf diese Technologie werfen. Geografische Informationen kommen bei fast allen Daten vor, die wir heute in einem Hadoop-System speichern. Auch Spatial- und Graph-Daten sind ideale Kandidaten für das Skalierungskonzept von Hadoop. Daten, kombiniert mit diesen Spatial-Informationen, lassen sich auch in großer Menge sehr leicht visualisieren, sind dann leichter zu verstehen und wir erhalten eine Reihe zusätzlicher Analyseverfahren.

Big Data Graph bringt eine neue Graphen-Datenbank in das Eco-System. In-Memory lassen sich Graphen-Analysen auf großen Datenmengen im HDFS durchführen. Hier ist der augenfälligste Anwendungsfall sicherlich die Analyse von Social-Media-Kommunikation und -Beziehungen. Man speichert die Kommunikations- beziehungsweise Freundschaftsbeziehungen in der Datenbank und kann wie in einem Netzwerk von Person zu Person Verbindungen erkennen. Anwen-

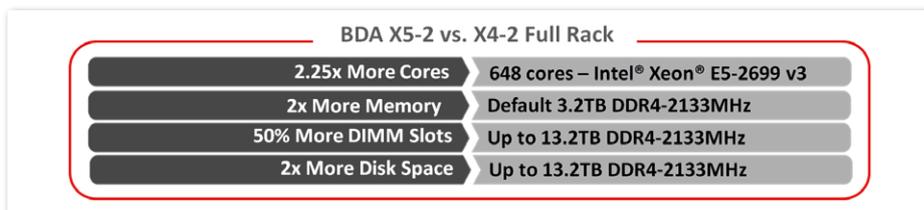


Abbildung 2: Vergleich der Big-Data-Appliance-Komponenten X4-2 mit dem jüngsten Release X5-2

dungsfälle sind aber auch Internet-of-Things-Lösungen mit ihren Interaktionen zwischen den Dingen oder „Cyber Security“ und Verbrechenbekämpfungs-Anwendungen.

### Mit den Hadoop-Daten arbeiten

Es gibt noch eine Reihe weiterer wichtiger Neuerungen. Einige davon sind:

- Für Oracle R Advanced Analytics for Hadoop wurden bereits zwei Algorithmen (GLM, Neural Network) auf die Spark-Technologie umgestellt. Das Ergebnis sind um die Faktoren 100 bis 200 verkürzte Laufzeiten. Diese Umstellung mit den Performance-Resultaten zeigen, wo die Reise in der Hadoop-Entwicklung hingeht. Weitere Data-Mining- und Analyse-Algorithmen werden folgen.
- Big Data Discovery steht jetzt in der Version 1.1 zur Verfügung. In dem Werkzeug sind jetzt noch mehr Datenarten (Hadoop und RDBMS) abfragbar.
- Für Streaming Data gibt es ein neues grafisches Interface, den Oracle Stream

Explorer. Hier haben Analysten ein intuitives Tool für Streaming-Analysen.

Die genannten neuen Lösungen machen es immer leichter, auch „Big Daten“ zu analysieren. Solche Tools fehlten ja in den ersten Jahren der Hadoop-Entwicklung.

### Von zu Hause aus testen

All diese Features kann man selbst ohne große Hürden ausprobieren. Oracle bietet hierfür die Oracle Big Data Lite Virtual Machine an. Diese wird für jede aktuelle Version der Big Data Appliance angepasst und erweitert. Sie ist perfekt geeignet, um die unterschiedlichen Aspekte von Big Data ohne besonderen Aufwand zu testen und ohne selbst einen eigenen Big-Data-Cluster aufbauen zu müssen (siehe „<http://www.oracle.com/technetwork/database/bigdata-appliance/oracle-bigdatalite-2104726.html>“).

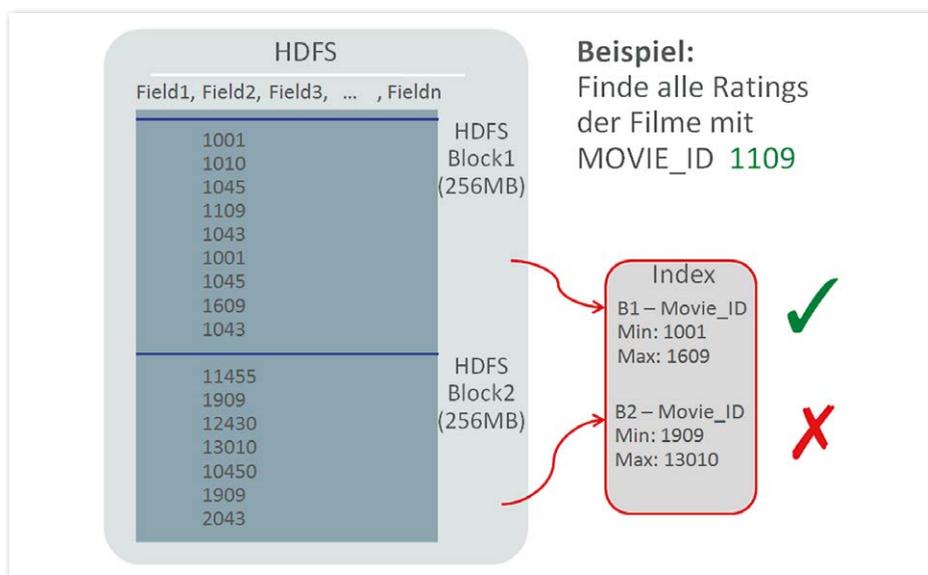


Abbildung 3: Der neue Storage Index arbeitet auf HDFS-Block-Ebene und erzielt eine verbesserte I/O-Performance durch das Überspringen nicht benötigter Blöcke



Jean-Pierre Dijcks



Ins Deutsche übersetzt von Alfred Schlaucher  
alfred.schlaucher@oracle.com

# Oracle-Datenbank – Schaltzentrale für Big Data, In-Memory und Exadata

Dr.-Ing. Holger Friedrich, sumIT AG

Oracle bietet mittlerweile verschiedene Speicher-Technologien als Quellen für die Oracle-Datenbank an, insbesondere die Oracle Big Data Appliance, Exadata Storage und den Columnar Store der In-Memory-Datenbank-Option. Jede dieser Technologien bietet eigene Performance-Features. Die Oracle-Datenbank kann auf alle diese Speicherarten transparent mit Standard-SQL zugreifen, was einzigartig im RDBMS-Markt ist.

Entscheidend ist zu wissen, welche der Speicher-Technologien und welche spezifischen Eigenschaften für die Oracle-Datenbank zur Verfügung stehen, welche Vorteile sich daraus ergeben, aber auch, welche Grenzen bestehen. Sind den Architekten und den Entwicklern diese Grenzen bewusst, kann eine System-Architektur aufgebaut und effizient genutzt werden, um die Anforderungen des Kunden mit dem besten Kosten-Nutzen-Verhältnis zu erfüllen.

Bis zur Einführung der Exadata-Storage-Technologie zur Oracle OpenWorld 2008, also vor sieben Jahren, basierte die Oracle-Datenbank jahrzehntelang auf derselben Speicher-Architektur und der daraus resultierenden Verarbeitungslogik. Diese bestand primär aus den folgenden drei Speicherarten:

- **System Global Area (SGA)**  
Dies ist der Bereich im Arbeitsspeicher (RAM) der Rechner, auf denen die Oracle-Datenbankprozesse laufen, der allen Prozessen des Datenbankservers gemeinsam zur Verfügung steht. Im Sinne unserer Diskussion der Speicherformen in Bezug auf Abfragen ist hier insbesondere der Block Buffer interessant. In diesem befinden sich Datenblöcke, die aus dem persistenten Speicher, also von Festplatten oder Flash-Memory, gelesen wurden und für weitere (Teil-)Abfragen im schnellen RAM gecacht werden.
- **Program Global Area (PGA)**  
Die PGA ist ein Arbeitsspeicher-Bereich, der individuelle Daten und Ver-

arbeitungsprozesse für jeweils einen Serverprozess enthält. Jeder Datenbankprozess bekommt beim Start für seine Lebenszeit eine eigene, exklusive PGA zugewiesen. Es gibt also genau eine PGA für jeden Serverprozess. Hintergrundprozesse allokieren ebenfalls ihre eigenen PGAs. Das Verhältnis von SGA zu PGA ist in *Abbildung 1* wiedergegeben.

Der spannende Aspekt der PGA, im Rahmen unserer Betrachtung alter und neuer Speicherformen, ist, dass hier bezüglich der logischen Abarbeitung von SQL-Abfragen hauptsächlich die

Musik spielt. Hier wird beziehungsweise wurde traditionell gefiltert, projiziert, sortiert, aggregiert, partitioniert, gehasht und gejoint, werden Muster gematcht, reguläre Ausdrücke verarbeitet, wird gerechnet und werden alle weiteren Verarbeitungsschritte ausgeführt.

- **Persistenter Speicher (Festplatten, Flash, etc.)**

Auf persistenten Speichermedien sind Datenblöcke und Redo-Information gesichert. Auch temporäre Daten wie Undo-Informationen und Teile von PGAs werden dort gespeichert; letzte-

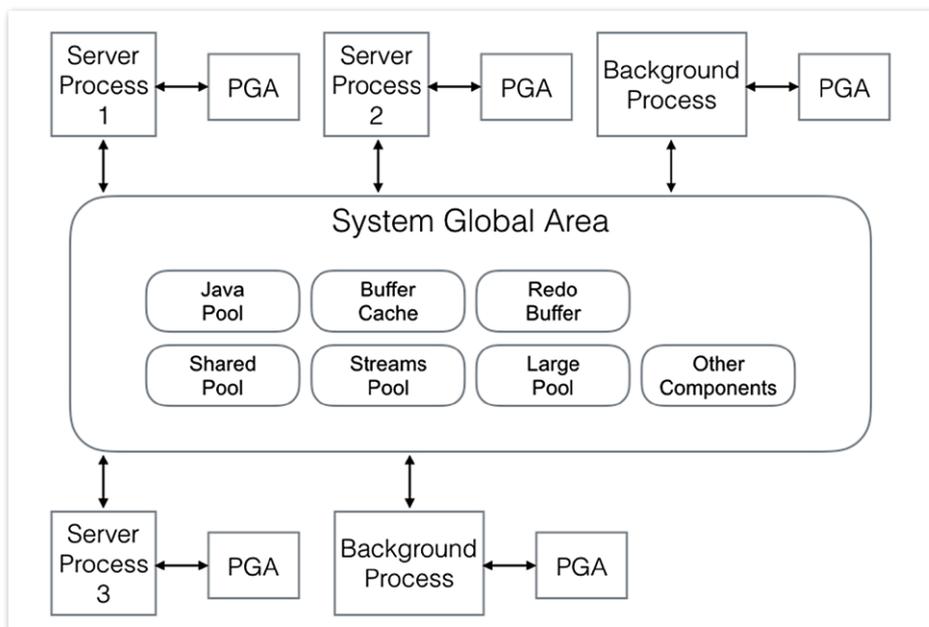
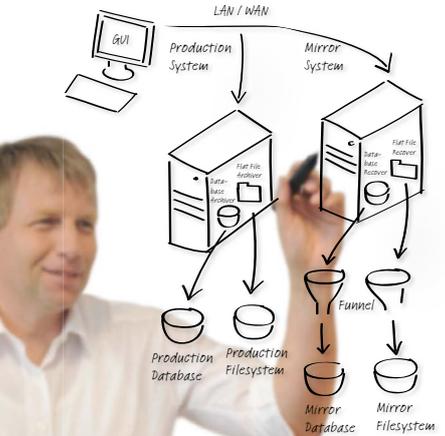


Abbildung 1: Traditionelle Arbeitsspeicher-Strukturen der Oracle-Datenbank

## Libelle BusinessShadow®



Unabhängig bezüglich

- Fehlerursache
- Entfernung
- Hardware / Architektur
- Komplexer Systeme

Schnelle Arbeitsaufnahme

- Mit konsistenten Daten
- Auf Knopfdruck
- Automatisiert
- ...

Hans-Joachim Krüger  
Chief Technology Officer  
Libelle AG

Erfahren Sie mehr:  
[www.Libelle.com/business](http://www.Libelle.com/business)



ORACLE Gold Partner



Libelle

Libelle AG  
Gewerbestr. 42 • 70565 Stuttgart, Germany  
T +49 711 / 78335-0 • F +49 711 / 78335-148  
[www.Libelle.com](http://www.Libelle.com) • [sales@libelle.com](mailto:sales@libelle.com)

re im Temp-Tablespace, sollte der Arbeitsspeicher in den Datenbankservern nicht ausreichen. Der klassische persistente Speicher ist dabei aus Sicht der Verarbeitungslogik von Datenbankabfragen nicht intelligent. Er stellt lediglich die Funktionalität zum Lesen und Schreiben von Dateien mithilfe geeigneter I/O-Schnittstellen bereit.

Eine aktuelle Oracle-12c-Datenbank arbeitet auch mit diesen Speicherbereichen, sofern man keine der neuen Speicher- und Speicherzugriffs-Technologien wie Exadata, Big Data (SQL) oder den In-Memory Columnar Store nutzt. Diese Architektur hat zwei prinzipielle Nachteile – einen, der sich auf Performance, und einen, der sich auf die Datenverfügbarkeit und Kosten auswirkt:

### • Performance

In der klassischen Oracle-Architektur werden einzig im Bereich des individuellen Verarbeitungsprozesses und seiner PGA die Daten wirklich verarbeitet. Die beiden anderen Speicherbereiche, SGA und persistenter Speicher, sind de facto Ablagen. Das heißt aber, dass eine Unmenge an Daten vom persistenten Speicher zum DB-Rechner und vom SGA in den jeweiligen DBA transportiert werden müssen, ohne dass sie zur Ergebnisberechnung benötigt werden. Grund dafür sind insbesondere die zeilenorientierte Speicherdarstellung, die Verwendung von Datenbankblöcken fixer Größe bei der persistenten Speicherung sowie im Block Buffer des SGA und die Tatsache, dass jegliche Verarbeitungslogik erst in den PGAs ausgeführt wird.

### • Verfügbarkeit und Kosten

Die Oracle-Query-Engine kann in der klassischen Architektur ausschließlich auf Daten arbeiten, die im Oracle-Datenbankblock-Format vorliegen. Die Datensätze müssen also in Zeilenform strukturiert und in Oracle-Datenblöcke zusammengefasst sein. Werden sie persistent gespeichert, müssen sie im Oracle-Datenbank-Fileformat formatiert gesichert werden. Diese Einschränkung bedingt aufwändige Vorarbeiten (Laden, Transformieren, Speichern), bevor neue Daten und Datenquellen für Abfragen erschlossen sind, was sowohl hohe

Kosten verursacht als auch eine längere Wartezeit erzeugt, bis Daten für Auswertungen zur Verfügung stehen.

Im September 2008 begann für die Oracle-Speicher-Architektur eine neue Zeitrechnung. In den folgenden Abschnitten werden die seitdem eingeführten drei neuen Speicher-Technologien Exadata Storage, Big Data (SQL) und der In-Memory Columnar Cache vorgestellt sowie ihre spezifischen Eigenschaften diskutiert.

### Exadata Storage

Auf der Oracle Open World 2008 kam der Oracle-CEO Larry Ellison auf die Bühne und versprach eine neue Appliance, mit der insbesondere datenintensive Abfragen zehnmal schneller verarbeitet würden als je zuvor. Diese Database Machine bestand aus Datenbank-Rechnern, einer InfiniBand-basierten Vernetzung und den sogenannten „Exadata-Speicherzellen“. Der Schlüssel zur zehnfachen Beschleunigung von Abfragen lag bei dieser Architektur nicht in den Rechnerknoten und auch nicht primär in der Anbindung der Exadata-Speicherzellen per InfiniBand. Der Hauptgrund für die überlegene Performance der heute einfach „Exadata“ genannten Appliance liegt im Aufbau und in der Funktionalität der Exadata-Storage-Zellen.

Eine Exadata-Storage-Zelle besteht aus achtzehn Festplatten (rotierend oder Flash), einer größeren Menge Flashspeicher und insbesondere einem eigenen Rechnerknoten. Der Rechnerknoten einer Einheit bearbeitet und administriert seine Speicherzelle. Auf dem Rechnerknoten jeder Exadata-Speicherzelle läuft die Exadata-Storage-Software. Diese bietet der Oracle-Datenbank-Software, die auf den eigentlichen, klassischen Datenbankservern läuft, weitaus komplexere Funktionen als nur das Lesen und Schreiben von Datenbankblöcken an. Eine Exadata-Storage-Zelle kann im Auftrag eines Datenbankserver-Prozesses zum Beispiel lokal folgende Funktionen ausführen:

- Daten filtern
- Daten projizieren
- Teil-Joins ausführen (Joins)
- aggregieren (wie zählen, summieren)
- Data-Mining-Funktionen berechnen (wie scoring)

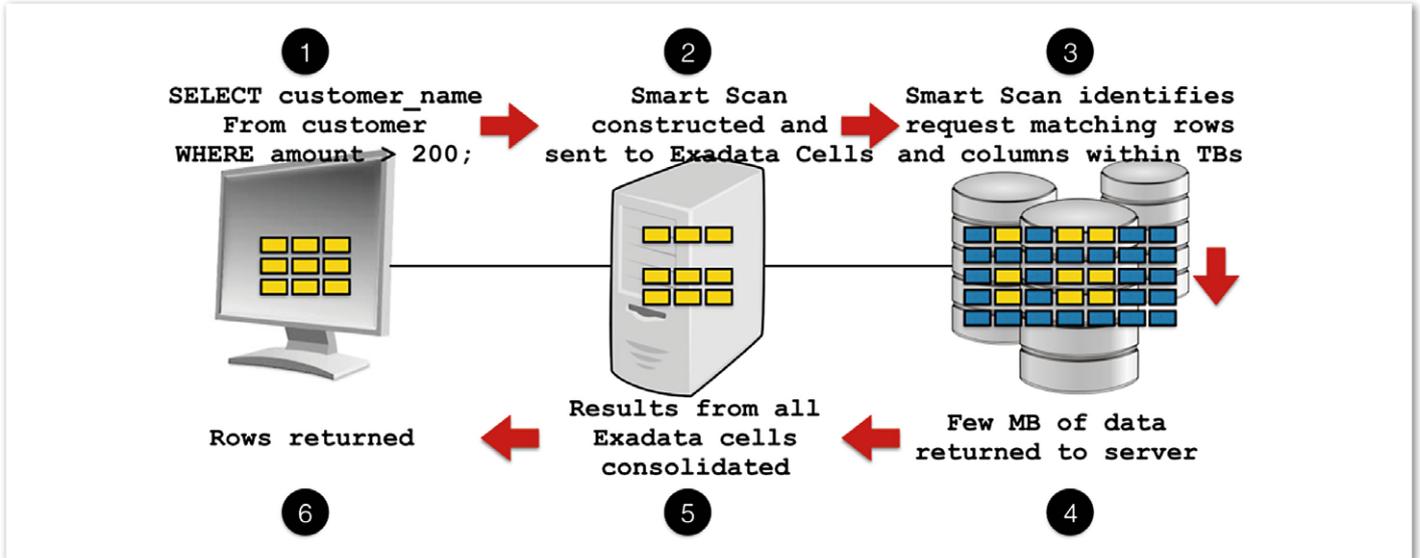


Abbildung 2: Smart Scanning in Exadata Storage

Um diese Funktionen möglichst effizient zu implementieren, nutzt die Exadata-Speicherzelle verschiedene Techniken wie Storage-Indizes, lokales Scanning und Hybrid Columnar Compression. Aus den Ergebnissen der Verarbeitung werden abfragespezifische Zeilen gebildet und diese im regulären Oracle-Datenbankblockformat zusammengefasst. Lediglich diese Teil-

ergebnisse zu einer Gesamtanfrage werden schließlich zu den Datenbankservers übertragen.

Der Performance-Gewinn resultiert also aus der Verlagerung von Teilberechnungen aus dem Datenbankservers und hier aus dem PGA heraus in den Storage-Layer und aus der damit einhergehenden Reduktion unnötig zu transportierender

Datenmengen. Die Fähigkeiten der Speicherzelle sind unter dem Begriff „Smart Scanning“ zusammengefasst; die Tatsache, dass eine Speicherzelle derart von einem Datenbankservers in Betrieb genommen werden kann, wird „Query Off-loading“ genannt. *Abbildung 2* stellt ein Beispiel für erfolgreiches Smart Scanning grafisch dar.

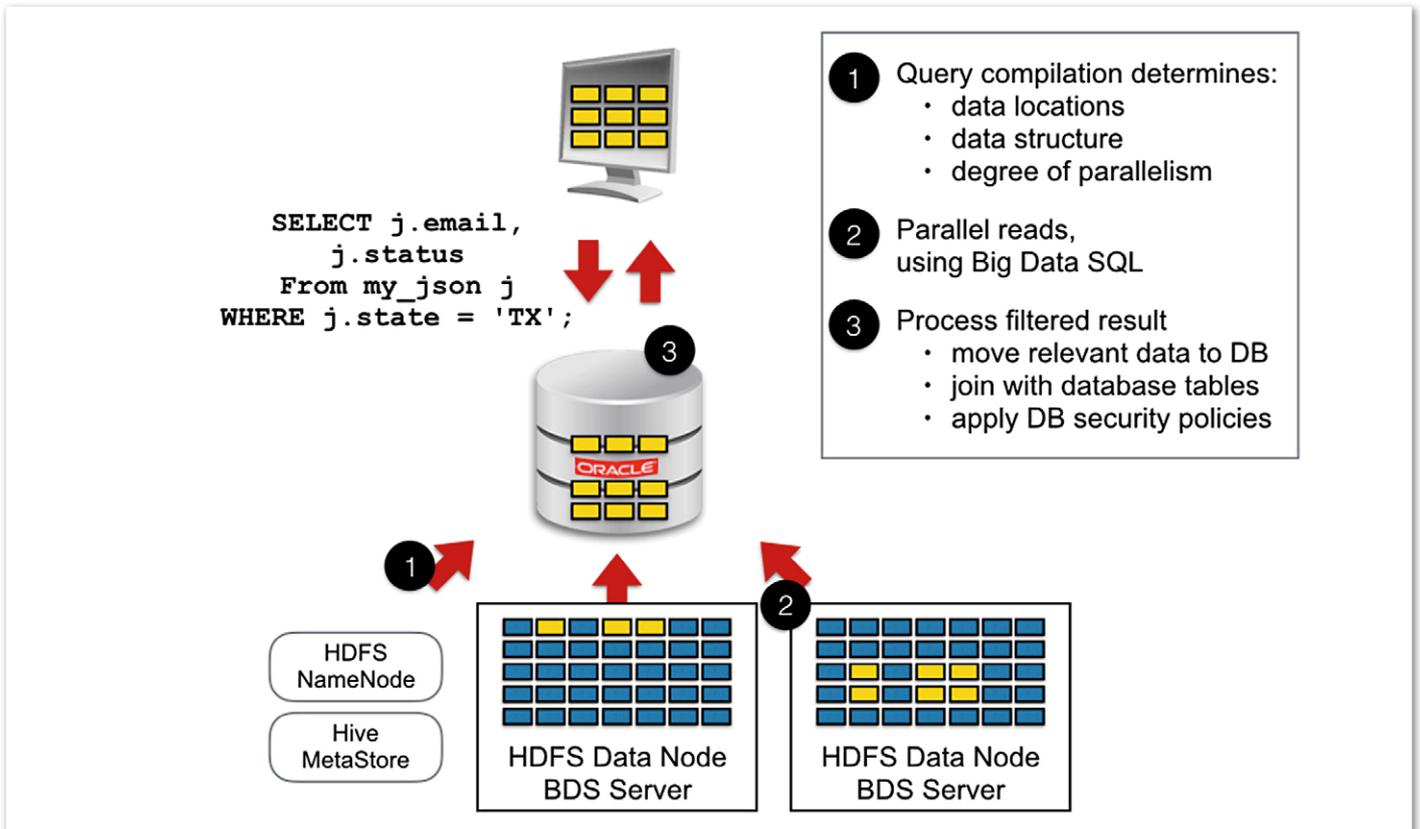


Abbildung 3: Ablauf der Datenverarbeitung mit Big Data SQL

- Verarbeitung in 2 Phasen und 6 Schritten

- Phase 1 - Vorbereitung

1. Dimensionen scannen
2. Schlüsselvektoren bilden
3. Akkumulator vorbereiten
4. Tmp-Tabellen für Dim-Attribute bilden

- Phase 2 - Berechnung

5. Fakten bzgl. Schlüsselvektoren scannen
6. Gefilterte Fakten mit TMP-Tabellen joinen

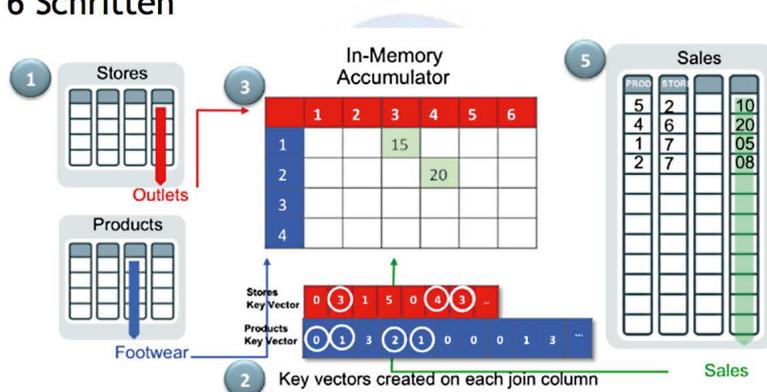


Abbildung 4: Ablauf einer In-Memory-Aggregation im Columnar Store

## Big Data SQL

Während Exadata Storage erfolgreich Performance-Probleme bei der Nutzung von Oracle-Datenbanken adressiert, bleibt das Problem des schnellen, kostengünstigen Zugriffs auf neue Daten von dieser Technologie unberührt. Für die schnelle Analyse großer Mengen rapide eintreffender Daten, die nicht aufwändig vorverarbeitet werden sollen, hat sich innerhalb der letzten Jahre die Big-Data-Technologie etabliert. Diese kombiniert einen redundanten, verteilten Dateispeicher mit verteilter Datenverarbeitung auf kostengünstigen Rechnerknoten. Die Technologie ist sehr gut für die dauerhafte Speicherung und Verwaltung auch massivster Datenmengen geeignet.

Auf dem Hadoop Distributed File System (HDFS) und auf No-SQL-Datenbanken wie HBASE gab es zur Daten-Verarbeitung und -Abfrage zunächst lediglich die batch-orientierte Verarbeitungsmethode des Java-basierten „MapReduce“. Mittlerweile ist die ganze Community dabei, typische Eigenschaften der relationalen Datenbank-Technologie wie die synchrone Datenabfrage mit Impala und den programmatischen Zugriff mit Spark zu entwickeln.

Ein Big-Data-System zu administrieren, bleibt jedoch aufwändig. Es ist gegenüber einer relationalen Datenbank sehr langsam (keine optimierten Datenformate), beim Datenzugriff recht ineffizient (keine Indizes und andere Optimierungen out of the box) und benötigt einen ganzen Zoo von Werkzeugen, um verschiedene Aspekte wie die Sicherheit zu bearbeiten.

Schließlich ist die Verbindung von Daten eines Big-Data-Systems mit anderen Datenquellen zum Zwecke der Auswertung eine Herausforderung.

Oracle hat sich all dieser Probleme angenommen. Das Angebot wuchs in den letzten Jahren mit der Entwicklung des Hadoop-Ökosystems mit. Zunächst gab es lediglich die Möglichkeit, mithilfe einer Erweiterung der „External Table“-Technologie Hive-Table-Definitionen zu nutzen und so direkt auf Dateien und Daten eines Hadoop-Clusters zuzugreifen. Damit konnten bereits Daten innerhalb der Oracle-Datenbank mit denen eines Hadoop-Clusters verbunden werden. Allerdings mussten bei jeder Anfrage erneut alle Daten einer Hive-Tabelle, also die Daten aller betroffenen Dateien, temporär in das RDBMS geladen werden. Erst dort erfolgten Filterung, Projektion etc.

Die Technologie hat also dieselben Nachteile wie die alte SGA/PGA/Storage-Architektur der Datenbank, plus die wesentlich langsamere Zugriffsgeschwindigkeit eines Hadoop-Clusters, da hier die Daten ja nicht schon in effizient zu lesenden DB-Blöcken und Zeilenstrukturen vorliegen, sondern erst beim Lesezugriff strukturiert werden („Schema on Read“).

Der nächste Entwicklungsschritt bestand im Anbieten einer speziellen Big Data Appliance (BDA), um den Kunden den Engineering-Aufwand abzunehmen, administrative Tätigkeiten zu vereinfachen und gute Cluster-Performance zu garantieren.

Der letzte und wichtigste Schritt jedoch war die Einführung der „Big Data SQL“-

Software (BDS). Diese ist eine Adaption der Exadata-Storage-Software, die analog zu anderen Hadoop-Erweiterungen wie Spark oder Yarn auf allen Clusterknoten eines Hadoop-Clusters installiert ist. Wie in den Exadata-Systemen bietet BDS einer Oracle-Datenbank nicht nur Services zum Zugriff auf Daten, sondern auch zur Vorverarbeitung derselben an. Hadoop-Daten aus Dateien im HDFS und aus No-SQL-Datenbanken werden dabei weiterhin über den „External Table“-Mechanismus der Datenbank deklariert. Danach kann transparent per SQL-Abfrage in der Datenbank auf die Hadoop-Daten zugegriffen werden.

In der gegenwärtigen Ausbaustufe bietet BDS noch nicht den gesamten Funktionsumfang, wie er auf Exadata-Storage-Zellen zur Verfügung steht. Die wichtigsten Performance-Beschleuniger zum Smart Scanning, nämlich Filterung, Projektion, Scoring von Modellen etc., stehen jedoch bereits zur Verfügung. *Abbildung 3* zeigt die prinzipielle Ablauf-Architektur in Big Data SQL. Der Zugriff mittels Big Data SQL bietet:

- Beste Performance auf Hadoop
- SQL-Verarbeitung auf beliebigen Daten
- Datenintegration von Daten innerhalb der Oracle-Datenbank mit beliebigen Hadoop-Quellen
- Volle Nutzung der Oracle-Sicherheitsfeatures (VPD, Redaction etc.)

Mit BDS hat Oracle also auch für die Hadoop-Welt Verarbeitungsfunktionalität in

den Storage-Layer verschoben. Die Query-Engine kann Teile ihrer Arbeit an den Storage-Layer delegieren. Wertvoller Prozessspeicher sowie die Datenbank-Server-CPU's werden entlastet und weniger Daten belasten die Netzwerkkapazität. Für den Anwender ergibt sich daraus eine extrem verbesserte Nutzung von Ressourcen, bei gleichzeitiger Erhöhung der applikativen Sicherheit. Außerdem resultiert eine massive Erweiterung der applikatorischen Möglichkeiten durch einfachere Datenintegration.

### In-Memory Columnar Store

Mit Big Data SQL hat Oracle die Welt der Hadoop-Cluster für die Oracle-Datenbank erschlossen. Gleichzeitig entwickelte sich aber an anderer Stelle, nämlich der analytischen High-End-Performance, ein interessanter Wettbewerb der Hersteller und Technologien. Dieser Wettbewerb führte zu einer Renaissance spaltenbasierter arbeitender Datenbanken.

Der Vorteil spaltenbasierter Datenbanken erschließt sich bei der Anwendung für analytische Queries sofort. Da analytische Abfragen in der Regel wenige Spalten einer Tabelle nutzen, dafür aber eine sehr große Anzahl von Zeilen, ist das Scanning für analytische Queries in spaltenorientierten Datenstrukturen wesentlich schneller. Dies gilt prinzipiell für Implementierungen aller Hersteller, seien es SAP HANA, Exasol oder andere. Wie leistungsfähig ein Columnar Store im Vergleich zum Wettbewerb bei gleichen Ab-

fragen und vergleichbarer Infrastruktur ist, hängt daher hauptsächlich davon ab

- welche weiteren Technologien neben dem einfachen Scanning zur Verfügung stehen
- wie der Columnar Store ins Gesamtsystem eingebettet ist

In-Memory Columnar Store besitzt, neben dem einfachen Scanning, eine Vielzahl zusätzlicher Features, die die Performance erhöhen. Beispiele hierfür sind SIMD-Vektorverarbeitung und das Testen von Filterbedingungen direkt auf binär komprimierten Daten. Entscheidender jedoch ist, dass auch der In-Memory Columnar Store lokale Funktionalität zur autonomen Bearbeitung von Teilabfragen besitzt. Diese sind unter anderem:

- In-Memory Storage Indices
- Filterung (Projektion ist in spaltenorientierten Strukturen nicht notwendig, da inhärent)
- Join-Bearbeitung mittels Bloom-Filtering
- In-Memory-Aggregation

Die Query-Engine der Datenbank kann also vom In-Memory Columnar Store nicht nur Blöcke anfordern, wie es beim klassischen Block Buffer der SGA der Fall ist, oder Teile eines Datenbank-Files wie beim klassischen Platten- oder Flashspeicher. Stattdessen ist die Situation ähnlich wie bei Exadata Storage und Big Data SQL. Die Query-Engine

kann Teilaufgaben direkt an den spaltenorientierten Speicher vergeben. Dieser löst die Aufgaben selbstständig sowie höchst effizient und übergibt danach die Ergebnisse zur weiteren Verarbeitung in der PGA an den anfragenden Serverprozess.

Teilaufgaben können dabei von großer Komplexität sein, zum Beispiel In-Memory-Aggregationen. Diese berechnen Ergebnisse komplexer dimensionaler Queries in einem Bruchteil der Zeit, die bislang bei der Verwendung zeilenorientierter Star-Schemata gebraucht wurde. Dies ist zudem möglich, ohne dass großer Aufwand für die Modellierung und Wartung unterstützender Strukturen (etwa Bitmap-Indizes) anfällt. *Abbildung 4* zeigt das Verarbeitungsschema einer In-Memory-Aggregation, die innerhalb des In-Memory Columnar Store durchgeführt wird.

### Oracle-Datenbank als Schaltzentrale

In den vorigen Abschnitten wurden drei Speicher-Technologien beziehungsweise deren Erweiterungen vorgestellt, um die Oracle das Portfolio im Verlaufe der vergangenen Jahre ergänzt hat. Jede dieser Technologien für sich löst Performance-Probleme, erschließt neue Möglichkeiten und – im Falle von Big Data SQL – auch ganz neue Daten-Formen und -Mengen.

Für Anwender richtig wertvoll werden diese Technologien aber erst durch die Tatsache, dass alle ihre Eigenschaften zentral und transparent aus dem Oracle-Datenbanksystem genutzt werden können,

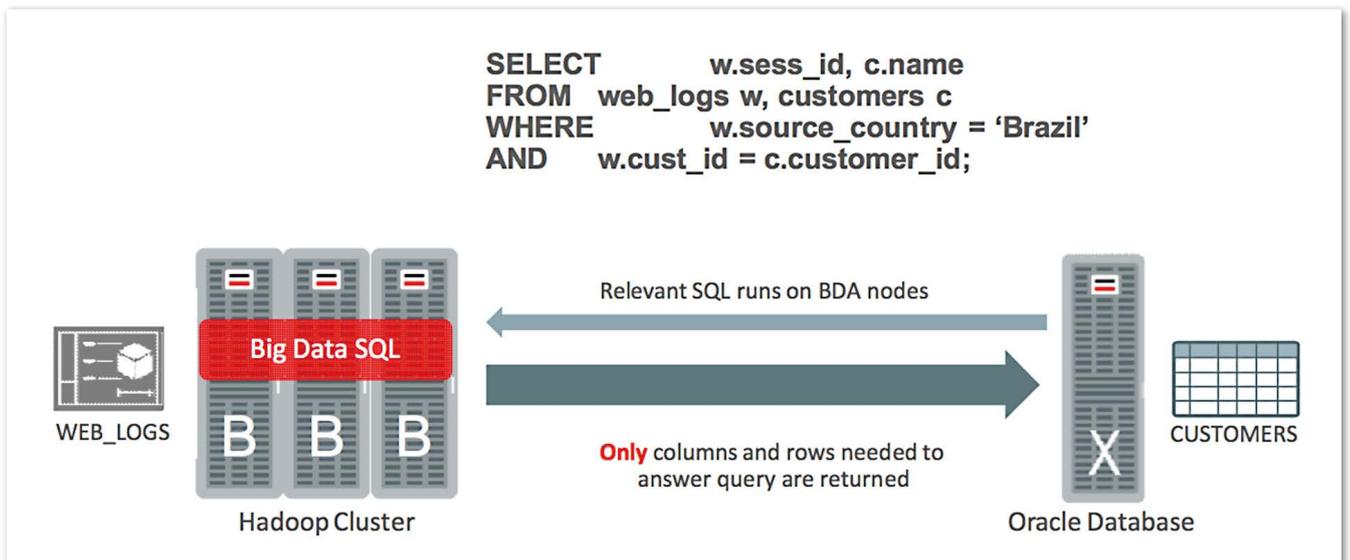


Abbildung 5: Integration von Hadoop- und Exadata-Daten mittels Big Data SQL

also die Nutzung der Kombination von Query-Offloading, dynamischer Quell-Integration, In-Memory-Performance und aller weiteren Eigenschaften ohne Änderung auch nur einer Zeile applikatorischen Codes möglich ist. Basis hierfür sind folgende Änderungen am Datenbankserver:

- Repräsentation aller von den Speichern verwalteten Strukturen im Data Dictionary der Datenbank
- Erweiterung von Query-Transformer, Cost-Based-Optimizer und Plan-Generator dahingehend, dass sie Pläne mit Nutzung der Eigenschaften der neuen Speicher-Technologien generieren und deren Kosten schätzen können
- Erweiterung der Execution-Engine um die Nutzung der neuen APIs

Die Erweiterungen erlauben es der Datenbank, bislang nicht mögliche Dinge zu tun wie:

- Transparente Abfragen über eine Struktur, deren Teile in mehreren verschiedenen Speicher-Technologien vorliegen. Beispielsweise liegen mehrere Spalten einer Struktur im In-Memory-Columnar-Speicher, andere als Zeilen im Exadata-Storage oder im Block Buffer vor. Oder Partitionen mit neusten Daten befinden sich im Columnar Store, etwas ältere im Exadata Storage Layer im Flash, noch ältere im Exadata Storage Layer auf den Festplatten und sehr alte Partitionen auf gewöhnlichem Plattenspeicher.
- Abfragen über mehrere Strukturen, die in verschiedenen Technologien liegen. Ein Beispiel ist der Join von Kundendaten, die im Speicher der Datenbank liegen, und Call-Data-Records auf diejenigen, die auf einem Hadoop-Cluster gespeichert sind. *Abbildung 5* zeigt einen solchen Anwendungsfall.
- Erweiterung und Anwendung des Sicherheitsmodells und der Policies der Datenbank auf einen Hadoop-Cluster. So können Daten aus der Big-Data-Welt in Datenbank-Applikationen eingebunden werden, ohne dass zusätzliche Fähigkeiten geschult werden müssen und administrative Aufwände entstehen.
- Ausführung von Abfragen zwischen Datenbank- und Big-Data-Welt, die bislang aufgrund von Sicherheitsanforderungen unmöglich waren. Sollen zum Beispiel

Transaktionen aus dem Hadoop-Cluster zu einem Kunden übertragen werden, die nur über die Kreditkartennummer verbunden sind, wobei diese jedoch aus Sicherheitsgründen nicht dargestellt werden darf, ist dies mit traditionellen Mitteln (Sicherheitsmanagement auf Hadoop und der DB getrennt) kaum möglich. Mit Big Data SQL kann die Datenbank den Join durchführen und erst danach im Result-Set die Join-Spalte unterdrücken oder per Redaction maskieren.

### Fazit

Ein Nachteil der schönen neuen Welt der verschiedenen Speicher-Technologien und – wie Oracle es nennt – des Query Franchising ist, dass für den Einsatz der oben diskutierten Technologien bestimmte Infrastruktur-Voraussetzungen gelten:

- In-Memory Columnar Store ist ab Datenbank-Release 12.1.0.2 in der Enterprise Edition der Datenbank als lizenzierbare Option enthalten
- Exadata Storage gibt es ausschließlich in Exadata und Super Cluster Appliances
- Big Data SQL erfordert die Verwendung einer Big Data Appliance. Zudem muss der zugreifende Datenbankserver, zumindest zum gegenwärtigen Zeitpunkt, auf einer Exadata Appliance laufen

Sind diese Voraussetzungen jedoch erfüllt, bekommt der Anwender durch Kombination der Speicher-Technologien eine Vergrößerung der Performance, der Möglichkeiten und der Freiheitsgrade bei der Applikations-Entwicklung und -Nutzung, wie sie bislang nicht möglich gewesen ist. Dies ist ein Punkt, an dem sich Oracle eindeutig und auch weit vom Feld der Wettbewerber sowohl in der Datenbank-, als auch in der Big-Data-Welt abgesetzt hat.



Dr.-Ing. Holger Friedrich  
holger.friedrich@sumit.ch

## Sparen Sie Zeit, Geld und Nerven.



Effizient und preiswert:  
DBConcepts.

Wir unterstützen Sie remote beim Betrieb von Oracle Datenbanken.

SLA ab 10hx5 bis 24hx7 inklusive

- proaktiver Überwachung
- rascher Reaktionszeit
- periodische Health Checks
- Backup und Recovery Tests



Die Oracle Experten

[www.dbconcepts.at](http://www.dbconcepts.at)  
Tel.: +43 1 890 89 990  
[office@dbconcepts.at](mailto:office@dbconcepts.at)

**ORACLE** Platinum Partner

# Social-Media-Auswertungen im Data Warehouse

Martin Frisch, OPITZ CONSULTING Deutschland GmbH

Big-Data-Ansätze ermöglichen es nicht nur, immer größere Datenmengen zu verarbeiten, sondern auch, neue Arten von Datenquellen für Auswertungen und Analysen heranzuziehen. Insbesondere Web-2.0-Inhalte, die Nutzer sozialer Netzwerke erstellen, bieten vielfältige Potenziale.

Dieser Artikel beschreibt ein Fallbeispiel aus einer Master-Thesis, in dem Kommentare aus Twitter zur Fernsehserie „Tatort“ für die Erweiterung eines Data Warehouse (DWH) verwendet wurden. Das Beispiel zeigt, wie Big-Data-Komponenten ein auf Oracle basierendes DWH ergänzen können, um Kommentare aus Twitter zu extrahieren, mit einer Sentiment-Analyse zu bewerten, im DWH bereitzustellen – und schließlich in Berichten aufzubereiten. Für die Ausgangssituation wurde zunächst ein Business-Intelligence-System mit einem klassischen DWH für die Tatort-Analyse aufgebaut. Dieses umfasst die Bereiche Datenquelle, Datensicherung, Verarbeitung und Auswertung (siehe Abbildung 1).

Als Datenquelle liegen Stammdaten zu den Ermittlern, den Teams und den Episoden des Tatorts sowie die absoluten und prozentualen Einschaltquoten in

CSV-Dateien vor. Der Bereich „Datensicherung“ entspricht einem DWH mit den drei Schichten „Stage“, „Core“ und „Mart“. Diese befinden sich in einer Oracle-12c-Datenbank. Die ETL-Verarbeitung geschieht mit dem Oracle Data Integrator (ODI), der die Daten aus den CSV-Dateien in die Stage-Schicht extrahiert. Von dort integrieren die Ladeprozesse die Daten in den Core, der nah an der dritten Normalform modelliert ist. Der Mart entspricht dagegen einem Starschema. Auf diesem setzt ein Dashboard Reporting mit Oracle Answers auf, einem Werkzeug der Oracle Business Intelligence Suite (Version 11g). In den Berichten können Anwender die Einschaltquoten nach folgenden Kriterien auswerten:

- Datum
- Episode

- Spielort
- Ermittler

Damit der Erfolg der Episoden nicht nur an den Einschaltquoten messbar ist, wurde das DWH um Kennzahlen erweitert, die die Stimmung der Twitter-Gemeinde zu den Tatort-Episoden oder bestimmten Komponenten und Eigenschaften wie Ermittlern, Teams, Handlung, Musik etc. widerspiegeln. Dabei war zu berücksichtigen, dass Daten aus sozialen Netzwerken im Vergleich zu Daten aus klassischen Quellen wie zum Beispiel einem operativen System einige Besonderheiten aufweisen. So sind Social-Media-Daten weit aus weniger strukturiert. Zudem ist eine auswertbare Größe zur Stimmung eines Tweets in diesem nicht direkt enthalten, sondern muss mit Methoden des Text-Mining aus dem jeweiligen Text heraus er-

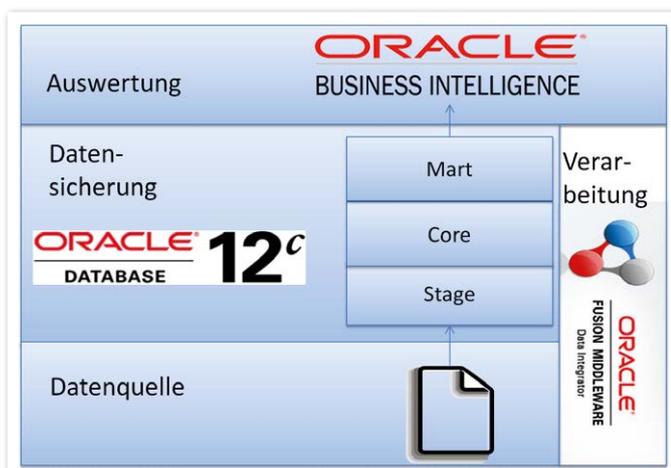


Abbildung 1: Klassisches Data Warehouse

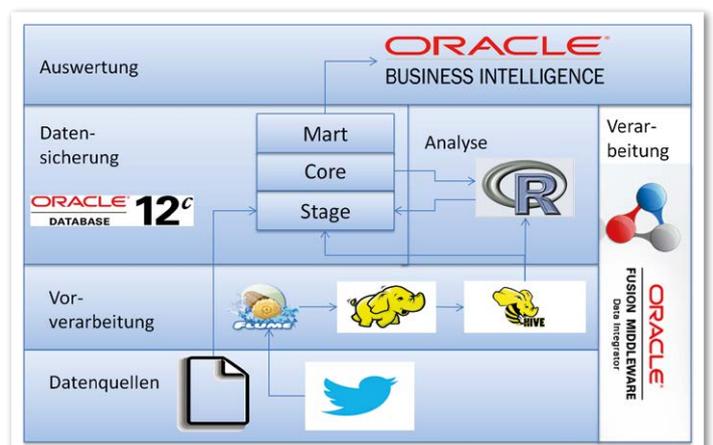


Abbildung 2: Erweitertes Data Warehouse

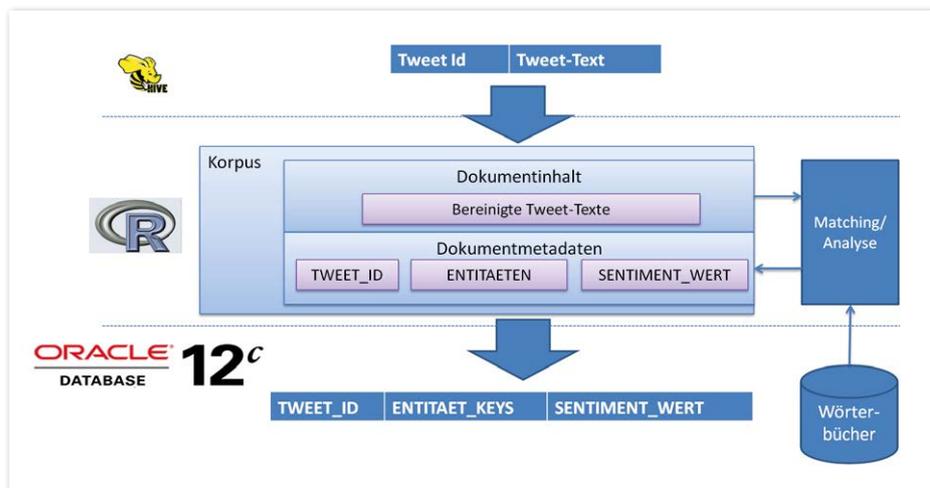


Abbildung 3: Analyse in R

mittelt werden. Für diese Aufgaben sind spezielle Werkzeuge erforderlich.

### Erweiterungen für die Social-Media-Analyse

Die bisherige Architektur wurde um einen Vorverarbeitungs- und einen Analyse-Bereich erweitert (siehe Abbildung 2). Zur Vorverarbeitung kommen Werkzeuge des Hadoop-Rahmenwerks zum Einsatz. Dessen Cloudera-Distribution steht auch auf der Oracle Big Data Appliance zur Verfügung. Mit diesen Tools können die relevanten Twitter-Daten extrahiert und in eine relationale Form überführt werden. Die Tweets werden mithilfe des Twitter-Streaming-API während der Episoden-Ausstrahlung mitgeschnitten.

Die Extraktion des Twitter-Streams geschieht mit Flume, einem Werkzeug, mit dem sich große Datenmengen bewegen lassen und das Streaming-Daten in Echtzeit verarbeiten kann. Dazu benötigt Flume die Konfiguration eines sogenannten „Agenten“, der aus Quelle („Source“), Zwischenspeicher und Ziel („Sink“) besteht. Als Quelle dient ein Java-Programm aus dem Cloudera GitHub [1], um das Streaming-API von Twitter als benutzerdefinierte Quelle verwenden und dafür die öffentliche Java-Bibliothek „Twitter4J“ nutzen zu können [2].

Zur Identifikation der relevanten Tweets wird das Hashtag „#tatort“ als Such-Schlüssel mitgegeben. Als Zwischenspeicher reicht der interne Speicher von Flume aus, da dort keine Transformationen stattfinden. Der Agent verwendet das Hadoop-Filesystem (HDFS) als Ziel. Dort wird zunächst das JSON-Format beibehalten, in

dem Twitter neben den Tweet-Texten noch zahlreiche weitere Attribute wie Zeitpunkt, Ort und Autor der Veröffentlichung liefert. Der Speicherpfad ist dynamisch, sodass Flume für jede Episoden-Ausstrahlung einen eigenen Ordner anlegt.

Hive kann in Kombination mit einem Serializer/Deserializer („SerDe“) unterschiedlichste (semi-strukturierte) Formate wie JSON in eine relationale Struktur überführen und mit der SQL-ähnlichen Abfragesprache HiveQL selektieren. Ein fertiger „SerDe“ für das JSON-Format findet sich beispielsweise auch im Cloudera GitHub. Mit diesem wurde eine nach Episoden partitionierte „External Table“ angelegt, die eine Tabellen-Struktur über die JSON-Daten im HDFS legt. Jede Partition zeigt auf genau einen durch Flume angelegten Ordner. Im ODI sind Befehle zum Anfügen und Löschen von Partitionen in Prozeduren gekapselt.

Da immer eine Episode in einem Batch verarbeitet wird, eignet sich die Partitionierung hier auch zum dynamischen

Filtern der Daten. Die Ladeprozesse bestimmen zur Laufzeit das Datum der zu verarbeitenden Episode und legen zu Beginn der Verarbeitung eine neue Partition an. Nach erfolgreicher Verarbeitung löschen die Ladeprozesse diese Partition wieder, wobei die Daten physisch im HDFS erhalten bleiben. Somit verweist die Tabelle während der Verarbeitung nur auf Daten zur aktuellen Episode und muss nicht mehr gefiltert werden. Vor und nach der Verarbeitung ist die Tabelle leer.

Zu diesem Zeitpunkt liegen die Twitter-Daten in einer Tabellen-Struktur vor, die aber noch stark denormalisiert ist. Sie enthält Redundanzen und nicht-atomare Attribute (wie Arrays oder zusammengesetzte Datentypen). Um diese aufzulösen, finden bereits in Hive erste Aufbereitungen mithilfe von vier Views statt, die die Informationen nach den folgenden Entitäten gliedern:

- Tweet(-Text)
- Twitter-Nutzer
- Ort
- Anzahl der Retweets

Informationen, die später im DWH und in den Berichten nur als Dimension oder Detail-Information Verwendung finden, werden direkt in den Stage-Bereich des DWH geladen. Dazu bietet der ODI einen entsprechenden Hive-Konnektor. Im Mart wird mithilfe dieser Daten eine (faktenlose) Fakten-Tabelle zu den Tweets und jeweils eine eigene Dimension für Twitter-Nutzer und Twitter-Orte bereitgestellt.

Eine besondere Herausforderung stellte die Harmonisierung der Ortsinformationen dar. Häufig sind diese leer oder enthalten unterschiedliche Granularität

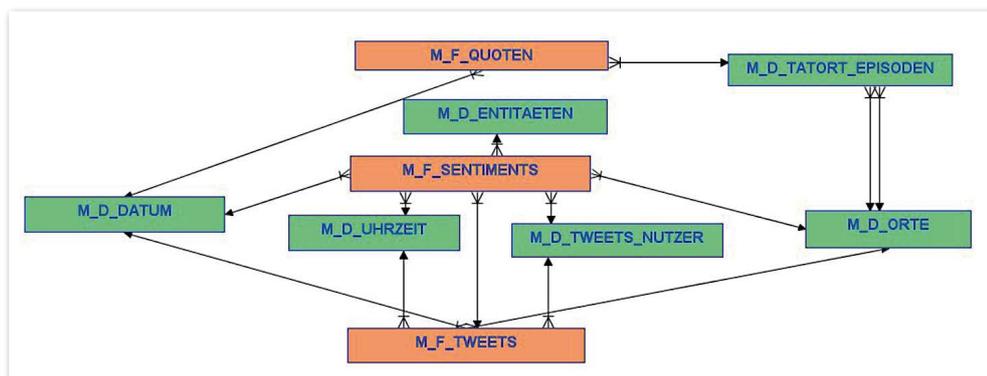


Abbildung 4: Das Mart-Datenmodell

ten, Sprachen und Schreibweisen für den gleichen Ort. Da die Ortskoordinaten vom Twitter-API mitgeliefert werden, konnte dieses Problem mit Referenzdaten gelöst werden, die ebenfalls über Ortskoordinaten verfügen.

### Analyse in R

Die Ermittlung der eigentlichen Stimmung geschieht im Analyse-Bereich mit der Oracle-Distribution des Statistik-Werkzeugs R (Version 3.01). Diese Distribution verfügt über Konnektoren, um auf Hive zuzugreifen und Daten mit der Oracle-Datenbank auszutauschen. R importiert, bereinigt und analysiert die Tweet-Texte aus Hive und sichert die Ergebnisse in der Oracle Datenbank (siehe *Abbildung 3*).

Neben den reinen Tweet-Texten importiert R auch die zugehörigen Tweet-IDs aus Hive. Die Texte werden als Dokumente in einem sogenannten „Korpus“ vorgehalten. Diese speziell für Dokumente ausgelegte Speicherstruktur erlaubt es, den eigentlichen Inhalten auch Metadaten zuzuordnen. Der Korpus für die Tatort-Tweets speichert die IDs und die Ergebnisse der Analyse als solche Metadaten. Die Analyse selbst erfolgt in drei Schritten:

1. Bereinigung der Tweet-Texte. Dabei werden insbesondere überflüssige und störende Textteile wie Stoppwörter, Sonderzeichen, Zahlen oder Leerzeichen entfernt. Zudem werden alle Buchstaben kleingeschrieben, Umlaute aufgelöst und Wortstämme gebildet.
2. Für jeden Tweet wird mit einem lexikonbasierten Ansatz der Sentiment-Analyse ein Stimmungswert ermittelt.
3. Jedem Tweet werden Entitäten zugeordnet, auf die sich der Tweet bezieht, etwa ein bestimmter Tatort-Ermittler.

Sowohl der lexikonbasierte Ansatz der Sentiment-Analyse als auch die Zuordnung der Entitäten verwenden Wörterbücher. Da R selbst über keinen persistenten Speicher verfügt, verwaltet die Oracle-Datenbank diese Wörterbücher und stellt sie in Schnittstellen-Views für R zur Verfügung.

Das Wörterbuch für die Sentiment-Analyse enthält Phrasen, die eine bestimmte Stimmung ausdrücken. Diese Stimmung wird im Wörterbuch durch eine Zahl zwischen „-1“ (sehr negativ) und „1“ (sehr positiv) ausgedrückt. Während der Analyse wer-

den diese Werte für die in einem Tweet-Text gefundenen Phrasen zusammengerechnet.

Im Wörterbuch für die Entitäten-Zuordnung befinden sich hingegen Kontext-Informationen und Eigenschaften, die im Wesentlichen auf den Stammdaten aus dem DWH basieren. Das sind in diesem Fall zum Beispiel Namen der Ermittler, der Spielorte oder der Episoden. Darüber hinaus besteht die Möglichkeit, zusätzliche Begriffe manuell zu konfigurieren.

Exemplarisch wurden einige Begriffe, die Charakteristika einer Krimiserie wie Handlung, Schauspieler oder anderes beschreiben, eingepflegt. Findet die Analyse einen Begriff aus diesem Wörterbuch in einem Tweet-Text, speichert sie in den Korpus-Metadaten für dieses Dokument einen eindeutigen Entitäten-Schlüssel zu diesem Begriff, der ebenfalls aus dem Wörterbuch stammt.

Nach der Analyse schreibt R die Ergebnisse in den Stage-Bereich des Tatort-DWH. Die gesamte R-Verarbeitung ist in kaskadierenden Skripten umgesetzt, die über einen einzigen Betriebssystembefehl vom ODI aufgerufen werden. Oracle R Enterprise kann die Skripte in der Oracle-Datenbank verwalten und über entsprechende SQL-Befehle starten, was in einem produktiven System mit vielen Skripten eine sinnvolle Alternative sein kann.

Nachdem sich die Ergebnisse der Analyse im DWH befinden, werden sie über die Tweet-ID den Twitter-Daten zugeordnet, die direkt von Hive in das DWH geladen wurden. Gleiches geschieht mit den Entitäten und den Stammdaten anhand der Entitäten-Schlüssel. Da die Stimmung der Tweets im Sentiment-Wert quantifiziert ist, kann sie im DWH wie klassische Quelldaten verarbeitet und aufbereitet werden.

Im Mart werden zusätzlich verschiedene Kennzahlen auf Basis des Sentiment-Wertes aus der Analyse berechnet. Beispielsweise wird der Sentiment-Wert auf einer Skala von „-1“ bis „1“ normalisiert. Durch eine Aggregation im Frontend kann daraus dann im Anschluss ein Durchschnittswert berechnet werden, der die Gesamtstimmung repräsentiert. Zudem werden die Tweets mit Flags versehen, die zeigen, ob der Tweet aufgrund des Stimmungswertes als positiv, neutral oder negativ zu bewerten ist.

Dank dieser Kennzeichnung kann in Berichten leicht die Kompletanzahl oder

auch der Anteil positiver oder negativer Tweets ermittelt werden. *Abbildung 4* zeigt das komplette Datenmodell der Mart-Schicht, das aus drei Faktentabellen (rot) und sechs Dimensionstabellen (grün) besteht.

### Fazit

Die Tatort-Analyse zeigt, wie durch das Anfügen eines Vorverarbeitungs- und Analyse-Bereichs auch Daten aus sozialen Netzen in einem DWH ausgewertet werden können. Oracle-Datenbank, ODI, Hadoop und R harmonieren dabei sehr gut miteinander. So profitiert beispielsweise die Analyse vom DWH als persistenter Speicherkomponente und Informationslieferant. Von dort können relevante Kontext-Informationen zur Domäne gefunden werden, was im Beispiel die Entitäten sind. Zudem können hier Konfigurationsdaten für die Analyse (etwa Wörterbücher) verwaltet werden.

Für die Umsetzung muss das DWH nicht umgebaut werden, sondern es reicht aus, es um einige Komponenten zu erweitern, die Spezialaufgaben beim Umgang mit Social-Media-Daten übernehmen. Der ODI kann weiterhin zur Steuerung und Überwachung der Gesamtverarbeitung eingesetzt werden. Gleiches gilt für die Oracle BI Suite als Frontend.

Durch das Beibehalten dieser Werkzeuge profitieren die Mitarbeiter eines Unternehmens weiterhin von ihren Kenntnissen in den bereits für sie bekannten Technologien. Dadurch haben sie weniger Berührungspunkte mit den neuen Datenquellen und hinzugekommenen Technologien.

### Quellen

[1] <https://github.com/cloudera/cdh-twitter-example>

[2] <http://twitter4j.org/en/index.html>



Martin Frisch

[martin.frisch@opitz-consulting.com](mailto:martin.frisch@opitz-consulting.com)

# Hadoop-Einstiegshürden meistern – mit Business Case, Skill-Aufbau und der richtigen Technologie zu Big Data

Oliver Herzberg und Slavomir Nagy, metafinanz Informationssysteme GmbH

Für einen kosteneffizienten Einstieg in Big Data verspricht Hadoop mehr Analyse-Potenzial als klassisches DWH. Doch die florierende Open-Source-Technik erfordert ein strategisches und konzeptionelles Umdenken. Der Artikel zeigt, worauf es beim Finden der Business Cases, dem Kompetenz-Aufbau und den Test-Umgebungen in der Cloud ankommt.

Auch wenn Big Data lange als ein etwas nebulöser Begriff galt, so sind zumindest die Herausforderungen für die Geschäftswelt offensichtlich. Unternehmen erhalten immer mehr Daten von Menschen und in letzter Zeit auch vermehrt von Maschinen; für die Analyse dieser Daten sind neue, mächtigere Technologien erforderlich. In diesem neu zu erschließenden Gebiet wurden inzwischen deutliche Schneisen geschlagen. Als wichtigste technische Plattform zur Bewältigung einer zukunftsorientierten Datenbewirtschaftung gilt in IT-Fachkreisen das Framework Hadoop.

Doch dieser Einigkeit zum Trotz wirft die Technologie nach wie vor Fragezeichen auf. Zum einen sind es die ständig neu entstehenden technischen Standards (siehe Kasten „Was ist eigentlich Hadoop?“), die einen klaren Überblick erschweren, zum anderen müssen Einsatzszenarien wohlüberlegt sein, um überzeugende geschäftliche Mehrwerte in BI-gesättigten Umgebungen zu entwickeln. Immerhin kann inzwischen schon eine Reihe von Unternehmen aus unterschiedlichen Branchen mit überzeugenden Anwendungsbeispielen aufwarten, die sich auf Gebieten wie der Betrugserkennung, der Kundenabwanderungs-Analyse oder der Fahrzeugdaten-Auswertung abspielen.

## **BARC-Studie beleuchtet Status quo**

Die Aussicht auf völlig neue Möglichkeiten der Datenauswertung reicht allein nicht

aus, um in Unternehmen den Einsatz einer neuen Technologie zu rechtfertigen. Das belegt auch die aktuelle Studie des Marktforschungs- und Beratungshauses BARC mit dem Titel „Hadoop als Wegbereiter für Analytics“. Bei vier Prozent der 261 Teilnehmer aus der DACH-Region ist Hadoop bereits Bestandteil der Unternehmensprozesse, weitere 14 Prozent betreiben ein Pilotprojekt und elf Prozent hegen Pläne dafür. 49 Prozent können sich ein Engagement immerhin vorstellen. Der Rest, mit 22 Prozent rund ein Fünftel der

Befragten, sieht noch überhaupt keinen Handlungsbedarf.

Bemerkenswert sind auch die Erkenntnisse darüber, wer in den Unternehmen die Treiber von Hadoop sind. Mit einem Anteil von 54 Prozent sticht hier die IT-Abteilung hervor, BI-Verantwortliche wurden bei 42 Prozent der Befragten als Motivator genannt, aus den Fachbereichen kommt in 26 Prozent der Fälle die Initiative. Insgesamt müsse nach den Erkenntnissen der Würzburger Analysten noch viel Überzeugungsarbeit geleistet werden, was den

### **metafinanz Informationssysteme: Navigator für die Big-Data-Welt**

Wie können Unternehmen ihre Daten gewinnbringend nutzen? Welche Technologien eignen sich, welche Kompetenzen sind vonnöten, wo ist der Start-, wo der Endpunkt? Bei all dem Hype, der sich im Moment um das Thema aufgebaut hat, ist es schwierig, einen Überblick zu bekommen. Die Strategie und die Technologie gibt es nicht aus einem Guss, sondern beides muss individuell aus verschiedenen Teilen zusammengesetzt werden. Big Data fordert das Business und die IT gleichermaßen heraus. Beratungsunternehmen wie die metafinanz bringen die Kunden gerade bei

kritischen Themen wie Skill-Entwicklung und effizienzorientierten Einstiegsszenarien sicher ans Ziel. Wir greifen dabei auf einen umfangreichen Erfahrungsschatz zurück, denn wir halten uns kontinuierlich auf dem Laufenden, was die neuen Technologien können, was sie bringen und wo sie sich einsetzen lassen. Gleichzeitig beraten wir unsere Kunden auch im Hinblick auf ihre Geschäftsprozesse, denn diese sind unzertrennlich mit der technologischen Entscheidung verknüpft. Wir unterstützen unsere Kunden dabei, lohnende Analyseprojekte zu ermitteln, Big-Data-Szenarien anhand von Prototypen zu testen und das Alignment zwischen Business und IT zu organisieren.

Mehrwert von Big Data und Analytics betreffe. Denn drei Viertel der befragten Unternehmen stufen Hadoop noch als weniger oder gar nicht wichtiges Thema ein.

### Hadoop adressiert Versäumnisse beim Daten-Management

Womit ein zentraler Punkt genannt wäre: Was sind die überzeugenden Gründe für den Einsatz von Hadoop? Entscheidend dürfte sein, dass die traditionellen Technologien für Daten-Management in den vergangenen drei Jahrzehnten nicht die Erwartungen der Anwender erfüllt haben. Kernsegmente wie BI und Data Warehouse kratzten nur an der Oberfläche und förderten aus den Daten nicht im erwarteten Maße nützliche Informationen und handlungsmotivierende Erkenntnisse zutage. Der Einsatz von Hadoop kann ein Ansatz sein, diese Herausforderungen im klassischen BI-Umfeld zu meistern.

Um es klarzustellen: Es ist nicht gemeint, nun alle historisch gewachsenen Daten-Management-Strukturen über Bord zu werfen. Der parallele Betrieb einer schnellen

und einer langsamen Infrastruktur ist eine sinnvolle Vorgehensweise für den Einstieg in Big Data. Alle strukturierten, unstrukturierten und teilstrukturierten Daten im Unternehmen, die beispielsweise nicht für das Enterprise Data Warehouse geeignet sind, lassen sich komplett in einen Hadoop-basierten Daten-Hub übernehmen. Auf diese Weise werden die Daten für Berichtswesen und Analysen verfügbar – und zwar wahlweise mit bestehenden BI- oder mit neuen Hadoop-Tools.

Nicht übersehen sollten Unternehmen dabei allerdings den Umstand, dass Hadoop völlig andere Techniken erfordert als klassische Business Intelligence, die auf strukturierten, integrierten und bereinigten relationalen Datenbank-Management-Systemen basiert. Vor allem aber braucht der Einstieg in die neue Welt auch ein Business-Szenario.

### Die Technik muss das Business unterstützen

Vor der Einführung neuer Technologien steht der Nachweis der Rentabilität inner-

halb eines bestimmten Zeitraums oder ein darauf aufbauendes Geschäftsmodell. Unternehmen steigen ein, indem sie lohnende Analyse-Projekte ermitteln und Big-Data-Szenarien anhand von Prototypen testen. Und, indem sie einen Transfer zwischen Business und IT organisieren. Denn es geht nicht um die Technik, sondern um die Frage, wie die Technik das Business unterstützen kann.

Zur Ermittlung des Business Case bieten sich diverse Grundüberlegungen zu dem gewünschten Ziel an – selbstverständlich abhängig vom jeweiligen Geschäft; dennoch gibt es einige allgemeine Fragen, die sich jedes Unternehmen stellen sollte: Sollen Kosten gesenkt und die Profitabilität des Geschäfts gesteigert werden? Etwa durch die Beschleunigung bestimmter Datenauswertungen. Sollen neue Formen der Kundenkommunikation gefunden werden? Wie durch die Auswertung von Kundenfeedbacks. Oder sollen Produkte um Services ergänzt werden? Durch die Nutzung von Vorhersagen beispielsweise lassen sich Wartungen proaktiv gestalten

## Welche aktuellen Probleme möchten Sie mit Ihrer Hadoop-Initiative adressieren/können Sie sich vorstellen mit einer Hadoop-Initiative zu adressieren?



Quelle: BARC Survey „Hadoop 2015“, n=171; alle Teilnehmer, die sich eine Hadoop-Initiative mindestens vorstellen können

Abbildung 1: Nehmen Sie sich die Zeit, die Leute und die Sorgfalt, um zu Beginn die richtigen Fragen zu stellen.

- das Ersatzteil ist dann schon beim Kunden, bevor die Maschine den Geist aufgibt. Oder möchte das Unternehmen vielleicht ein völlig neues Geschäft starten? Indem es beispielsweise seine Produkte oder Services stärker individualisiert.

Auch innerhalb der IT-Organisation lassen sich solche Gedanken durchspielen. Wenn die Datenmanagement-Infrastruktur historisch bedingt zu aufwändig in der Pflege und Weiterentwicklung ge-

worden ist, hilft eine agile Technologie sicherlich, das Dilemma zwischen Kostendisziplin und dem Wunsch nach Innovationen aufzulösen. Denn im Gegensatz zu den hohen Aufwänden und langen Zeiträumen, die im Bereich Datenmanagement selbst bei Einstiegsszenarien benötigt werden, lassen sich mit Technologien wie Hadoop in der Regel binnen weniger Wochen oder höchstens Monaten Testszenarien durchspielen.

### Größte Herausforderung ist der Know-how-Aufbau

Ist die Entscheidung für den Schritt in die neue Welt gefallen, wartet schon die nächste Herausforderung: der Aufbau der richtigen Mannschaft. Das richtige Know-how zu entwickeln, ist derzeit eine der größten Hürden, wie auch aus der BARC-Studie hervorgeht. Gefragt nach den Haupthindernissen bei der Einführung von Big Data nannten 68 Prozent der Befragten das fehlende technische und 65 Prozent das fehlende fachliche Know-how.

Derartige Schwierigkeiten sind auch uns aus eigener Erfahrung bekannt. Die Autoren befassen sich schon seit Längerem mit Hadoop und mussten einiges lernen. Nachdem am Markt kaum Experten zu finden waren, stellten sie sich die Frage, welche IT-Expertengruppe die besten Voraussetzungen für die Skill-Entwicklung zum Hadoop-Spezialisten mitbringt. Zur Wahl standen Java- oder Datenbank-Entwickler sowie Linux-Administratoren. Idealerweise sollten die Kandidaten alle drei Bereiche abdecken, aber diese findet man nur selten und wenn, dann sind sie teuer. Die Entscheidung fiel damals auf Datenbank-Entwickler, schließlich brachten sie bereits Wissen im Umgang mit großen Datenmengen beziehungsweise mit dem Management von Daten mit.

Doch schnell machten sich kulturelle Probleme bemerkbar. Richtig ist, dass es um die Verarbeitung von Datenmassen geht. Das kennen und können Datenbank-Entwickler. Dennoch, neu ist für sie die bisher ungewohnte Art zu arbeiten. Hadoop basiert auf einer in Java entwickelten Umgebung. Das Arbeiten ist schneller und nicht so klar abgegrenzt wie mit klassischen DWH- oder BI-Systemen.

Auch ist die Technologie jünger und der Markt sehr dynamisch. Die Open-Source-Community entwickelt sie stetig weiter -

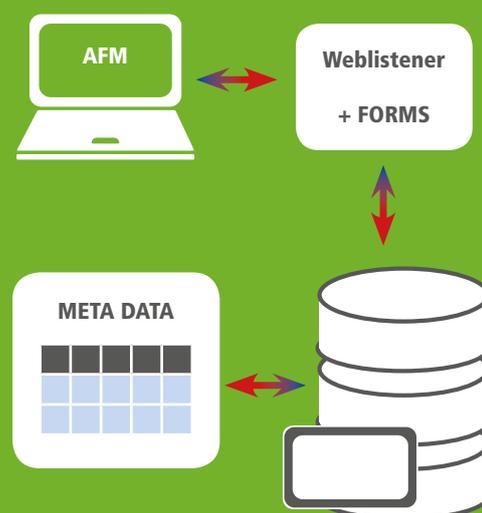
### Traditionelles BI stößt an seine Grenzen

Der Forrester-Analyst Boris Evelson stellt in seinem Blog fest, dass die traditionellen Technologien für Datenmanagement in den letzten drei Jahrzehnten nicht die Erwartungen der Anwender erfüllt haben. Kernsegmente wie BI und Data Warehouse kratzen nur an der Oberfläche und fördern nicht im erwarteten Maße nützliche Informationen und handlungsmotivierende Erkenntnisse aus den Daten zutage. Er nennt dafür folgende Gründe:

- *Unternehmen werten zu wenige Daten aus*  
Unternehmen ziehen nur 40 Prozent ihrer strukturierten Daten für strategische Entscheidungen heran.
- *Unstrukturierte Daten werden lediglich zu einem Drittel in Analysen berücksichtigt*
- *BI ist zu komplex*  
Frühere sowie einige aktuelle Plattformen für Business Intelligence sind zu kompliziert. Sie erfordern die Integration von Dutzenden Komponenten unterschiedlicher Hersteller. Deshalb dauert es so lange und ist so kostspielig, aus der Gesamtheit der Daten eine „single version of truth“ herauszudestillieren.
- *BI-Architekturen sind unflexibel*  
Viele Unternehmen brauchen zu lange, um das Idealziel einer zentralisierten BI-Umgebung zu erreichen. Wenn sie glauben, angekommen zu sein, liegen schon wieder neue Datenquellen vor, neue Regulatoriken oder neue Kundenwünsche.

# APEX-FORMS MASHUP

TOOLS that offer co-existence of your FORMS and Web2 environment



### Single-Sign-On authentication

- Unified user interface for PC and mobile and similar look & feel for FORMS and APEX
- Fit for large, high definition screens and zoom also for FORMS

### Contact

Robert Johannesson,  
roj@softbase.dk

Case story and examples are available

Learn about your new roadmap option

Get first impression at  
[www.softbase.dk](http://www.softbase.dk)

Meet us at DOAG in Nürnberg

Soft **BASE**

ORACLE PARTNER

### Fünf Empfehlungen für den risikofreien, schnellen und günstigen Einstieg in Hadoop

1. Gewinnbringendes Szenario auswählen
2. Die richtigen Menschen finden und ausbilden
3. Großes Augenmerk auf Security legen
4. Die Cloud als Chance für schnelle Resultate nutzen
5. Datenqualität im Auge behalten

und das mit einer hohen Geschwindigkeit. Lehrgänge oder aktuelle Literatur gibt es kaum, die Experten diskutieren ihre Lösungsszenarien vor allem auf Veranstaltungen und in Internetforen.

Hier die richtigen Leute zu finden und Skills aufzubauen, ist daher weniger eine Frage des vorhandenen technischen Know-hows als einer gewissen Mentalität. Ohne Begeisterung geht es nicht. Ohne Beharrlichkeit auch nicht. Denn während klassische Datenbank-Experten gewohnt sind, mit kurzen, klar identifizierbaren Fehlercodes zu arbeiten, müssen Hadoop-Fachleute mit Java-Fehlermeldungen klarkommen, die schon einmal mehrere Bildschirme füllen. Es sind Menschen gefragt, die sich aus eigener Kraft in das Thema einarbeiten, durch Probleme beißen und auch einmal Rückschläge hinnehmen können. Unsere Erfahrung hat außerdem gezeigt, dass die Kollegen ständig am Ball bleiben müssen. In der Welt von Big Data und Hadoop kann der Anschluss schon nach wenigen Monaten verloren gehen.

Auch im Management sind neue Skills gefragt. Der Einstieg in Big Data ist mit Risiken behaftet, die in Kauf genommen werden müssen. Und es gibt keine Standards, auf die man sich verlassen kann. Auch aus Managementsicht ist also Ausdauer gefragt, ebenso wie die Bereitschaft und Fähigkeit, mit Risiken und Unsicherheit umzugehen.

Für Projektmitarbeiter wie für Manager kommt es also weniger auf eine bestimmte fachliche Expertise an, sondern auf die Bereitschaft, ausgetretene Pfade zu verlassen, sich mit Neuem auseinanderzusetzen. Da sich Business und IT in

Einklang setzen müssen, empfehlen wir nicht zuletzt außerdem den Aufbau einer interdisziplinären Mannschaft. Optimal sind Teams mit Menschen, die kommunizieren können, sich mit Datenauswertungen auskennen, das Geschäft verstehen und selbstverständlich auch technisches Verständnis besitzen (*siehe Abbildung 1*).

### Cloud-Umgebungen senken die Einstiegshürden

Treiber der Technologie und damit letztlich des Marktes ist die weltweit verteilte Community. Sie ist es, die diverse Verbesserung und Neuerungen entwickelt. Prominente Unterstützung gibt es unter anderem von Playern wie Yahoo, Facebook oder Google. Es kann jedoch weder von einem Standard die Rede sein, noch ist erkennbar, welche Entwicklungen von

Dauer sein werden. Das bedeutet, dass manche der neuen Ansätze weiterverfolgt werden wie zum Beispiel „Spark“, andere hingegen wieder verschwinden.

Ein schneller Ein- und ein schneller Ausstieg müssen also möglich sein. Daher bieten sich Cloud-Umgebungen an, um beweglich zu bleiben, kostengünstig in Hadoop einzusteigen und mögliche Szenarien auszuprobieren. Amazon etwa stellt hier mit AWS eine inzwischen sehr ausgereifte Virtualisierungsumgebung bereit, die sich für das Erstellen von Machbarkeitsstudien (Proof of Concept) nutzen lässt. Als weiterer Massenanbieter mischt inzwischen auch Microsoft mit Azure HDInsight in der Big-Data-Szene mit. Ein weiterer Player auf dem globalen Markt ist Google. Die Company stellt die für eigene Services genutzten Infrastrukturen wie YouTube Shared auch an-

### Was genau ist Hadoop?

Hadoop gilt derzeit als eine der Kerntechnologien im Big-Data-Umfeld. Doch anders als der Begriff suggeriert, steckt dahinter eine schnell wachsende Vielfalt von Technologien, Projekten und Anbietern, die eine klare Definition nahezu unmöglich macht. Einige Experten wie der Gartner-Analyst Merv Adrian und der Cask-Manager Andreas Neumann haben sich in jüngster Zeit damit befasst, eine grobe Übersicht zu vermitteln.

Als erster Ausgangspunkt dient in solchen Fragen Wikipedia, aber der dortige Eintrag erweist sich als sehr vage: „Apache Hadoop ist ein freies, in Java geschriebenes Framework für skalierbare, verteilt arbeitende Software. Es basiert auf dem MapReduce-Algorithmus von Google sowie auf Vorschlägen des Google-Dateisystems und ermöglicht es, intensive Rechenprozesse mit großen Datenmengen auf Computerclustern durchzuführen“. Laut Neumanns Recherche eignen sich jedoch weder „Open Source“ noch „Java“ zu einer eindeutigen Charakterisierung, weil es hiervon einige Abweichler im Markt gibt. Auch ein Muster an Standardkomponenten für eine Hadoop-Distribution ist

nicht zu erkennen. Am Ende vergleicht er Hadoop mit einem Puzzle aus vielen Technologien: Um es zu lösen, fehlen immer wieder einzelne Teile, andere Teile scheinen zu einem anderen Puzzle zu gehören und die Anbieter im Markt spielen die Rolle, das Puzzle für die Kunden zu lösen.

Adrian verfolgt einen Definitionsansatz, bei dem er untersucht hat, welcher Hadoop-Anbieter welches der vielen Hadoop-bezogenen Projekte unterstützt. Ausgangspunkt sind die drei auf der Apache-Hadoop-Hauptseite genannten Projekte HDFS, YARN und MapReduce. In seiner Matrix mit den relevanten Herstellern ergänzt er die obigen drei mit HBase, Hive, Pig, Spark und Zookeeper, die von allen gelisteten Herstellern unterstützt werden. Es folgen diverse Projekte, die jeweils für Teile der Anbieter relevant sind, und so summiert sich am Ende die Zahl der Hadoop-bezogenen Projekte auf 48. Ein Ende der Fahnenstange ist damit allerdings keinesfalls erreicht.

Von einer eindeutigen Definition ist man mit dieser Erkenntnis weiter weg denn je. Aber gleichzeitig zeigt sie: Kaum ein IT-Segment scheint aktuell dynamischer und innovativer als der Hadoop-Markt zu sein.

deren Unternehmen zur Verfügung. Wie bei allen Cloud-Szenarien kommt aber dem Thema „Datensicherheit und Datenschutz“ eine geschäftskritische Bedeutung zu. Deswegen wird der Einsatz gerade in Deutschland kritisch und zum Teil kontrovers diskutiert

Die Plattformen stellen Infrastruktur as a Service sowie das Hadoop-Framework zur Verfügung, um Big-Data-Anwendungsfälle durchzuspielen. Der Vorteil liegt klar darin, dass sich die Nutzer voll auf den fachlichen Use Case konzentrieren können, statt sich um den Aufbau und den Betrieb der Infrastruktur zu kümmern. „Pay per use“, also nur für die wirkliche Zeit der Nutzung zu bezahlen, ist ein weiterer attraktiver Aspekt, der für den Einsatz spricht.

Da die Datenverarbeitung in der Cloud für viele Unternehmen aus datenschutzrechtlichen Gründen ein kritisches Unter-

fangen ist, müssen bei dieser Lösung natürlich besondere Vorsichtsmaßnahmen getroffen werden. Um hier jegliche Risiken zu vermeiden, gilt es im ersten Schritt, frühzeitig die genutzten Daten zu klassifizieren und individuelle Schutzmechanismen (organisatorisch und technisch) aufzusetzen. Eine frühzeitige Einbindung der Sicherheits- und Datenschutzbeauftragten im Unternehmen stellt sicher, dass das Vorhaben nutzenstiftend eingesetzt werden kann. Denn nicht alle Daten sind personenbezogen oder besonders schutzbedürftig.

Für den Umgang mit sensiblen Daten empfiehlt sich, sie in einer dedizierten Umgebung im Unternehmen zu halten oder mit mittlerweile am Markt verfügbaren Verschlüsselungslösungen zu schützen – und die Cloud für die Verarbeitung nicht sensibler Daten zu nutzen, zumindest für den Einstieg.



Oliver Herzberg  
oliver.herzberg@metafinanz.de



Slavomir Nagy  
slavomir.nagy@metafinanz.de

## Aufbau eines Semantic-Layers zwischen Datenbank und Hadoop

Matthias Fuchs, capgemini Nürnberg

Kein Business-Intelligence-Hersteller kommt momentan am Thema „Hadoop“ vorbei. Eine Integration ist Pflicht. Wie können Daten aus Hadoop eingebunden werden? Wie wirkt sich die Einbindung der Daten auf die Abfragegeschwindigkeit aus? Welche Art von Daten können aus BI-Sicht verwendet werden? Welche Möglichkeiten ergeben sich im Oracle-Umfeld?

Hadoop ist ein Framework, das es ermöglicht, verteilt arbeitende Rechenprozesse mit großen Datenmengen auf einem Cluster aus mehreren Servern durchzuführen. Zu den Basis-Komponenten gehören ein Dateisystem HDFS („Hadoop Distributed File System“) und die Möglichkeit, nach dem MapReduce-Algorithmus Berechnungen über mehrere Knoten durchzuführen. Erst durch die Entstehung von

meist Open-Source-Applikationen auf Basis von Hadoop wie Hive oder Spark wurde Hadoop zu einer Plattform, um vielfältige Daten zu speichern, zu analysieren und zu verarbeiten.

### Der Semantic-Layer

Im Rahmen von Business Intelligence versteht man unter einem Semantic-Layer eine Repräsentationsebene von Daten,

die es Endbenutzern ermöglicht, ohne technisches Wissen Daten einfach abzufragen. Der Begriff geht auf Business Objects zurück. Der Layer soll es ermöglichen, komplizierte Bezeichnungen beziehungsweise Beziehungen auf Objekte wie Produkte, Kunden oder Umsatz zurückzuführen. Der Semantic-Layer ist zentraler Bestandteil in den meisten BI-Tools von SAP, IBM, MicroStrategy oder Oracle.

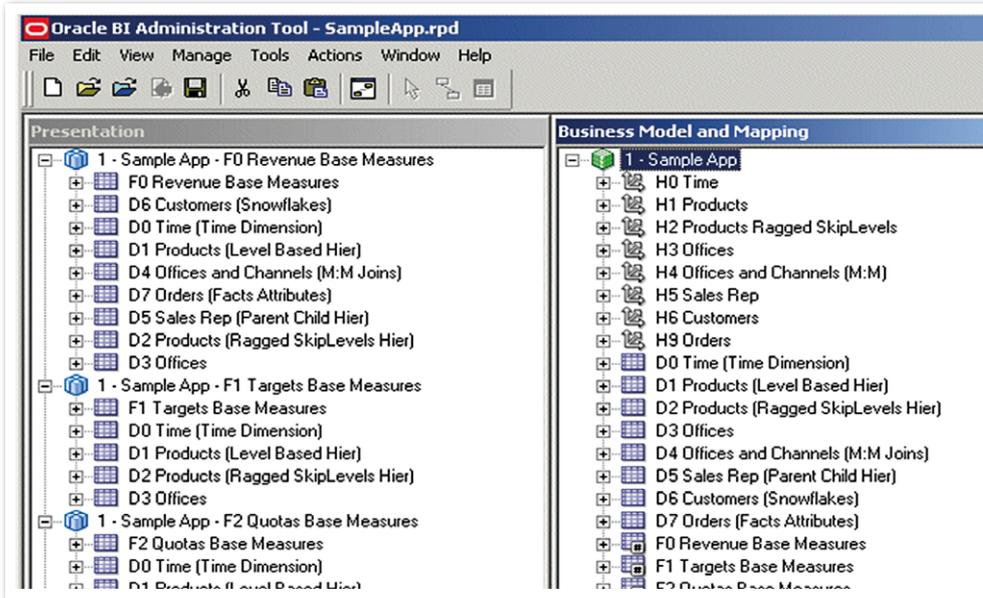


Abbildung 1: Semantic-Layer im OBIEE Admin

Ein Beispiel für einen Semantic-Layer ist im Administrationstool von Oracle Business Intelligence Enterprise Edition (OBIEE). Dort findet ein Mapping der physikalischen Tabellen auf das Business-Modell statt. Das Mapping wird im Repository abgelegt und lässt sich von OBIEE-Dashboards oder Reports abfragen. Eine Verwendung des so erstellten Layers geschieht im Rahmen von OBIEE, eine Migration auf Tools von anderen Herstellern ist meist nicht möglich (siehe Abbildung 1).

**Vor- und Nachteile**

Um Business-Anwendern einen einfachen Umgang mit den Daten zu ermöglichen, ist ein Semantic-Layer zwingend notwendig. Ohne diesen sind Ad-hoc-Queries nur machbar, wenn die Strukturen aus Tabellen und Views in der Datenbank dem Endbenutzer im Detail bekannt sind. Ebenso sollte der Layer mehr sein als nur ein direktes Mapping auf die Tabellen und Views in der Datenbank.

Andererseits ist ein erstellter Layer zwischen den verschiedenen BI-Tools verschiedener Hersteller nicht migrierbar oder wiederverwendbar. Bei Verwendung von mehreren Tools unterschiedlicher Hersteller muss also jedes Mal ein Semantic-Layer neu angelegt werden. Änderungen sind parallel in allen Werkzeugen nachzuziehen.

**Semantic-Layer und Big Data Hadoop**

Mit Big Data verbindet man unter anderem eine Sammlung vieler unterschied-

licher Daten. Diese unterschiedlichen Daten werden in einem verteilten, kostengünstigen Dateisystem wie Hadoop abgelegt. Ziel aus BI-Sicht ist es, auf diese Daten zuzugreifen. Somit ist eine Integration in den Semantic-Layer erforderlich. Zusätzlich zu den vorhandenen relationalen Daten sollen zum Beispiel XML/Rest- oder NoSQL-Daten, wie sie im Hadoop-Filesystem abgelegt werden, einbezogen werden. Der Aufbau des Layers wird schwieriger und aufwändiger.

Für den ersten Überblick über die Inhalte in einem Big-Data-Hadoop-Cluster haben sich Ansätze wie Semantic Data Preparation ergeben. Dabei wird versucht, mit visuellen Tools die Zusammenhänge in den Daten zu finden. Bei Oracle fällt dies in die Reihe „Big Data Discovery“. Es werden die Daten indiziert und ver-

sucht, Zusammenhänge zu finden. Man baut keine Strukturen auf, die eine Möglichkeit bieten, diese Daten im Semantic-Layer zu verarbeiten. Es findet also nur eine Vorverarbeitung statt, die als Basis für eine Verwendung der Daten im BI-Frontend dienen kann.

**Die Datenvielfalt**

In Hadoop können Daten in vielfältiger Weise abgelegt werden. Neben der Ablage in einer relationalen Datenbank wie Hive können Daten in Form von beliebigen Files abgelegt sein, dazu gehören auch binäre Daten wie Fotos oder Filme, XML/REST/JSON-Daten oder völlig strukturlos als Textdateien (siehe Abbildung 2). Eine direkte Verarbeitung in BI-Tools über HDFS auf File-Ebene ist nicht möglich, da ein Schema oder eine Struktur für die Daten fehlt. Im ersten Schritt muss diese Struktur aufgebaut werden. Wie ein Schema aufgebaut sein kann, wird unter anderem im HCatalog abgelegt.

**HCatalog – Metadaten-Layer in Hadoop**

Die Daten in Hadoop sollen über eine einheitliche Schnittstelle zugreifbar sein. Als Basis dient hier die relationale DB auf Basis von Hadoop oder Hive. Hive unterstützt die meisten Datentypen, die auch in einer relationalen Datenbank wie etwa Oracle vorhanden sind. Dazu kommt als Ergänzung die Funktion der Serializer/Deserializer (SerDes). Diese ermöglichen es, beliebige Objekte aus dem Hadoop-Filesystem oder anderen Applikationen zu lesen und in Hive zu verarbeiten. Anschließend können diese wieder geschrieben („deserialized“) oder an andere Program-

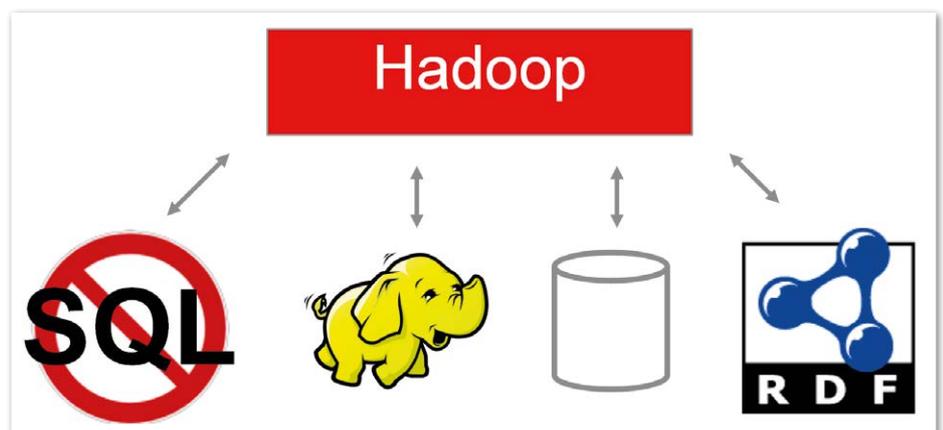


Abbildung 2: Die Datenvielfalt in Big Data

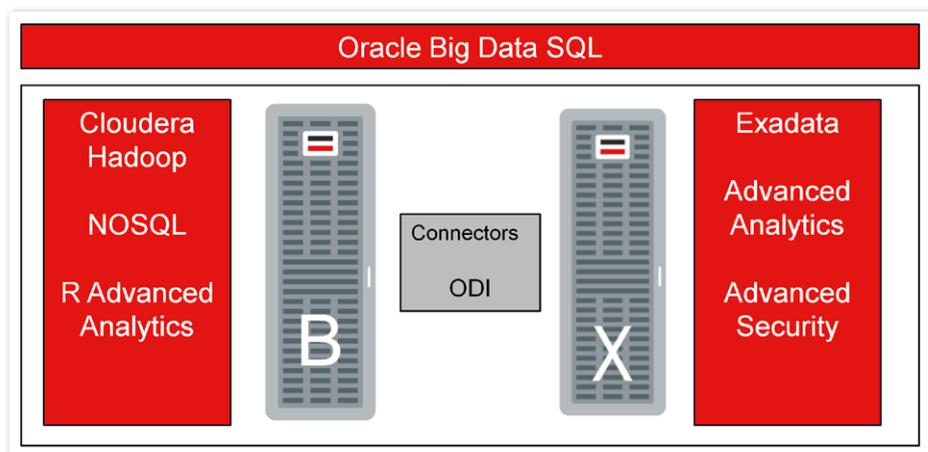


Abbildung 3: Big Data SQL

me weitergegeben werden. SerDes sind für viele Applikation beziehungsweise Datei-Typen vorhanden (HBase, Json, Avro etc.), können aber auch selbst in Java entwickelt werden.

Die Konfiguration, wie man auf die Daten zugreift, legt Hive im HCatalog ab. Dadurch entsteht eine Art semantischer Layer, der Strukturen und Schemas der Daten beschreibt.

Der HCatalog kann über eine REST-Schnittstelle abgefragt werden. Somit wird der Katalog nicht nur in Hive genutzt, sondern lässt sich von beliebigen Applikationen einsehen. Dazu gehören Cloudera Impala oder Big Data SQL. Dies sind selbstständige Applikationen auf dem Hadoop-Stack mit dem Ziel, schnelle Abfragen und Analysen zu ermöglichen. Sie bringen eine eigene Prozess-Engine mit, vergleichbar zu MapReduce. Die Tools lesen die Informationen aus dem Katalog und können daraus Basis-Strukturen als Tabellen und Views aufbauen. Der Zugriff auf die Daten findet nicht über Hive/MapReduce statt, sondern direkt zwischen der Applikation und HDFS unter Benutzung der eigenen Prozess-Engine.

### Performance-Unterschiede

Die Performance ist gerade bei Abfragen oder Dashboards entscheidend. Das Hadoop-Filesystem (HDFS) ist zum Lesen und Schreiben optimiert. Ein Update einzelner Blöcke war nicht vorgesehen. Es kann große Datenmengen schnell und parallel verarbeiten. Als Prozess-Engine dient von Beginn an MapReduce. Sie ist optimiert, um große Datenmengen in

Batchabläufen effektiv zu verarbeiten. Die Anforderungen an die Performance, speziell für Abfragen auf Teilmengen, stiegen und somit wurde die Prozess-Engine weiterentwickelt. Es entstanden „Tez“ und „Spark“ als Rechenkern, analog zu MapReduce. Ziel ist es, eine schnellere Verarbeitung als mit MapReduce zu erreichen. Gerade Spark versucht das mit verteilten In-Memory-Strukturen zu erreichen. Ebenso entstanden Applikationen mit eigener Prozess-Engines wie Impala, HAWQ oder Big Data SQL.

### Einbindung in den Semantic-Layer

Für die Einbindung in den Semantic-Layer ergeben sich vielfältige Möglichkeiten. Die Verwendung des HCatalog ist praktikabel. Eine Kombination mit einer performanten Prozess-Engine erscheint konsequent, um die Abfragen im Report oder Dashboard im Bereich von Sekunden zu halten. Der Connect zu Hive wird von fast allen Tools unterstützt, die Abfragesprache HiveQL ist SQL sehr ähnlich.

Hive wiederum unterstützt den Connect auf vielfache Daten innerhalb des Hadoop-Ökosystems. Ein Zugriff auf die meisten Daten in Hadoop-Umfeld ist somit gewährleistet. Die Konfiguration wird im HCatalog abgelegt und kann von weiteren SQL-Datenbanken benutzt werden. Neben Hive gibt es viele SQL-Datenbank-Applikationen im Hadoop-Ökosystem.

Alle Hadoop-Distributionen wie Cloudera, Hortonworks oder Pivotal bieten hier spezielle, für Query-optimierte Datenbanken an. Der Schwerpunkt in diesen Datenbanken liegt in der Abfrage-Performance und nicht in der Massen- und

Batchverarbeitung. Ebenso entsprechen sie im Gegensatz zu Hive dem ACID-Ansatz (Atomicity, Consistency, Isolation, Durability). Diese Funktionalität ist in bewährter Weise von einer Oracle-Datenbank abgedeckt; ein Einsatz zweier gleichartiger Datenbank-Typen ist nicht wünschenswert. Ebenso soll ein Duplizieren der Daten vermieden werden.

Die Lösung ist eine Einbindung der Hadoop-Daten möglichst einfach und performant in die bestehende Infrastruktur einer Oracle-Datenbank. Die Vorteile der Hadoop-Speicherung wie die günstige Datenablage sowie die Möglichkeit, viele Daten in einem Batchlauf zu analysieren und große Historien vorzuhalten, können genutzt werden. Die Art des Zugriffs – über die Oracle-Datenbank – verändert sich nicht. Daraus ergeben sich aus Oracle-Sicht zwei Möglichkeiten, Hadoop in ein BI-Tool einzubinden:

- Einbindung über die Datenbank mittels Big Data SQL beziehungsweise Konnektoren (siehe Abbildung 3)
- Direkte Einbindung über HiveQL in die BI-Schicht (siehe Abbildung 4)

### Big Data SQL und Konnektoren

Big Data SQL bietet die Möglichkeit, Objekte direkt aus der Datenbank abzufragen, die im HCatalog von Hadoop abgelegt sind. Big Data SQL ist eine Oracle-Applikation, die speziell für die Cloudera-Distribution entwickelt wurde. Neben der Integration von Tabellen und Views direkt in die Oracle-Datenbank ist mit Big Data SQL ein Offloading von Funktionen in den Hadoop-Cluster möglich. Dies ist ein ähnliches Verhalten wie bei der Database Machine mit den Exadata-Storage-Zellen. Dort werden ebenfalls rechenintensive Operationen aus der Datenbank ausgelagert.

Momentan ist eine Exadata-Datenbank als Integrationslayer für Big Data SQL erforderlich. Die Oracle Big Data Appliance mit der Cloudera-Distribution dient als Hadoop-Basis. Nur dort ist das Feature „Big Data SQL“ vorhanden.

Der entscheidende Vorteil aus der Kombination von Datenbank und Hadoop ist der Join von Datenbank- mit Hadoop-Tabellen. Der Datenbank-Optimizer erstellt über alle beteiligten Objekte den optimalen Ausführungsplan. Ein Offloading, also das Auslagern der Rechenlast nach

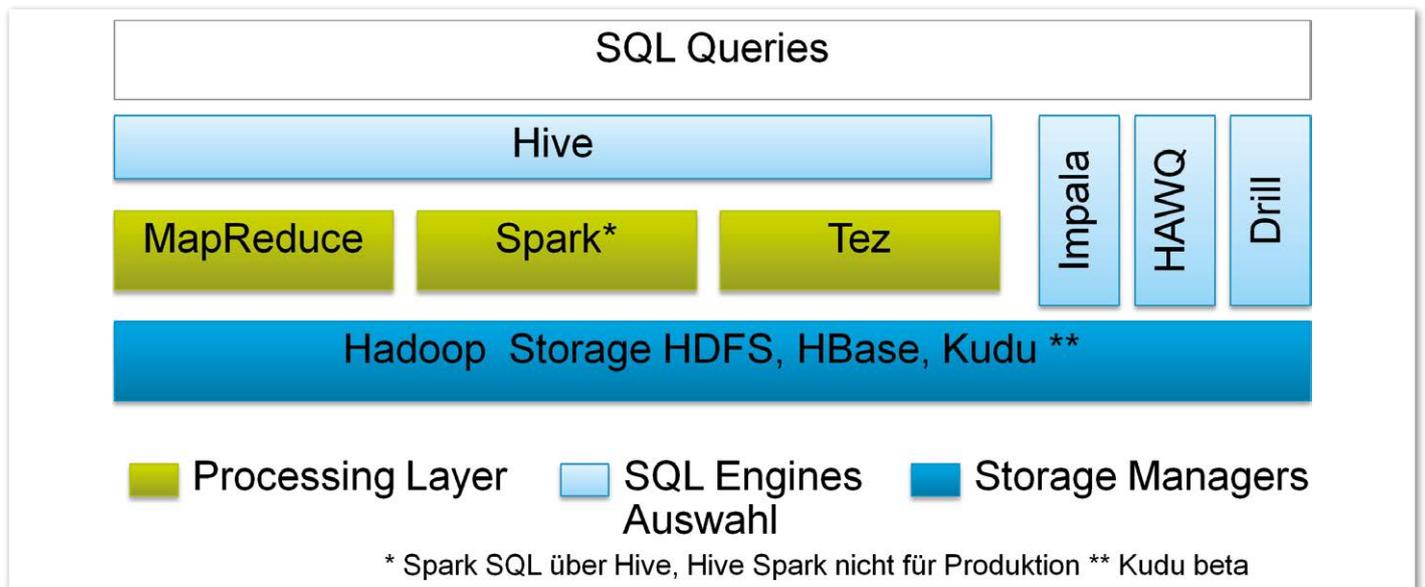


Abbildung 4: Direkter Hive Connect

Hadoop, findet statt. Dies ist möglich, da eine eigene Prozess-Engine im Hadoop-Cluster der Big Data Appliance läuft und dediziert angesprochen wird.

Beim Einsatz der Konnektoren statt Big Data SQL ist eine Einbindung der Hive-Tabellen in die Datenbank ebenfalls möglich. Dies funktioniert ohne eine Big Data Appliance, ohne Exadata und mit anderen Distributionen als Cloudera.

Ebenso lassen sich Daten aus der Datenbank in das Hadoop-Filesystem auslagern. Dazu sind ein Export einer Tabelle mit Oracle Data Pump, das Kopieren der Daten in das Hadoop-Filesystem sowie das Einbinden des Exports über Big Data SQL und externe Tabellen mit den Konnektoren in die Datenbank notwendig.

### Big Data SQL vs. HiveQL oder Impala

Die Vorteile von Big Data SQL sind offensichtlich:

- Es können gleichzeitig Objekte aus Hadoop und aus der Datenbank abgefragt werden und die Rechenoperationen sind auf beide Systeme verteilt. Aggregationen, die Daten aus beiden Systemen benötigen, erfolgen nicht erst im BI-Layer, sondern direkt bei den Daten bis hin zum Storage-System einer Big Data Appliance oder der Exadata.
- Die Daten werden über SQL in Rahmen eines Connect auf die Datenbank abge-

fragt, ein direkter Connect auf Hadoop findet nicht statt. Sicherheit und Auditing greifen wie bisher; es sind keine neuen Konzepte notwendig. Die Implementation ist einfach und schnell.

- Ist im BI-Tool ein Oracle SQL Connect möglich, sind über Big Data SQL oder die Konnektoren Abfragen auf Hadoop ohne Konfigurationen in den Data-Sourcen des BI-Tools durchführbar.

### Fazit und Ausblick

Die Integration von Hadoop-Daten in den BI-Layer ist möglich. Sind die Objekte in Hadoop („HCatalog“) bekannt und für das Data Warehouse verwendbar, können diese ohne großen Aufwand eingebunden werden.

Mit Big Data SQL stellt Oracle eine Möglichkeit bereit, hoch performant Daten im Hadoop-Kontext abzufragen. Big Data SQL versucht dabei, Rechen-Operationen möglichst nahe an den Daten ablaufen zu lassen, bei Exadata in den Exadata-Zellen, bei Hadoop in den Hadoop Data Nodes. Die Ressourcen der Systeme werden optimal ausgenutzt.

Ideen wie „Schema on Read“ oder „Schema on Demand“ sind zurzeit oft zu finden. Was darunter verstanden wird, ist allerdings unterschiedlich. Eine Annahme dabei ist, dass die Daten nicht strukturiert abgelegt sind. Beim Zugriff wird die Struktur über die Daten erzeugt. Die Verarbeitung ist variabel, da keine aufwändigen ETL-Prozesse implementiert werden müssen, die

die Daten strukturieren. Diese Informationen über die Struktur sind aufzubauen. Das passiert aber im Hadoop-Kontext bereits innerhalb des Clusters (Hive, Spark, R etc.). Somit gehen Informationen über die Strukturen der Daten des klassischen Semantic-Layers über in den Hadoop-Cluster beziehungsweise in die dort verwendeten Applikationen.



Matthias Fuchs  
Matthias.fuchs@capgemini.com



# Native JSON-Unterstützung in Oracle12c

Carsten Czarski, ORACLE Deutschland B.V. & Co. KG

Mit der neuesten Version 12.1.0.2 der Oracle-Datenbank ist erstmals die native Unterstützung für das JSON-Format möglich. JSON kann in die Oracle-Datenbank gespeichert und mithilfe von SQL/JSON-Funktionen ausgewertet werden. Neben dieser „SQL and JSON duality“ bietet die ebenfalls mitgelieferte REST-Webservice-Schnittstelle die Möglichkeit, die Oracle-Datenbank als „JSON Document Store“ zu betreiben – völlig ohne SQL.

Das JavaScript-Object-Notation-Format (JSON) setzt sich vor allem bei Anwendungsentwicklern mehr und mehr für den Datenaustausch und oft auch als Datenablage-Format durch. JSON ist im Grunde genommen JavaScript-Syntax. Kodiert man mit JavaScript-Code ein Objekt-Literal, liegt de facto JSON vor (*siehe Listing 1*).

JSON bildet wie XML die Daten in hierarchischer Struktur ab. Allerdings gibt es für den Umgang mit JSON weit weniger Standards als für XML – und die bestehenden sind weniger strikt. In der Praxis finden sich zwei typische Anwendungsgebiete für JSON. Zum einen wird es besonders im Umfeld von Web-Anwendungen als Datenaustausch-Format zwischen Server und Browser eingesetzt. Das ist einleuchtend, denn der Browser kann JSON, da es JavaScript-Syntax ist, problemlos und ohne Weiteres verarbeiten.

Das zweite Anwendungsgebiet ist die Datenablage im JSON-Format – diese

kommt vor allem dann zum Einsatz, wenn es nicht oder nur schwer möglich ist, ein festes Datenmodell und damit ein relationales Schema zu definieren. So kommen immer mehr Anwendungen in die Situation, dass flexible, sich häufig ändernde Attribute gespeichert werden müssen. Das Datenbank-Schema jedes Mal anzupassen, ist zu aufwändig – gefragt ist die einfache Ablage –, das Interpretieren der Daten geschieht dann später, wenn sie ausgewertet werden.

JSON bietet sich dafür an; gerade in einer Oracle-Datenbank kann es die perfekte Ergänzung zum relationalen Modell sein. Die statischen Teile des Datenmodells werden nach wie vor klassisch mit Tabellen und Spalten modelliert, die flexiblen Teile als JSON abgespeichert. Nun ist es natürlich nötig, dass man die als JSON gespeicherten Daten mit Mitteln der Datenbank, also SQL, bearbeiten und abfragen kann – und genau das ist der Schwerpunkt der JSON-Unterstützung in Oracle 12c (*siehe Abbildung 1*).

```
{
  "artikel": {
    "titel": "JSON in Oracle12c",
    "seiten": 3,
    "themen": ["Datenbank", "JSON", "SQL", „Tabellen“],
    "publikation": "DOAG Online"
  }
}
```

Listing 1

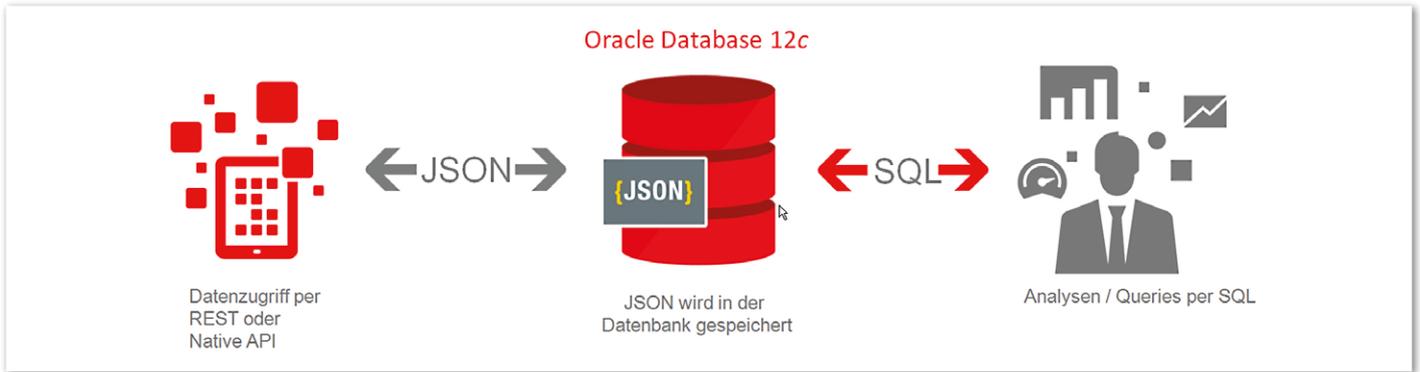


Abbildung 1: JSON and SQL Duality in Oracle12c

SQL ist allerdings nicht die einzige Schnittstelle für JSON in der Oracle-Datenbank. Darüber hinaus verfügt die Datenbank über eine REST-Webservice-Schnittstelle, mit der Anwendungen (völlig ohne SQL) JSON-Dokumente speichern, bearbeiten oder abfragen können. Dies ist insbesondere für Web-Applikationen ohne dedizierte JDBC-Datenbank-Schnittstelle interessant. Die Oracle-Datenbank kann so als „JSON Document Store“ (ganz ohne SQL) verwendet werden.

**JSON in der Datenbank speichern**

Im Gegensatz zu XML wurde für JSON kein eigener Datentyp in der Oracle-Datenbank eingeführt. JSON wird als „VARCHAR2“, „CLOB“ oder „BLOB“ in Tabellenspalten abgelegt. Damit ist es sogar möglich, JSON und Nicht-JSON in ein- und derselben Tabellenspalte zu mischen (ob das allerdings sinnvoll ist, ist eine andere Frage).

```
create table PO_JSON (
  ID      number(10) primary key,
  FILE_NAME varchar2(500),
  JSON    clob,
  constraint CK_JSON_IS_JSON check (JSON is json)
)
```

Listing 2

```
insert into PO_JSON (id, file_name, json) values (
  3, 'third-file.js', 'Ein Text'
)
*
```

FEHLER in Zeile 1:  
ORA-02290: CHECK-Constraint (JSONTEST.CK\_JSON\_IS\_JSON) verletzt

Listing 3

Möchte man feststellen, ob der Inhalt einer Tabellenspalte syntaktisch korrektes JSON ist, leisten die neuen SQL-Ope-

ratoren „IS JSON“ und „IS NOT JSON“ wertvolle Dienste. Wie das folgende Beispiel in Listing 2 zeigt, kann man diese auch als Check-Constraint verwenden.

JSON-Dokumente können in der Tabelle nun mit gewöhnlichen „SQL INSERT“-Anweisungen abgelegt werden. Möchte man in dieser Tabelle einen normalen Text speichern, der kein JSON repräsentiert, so wird der Check-Constraint eine Fehlermeldung auslösen (siehe Listing 3). Gültige JSON-Dokumente werden dagegen anstandslos entgegengenommen; Abbildung 2 zeigt ein etwas komplexeres JSON-Beispiel.

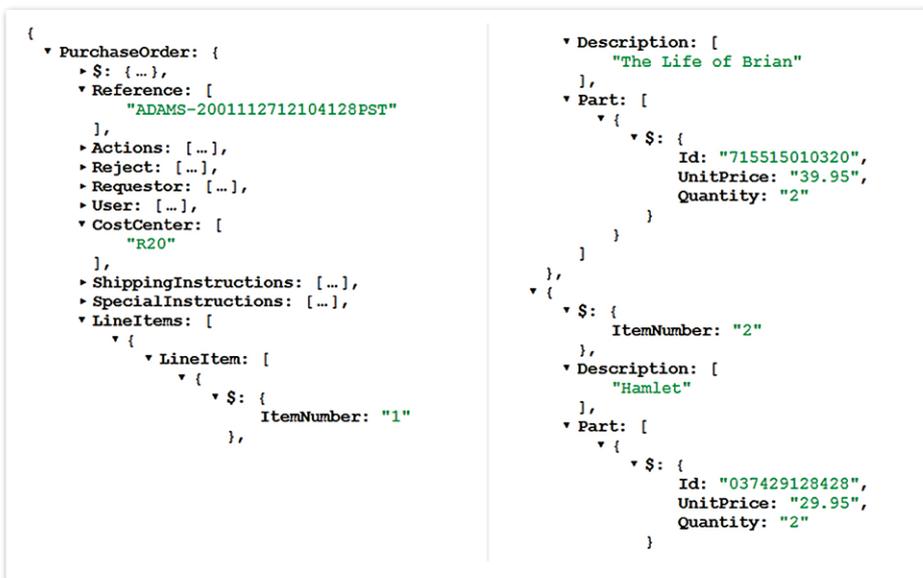


Abbildung 2: JSON-Beispiel „PurchaseOrder“

**SQL/JSON-Funktionen: JSON\_VALUE**

Die neue SQL/JSON-Funktion „JSON\_VALUE“ extrahiert einzelne skalare Werte aus den JSON-Dokumenten. Per Punkt-Notation gelangt man zum gewünschten Knoten in der Hierarchie (siehe Listing 4).

Die Funktion „JSON\_VALUE“ nimmt (wie auch die anderen SQL/JSON-Funktionen) zunächst das JSON-Dokument (oder die

```
select json_value(
  json,
  '$.PurchaseOrder.Reference[0]' returning VARCHAR2
) from po_json where rownum < 10;
```

```
JSON_VALUE
-----
FORD-20021009123336872PDT
JONES-20011127121050471PST
:
9 Zeilen ausgewählt.
```

Listing 4

```
select json_query(
  json, '$.PurchaseOrder.ShippingInstructions[0]'
) from po_json where rownum=1;
```

```
JSON_QUERY
-----
{"name":["Gerry B. Ford"],"address":["100 Oracle Parkway\r\nRedwood
Shores\r\nCA\r\n94065\r\nUSA"],"telephone":["650 506 7100"]}
```

Listing 5

```
select json_query(
  json, '$.invalid.json.path'
) from po_json where rownum=1;
```

```
JSON_QUERY
-----
- SQL NULL -
```

Listing 6

```
select e.json.PurchaseOrder.Reference
from po_json e where rownum <= 10;
```

```
PURCHASEORDER
-----
["FORD-20021009123336872PDT"]
["JONES-20011127121050471PST"]
["MARTIN-20011127121050401PST"]
:
10 Zeilen ausgewählt.
```

Listing 7

Tabellenspalte mit den JSON-Dokumenten) und dann einen JSON-Pfadausdruck entgegen. Darin ist das JSON-Dokument selbst mit dem Dollarzeichen repräsentiert; von dort aus navigiert man per Punktnotation durch die Hierarchie. Auch in Arrays kann navigiert werden, wie das obige Beispiel zeigt: „[0]“ repräsentiert das erste Objekt eines Arrays.

### Weitere SQL/JSON-Funktionen

„JSON\_QUERY“ ist das Gegenstück zu „JSON\_VALUE“: Diese Funktion dient dazu, JSON-Fragmente aus dem Dokument auszuschneiden. Der JSON-Pfadausdruck darf nun nicht mehr auf einen skalaren Wert zeigen – er muss auf ein JSON-Array oder Objekt verweisen (*siehe Listing 5*).

Im Fehlerfall liefert „JSON\_QUERY“ genauso wie „JSON\_VALUE“ standardmäßig „SQL NULL“ zurück. Dieses Verhalten kann jedoch durch Parameter verändert werden (*siehe Listing 6*).

### SQL-Punktnotation als Alternative

Anstelle der Funktionen „JSON\_QUERY“ oder „JSON\_VALUE“ kann man auch mit der „SQL-Punktnotation“ arbeiten (*siehe Listing 7*). Wie man sieht, braucht es dann keine „JSON\_QUERY“- oder „JSON\_VALUE“-Funktion mehr. Voraussetzung ist allerdings, dass, wie eingangs dargestellt, ein Check-Constraint mit „IS JSON“ auf der Tabellenspalte eingerichtet ist. Sonst funktioniert die vereinfachte Punktnotation nicht.

„JSON\_TABLE“ ist die mächtigste und mit Sicherheit auch interessanteste der SQL/JSON-Funktionen, denn sie erlaubt (ähnlich wie ihr XML-Pendant „XMLTABLE“) das Projizieren von JSON-Elementen auf relationale Spalten. Es können also in einem einzigen „JSON\_TABLE“-Aufruf mehrere JSON-Knoten selektiert und diese dann als relationale Ergebnisspalten projiziert werden. Die Abfrage in *Listing 8* zeigt dies: Es werden aus jedem JSON-Dokument drei Elemente extrahiert und als Ergebnis-Tabelle mit drei Spalten zurückgegeben.

„JSON\_TABLE“ kann aber noch weiter gehen: So ist es möglich, auch die enthaltenen Arrays („1:n“-Hierarchien) zu verarbeiten. Im Beispiel enthalten die „PurchaseOrder“-Dokumente jeweils ein Array mit den „LineItems“, also den bestellten Gegenständen. Natürlich kann eine „PurchaseOrder“ auch mehrere „LineItems“ beinhalten. Möchte man also alle „LineItems“ aller JSON-Dokumente als relationale Tabelle aufbereiten haben, so nimmt man dafür wiederum „JSON\_TABLE“ und die „NESTED PATH“-Klausel (*siehe Listing 9*).

„JSON\_TABLE“ ist eine extrem mächtige SQL-Funktion, man denke nur an das Weiterverarbeiten dieses Ergebnisses mit SQL-Aggregats-Funktionen wie „SUM“, „COUNT“ oder „AVG“. Die Möglichkeiten von SQL können nun sehr einfach auf JSON-Dokumente angewendet werden.

### Fazit

Mit den im Release 12.1.0.2 eingeführten SQL/JSON-Funktionen geht die Oracle-Datenbank, was die Unterstützung von JSON angeht, einen großen Schritt nach vorn. JSON-Daten können nun mit nativen

```

select reference, requestor, costcenter
from po_json, json_table(
  json,
  '$.PurchaseOrder'
  columns (
    reference varchar2(30) path '$.Reference[0]',
    requestor varchar2(25) path '$.Requestor[0]',
    CostCenter varchar2(4) path '$.CostCenter[0]'
  )
)

```

REFERENCE	REQUESTOR	COST
FORD-20021009123336872PDT	Gerry B. Ford	R20
JONES-20011127121050471PST	Richard J Jones	R20
:	:	:
SCOTT-20011127121051793PST	Susan T. Scott	R20

678 Zeilen ausgewählt.

Listing 8

```

select reference, requestor, num, descr, quantity
from po_json, json_table(
  json,
  '$.PurchaseOrder'
  columns (
    reference varchar2(30) path '$.Reference[0]',
    requestor varchar2(25) path '$.Requestor[0]',
    CostCenter varchar2(4) path '$.CostCenter[0]',
    nested path '$.LineItems[*].LineItem[*]' columns (
      num          number          path '$.\u0024.ItemNumber',
      descr        varchar2(40)    path '$.Description[0]',
      quantity     number          path '$.Part[0].\u0024.Quantity'
    )
  )
)

```

REFERENCE	REQUESTOR	NUM	DESCR	QUANTITY
FORD-20021...	Gerry B. Ford	1	Ordet	4
FORD-20021...	Gerry B. Ford	2	The Naked Kiss	3
FORD-20021...	Gerry B. Ford	3	Charade	2
:	:	:	:	:
ALLEN-2002...	Michael L. Allen	8	Sid & Nancy	2

14913 Zeilen ausgewählt.

Listing 9

SQL-Mitteln gelesen, interpretiert und verarbeitet werden – hinzu kommt der ganze SQL-Befehlsumfang.

Für PL/SQL Stored Procedures fehlt die JSON-Unterstützung im ersten Release noch – das Development-Team arbeitet daran. Hier lohnt es sich, einen Blick auf Apex 5.0 zu werfen. Das darin enthaltene PL/SQL-Paket „APEX\_JSON“ kann man auch schon auf der Early-Adopter-Instanz ausprobieren.

### Weitere Informationen

- [1] JSON auf Wikipedia: [http://de.wikipedia.org/wiki/JavaScript\\_Object\\_Notation](http://de.wikipedia.org/wiki/JavaScript_Object_Notation)
- [2] Oracle12c Dokumentation. JSON in der Oracle-Datenbank: <http://docs.oracle.com/database/121/ADXDB/json.htm#ADXD6246>
- [3] Artikel zum Thema in der APEX Community (mit Beispieldaten-Download): <http://tinyurl.com/oracle12cjson>



Carsten Czarski  
carsten.czarski@oracle.com

## Nach langer Wartezeit: Oracle Forms und Reports 12c erschienen

Nach langer Verzögerung hat Oracle zum Auftakt der Oracle OpenWorld in San Francisco das Release von Forms und Reports 12c verkündet. Der Konzern verspricht mit der neuen Version zahlreiche neue Features und Performance-Optimierungen. Entgegen vieler Mutmaßungen ist Oracle Forms also keinesfalls tot, sondern soll sich den gewachsenen Ansprüchen an ein zeitgemäßes Entwicklungswerkzeug anpassen. Die neue

Version von Oracle Forms und Reports verspricht mehr Features und höhere Performance. Die neuen Features von Oracle Forms und Reports 12c sind:

- **Kein Browser mehr notwendig**  
Oracle Forms 12c ist mit einem eigenen Browser ausgestattet, so werden Kompatibilitätsprobleme vermieden.
- **Forms2XML-Integration**  
Die Forms2XML-Applikation wurde in den Forms Builder integriert.
- **Verbesserte Usability**  
Benutzer können Elemente innerhalb der Oberfläche verschieben, verkleinern und vergrößern.
- **Verbesserte Single Sign-on-Integration**  
Einmal angemeldet ist die Login-Session für unterschiedliche Anwendungen gültig, die Forms-Applikation lässt sich damit leichter in heterogene Umgebungen integrieren.



# Zweiter Frühling für die Analyse unstrukturierter Daten

Christopher Thomsen, OPITZ CONSULTING GmbH

Die meisten weltweit verfügbaren Daten sind unstrukturiert, die Mehrheit dieser unstrukturierten Daten sind Texte. Doch die meisten beschreibenden und vorhersagenden Analyse-Verfahren können den Informationswert dieser Daten nicht nutzen. Dabei können vor allem textuelle Daten aus sozialen Medien, Mails, Geschäftsberichten, Webseiten-Inhalten, Fachartikeln und den vielen weiteren Quellen von textbasierten Informationen dabei helfen, Trends zu erkennen, Profile zu schärfen oder Suchen zu optimieren. Dieser Artikel erläutert die Grundlagen der Text-Analyse und gibt einen Einblick in Volltextsuche, Natural Language Processing und Machine-Learning-Verfahren.

Unstrukturierte Daten sind Daten, die für konventionelle Verarbeitungssysteme keine erkennbare Struktur aufweisen. Der Anteil dieser Daten an allen digital gespeicherten Informationen beträgt etwa 80 Prozent. Bisher beschränkt sich die Mehrheit aller Werkzeuge zur beschreibenden und vorhersagenden Analyse auf

die verbleibenden 20 Prozent. Dabei findet man unstrukturierte Daten überall. Sie lassen sich grob in zwei Gruppen untergliedern: in maschinengenerierte und in von Menschen generierte Daten.

Zu ersteren gehören vor allem Bilder und Videos, die von einer Kamera erzeugt werden, aber auch Satelliten-, Radar- und

Sonaraufnahmen. Die von Menschen generierten unstrukturierten Daten besitzen meist eine übergeordnete gemeinsame Struktur: die Syntax der Sprache, die diese Menschen sprechen. Zu diesen Daten gehören E-Mails, Dokumente, Social-Media-Daten, SMS und andere Textnachrichten sowie der Inhalt von Internetseiten.

**Oracle Text:**

```
CREATE INDEX my_index ON MyDocs(my_text)
  INDEXTYPE IS CTXSYS.CONTEXT PARAMETERS
  ('FILTER CTXSYS.NULL_FILTER SECTION GROUP CTXSYS.HTML_SECTION_GROUP');
```

**tsearch2:**

```
UPDATE MyDocs my_index = to_tsvector('german', coalesce(my_text, ''));
```

**Apache Lucene:**

```
new QueryParser(LUCENE_VERSION, "", new GermanAnalyzer(LUCENE_VERSION)).parse(my_text);
```

Listing 1: Funktionen der Text-Extraktion in verschiedenen Textanalyse-Frameworks

Diese Formate enthalten zwar nicht die Informationsdichte strukturierter Formate, doch allein aufgrund der ungeheuren Menge an verfügbaren unstrukturierten Daten darf diese Informationsquelle bei ganzheitlichen Analyseansätzen nicht ignoriert werden.

Klar abzugrenzen von dieser Definition sind demnach Datenformate, die eine klare, wenn auch nicht schematisch festgelegte Struktur aufweisen. Darunter fallen beispielsweise Sensordaten, Serverlogs, XML- und JSON-Dokumente sowie Graphen. Diese Formate werden meist unter den semistrukturierten Daten zusammengefasst.

**Text-Extraktion**

Unabhängig davon, welches Ziel man bei der Arbeit mit Texten verfolgt, steht normalerweise am Anfang stets die Text-Extraktion, ein Verfahren, mit dem aus dimensionslosen Satzkonstrukten Einzelbausteine des Textes extrahiert und in eine normalisierte Form gebracht werden können. Eine Text-Extraktion läuft in vier Schritten ab:

- **Filtering**  
Das Entfernen nicht benötigter Interpunktion und die Zusammenführung von getrennten Begriffen, etwa beim Zeilenumbruch mit Bindestrich.
- **Tokenization**  
Unterteilung des Textes in einzelne Wörter. Hierbei werden Bindestriche und andere sprachabhängige Gegebenheiten zur Verkettung von Wörtern berücksichtigt.
- **Recognition**  
Die Erkennung von Entitäten unter den Tokens. Je nach Text-Parser kommen hier unterschiedliche Sets an Wörterbüchern und reguläre Ausdrücke („Regex“)

zum Einsatz. So werden häufig Namen, Orte, Geldbeträge, Jahreszahlen, Daten und Telefonnummern erkannt. Auch sogenannte „Stop Words“, also Wörter, die für die spätere Weiterverarbeitung aufgrund ihrer Häufigkeit und fehlenden Aussagekraft unnötig sind (wie „ist“, „werden“, „machen“), werden in diesem Schritt entfernt.

- **Stemming**  
Die Reduzierung aller Wörter, die im Schritt „Recognition“ nicht als Eigennamen identifiziert worden sind, auf ihren Wortstamm. So wird aus „schreiben“, „schrieb“, „schreibt“ oder „geschrieben“ in allen Fällen „schreib“. Auf diese Art und Weise entfällt beim späteren Wortvergleich oder bei Suchfunktionen die Komplexität, die eine Sprache durch Deklination und Konjugation bekommt.

Dieses zunächst äußerst kompliziert wirkende Verfahren wird heute von allen Werkzeugen zur Textanalyse und Volltext-

suche unterstützt und lässt sich üblicherweise mit einem einzigen Befehl ausführen. Listing 1 zeigt dies an drei Beispielen.

Diese vorhandenen Werkzeuge eignen sich natürlich nur dann für die Anwendung, wenn die zur Verfügung stehenden Text-Extraktoren – also Recognition-Wörterbücher, RegEx für „Stop Words“ und weitere zu erkennende Entitäten – ausreichen. Ist dies nicht der Fall, müssen für den Extraktionsprozess eigene Text-Extraktoren entwickelt werden.

Unabhängig von der verwendeten Technologie verläuft die Entwicklung der Text-Extraktoren nach einem ähnlichen Muster. Als Grundlage für die Annotation des Textes wird ein repräsentativer Beispielauszug des Gesamtdatenbestands in das Entwicklungssystem importiert. Der Entwickler definiert die Attribute und Formate, die er als Ergebnis der Analyse erhalten möchte, und nutzt Wörterbücher sowie RegEx, um alle Wörter in den Beispieltextrakten zu annotieren, die er für die Ermittlung seiner Zielattribute benötigt. Diesen Vorgang wieder-

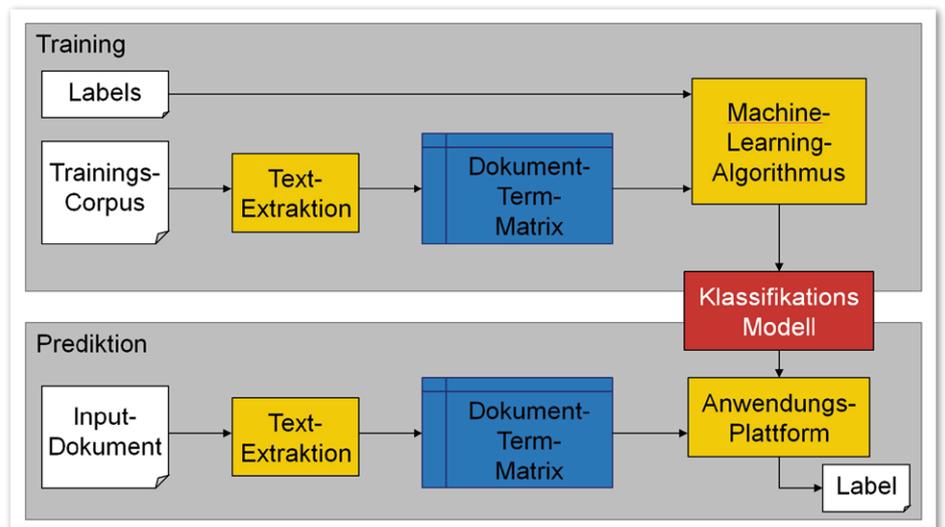


Abbildung 1: Evolution der Textanalyse

holt er inkrementell, um eine wachsende Abdeckung zu erzielen.

### Natural Language Processing

Alternativ zu Wörterbüchern und regulären Ausdrücken bieten Natural Language Processing Engines (NLP) weitere Verfahren, um die extrahierten Tokens eines Textes mit weiteren Attributen zu annotieren. Dies können beispielsweise sprachsyntaktische Informationen der Grammatik sein. Das Angebot an NLP-Engines ist vielfältig – ob Open Source oder kommerziell verfügbar. Es gibt hier keine klaren Marktführer, da die Anforderungen sehr unterschiedlich sind und die Analyse-Qualität je nach Anwendungsfall stark variieren kann. Zu den grundlegenden Funktionalitäten einer NLP-Engine gehören:

- Satzerkennung, nicht nur anhand von Interpunktion, sondern auch anhand des Vorhandenseins aller Satzbestandteile

- vorgefertigte RegEx für die Erkennung von Geldbeträgen, Namen und Ähnlichem
- Categorizer, die meist basierend auf einem sehr generischen Corpus den Text einer Gattung zuordnen
- Syntax Parser, die alle Tokens eines Satzes in ihren syntaktischen Eigenschaften bestimmen

Mithilfe dieser Anreicherungen können weitere Filterungen durchgeführt oder Bezüge hergestellt werden. So kann zum Beispiel bei einer Suche explizit zwischen dem substantivierten Gebrauch des Worts und dem Gebrauch als Adjektiv unterschieden werden.

### Volltextsuche

Volltextsuche ist weit mehr als die Suche einer Zeichenkette in einem Text oder die Anwendung eines regulären Ausdrucks. Sie bildet die Grundlage für höherentwickelte Textanalyse-Verfahren (siehe Abbildung 1):

- *Lexem-Erkennung*  
Mittels Stemming werden Worte unabhängig von ihrer Deklination oder Konjugation gefunden
- *Volltext-Indizierung*  
Durch die Indizierung der Texte ist die Suche deutlich schneller möglich als mit aktuellen RegEx-Interpretern
- *Ranking-Informationen*  
Informationen, die zur Bestimmung des Grads der Übereinstimmung eines Textes mit dem Suchbegriff herangezogen werden können
- *Interpretation von Satzbau und Satzzeichen via NLP*  
Beispielsweise würde eine Volltextsuche nach „neue Informationen schnell gewinnen“ in einem Text auch die Passage „durch schnelle Informationsgewinnung“ finden, was mit dem bloßen Matching von Zeichenketten oder mit RegEx nicht möglich gewesen wäre.

## Boosten Sie Ihre Kenntnisse mit unserem InSite Workshop-Programm 2016!

**dbi** InSite  
Workshops

Insider-Wissen von IT-Experten: Unsere massgeschneiderten Workshops für Oracle, SQL Server, MySQL, Linux & mehr.

Phone +41 32 422 96 00 · Basel · Lausanne · Zürich

[dbi-services.com/de/schulungen](http://dbi-services.com/de/schulungen)



Infrastructure at your Service.

**dbi** services

Die Volltextsuche gliedert sich in die Indizierung der Texte und die eigentliche Abfrage. Bei der Indizierung wird nach der Text-Extraktion ein Index für alle Tokens in allen indizierten Texten angelegt. Dieser Index beinhaltet das normalisierte Token – also das normalerweise auf den Wortstamm reduzierte Wort – sowie alle Positionen im Text, an denen das Token auftaucht. Bei der Suche wird der eigentliche Text also gar nicht prozessiert, sondern es wird lediglich der Volltext-Index mit den Suchbegriffen abgeglichen, die zuvor den Prozess der Text-Extraktion durchlaufen haben.

Neben dem schnellen Finden von Übereinstimmungen in Texten ist das Ranking der Suchergebnisse ein wichtiger Bestandteil der Volltextsuche, um diese dem Nutzer beziehungsweise der verarbeitenden Applikation in nach Relevanz sortierter Reihenfolge zur Verfügung zu stellen. Die dabei für das Ranking herangezogenen Faktoren können je nach Anwendungsfall variieren, üblicherweise fließen jedoch mindestens diese drei Faktoren mit ein:

- **Lexikalische Relevanz („lexical“)**  
Wie häufig kommen Suchbestandteile in dem Dokument vor beziehungsweise wie hoch ist die Trefferdichte?
- **Nachbarschaftsrelevanz („proximity“)**  
Wie nahe liegen die Suchbestandteile (bei der Suche mehrerer Begriffe) beieinander?
- **Strukturelle Relevanz („structural“)**  
Wie wichtig sind die Teile des Textes, in denen die Suchbegriffe vorkommen?

Diese Punkte werden oft mit weiteren Meta-Informationen wie Aktualität des Textes, Bezügen von anderen Dokumenten, Format, Textgattung etc. kombiniert. Jeder dieser Faktoren wird dann üblicherweise als ein normalisierter Wert zwischen 0 und 1 abgebildet und mit den anderen Faktoren multipliziert. Das Ergebnis liefert eine gute erste Einordnung, für eine Feinabstimmung muss jedoch noch an den oft manuell zu setzenden Wichtungsfaktoren gedreht werden.

Zum Beispiel kann die lexikalische Relevanz, also die Dichte der Suchbegriffe in den Texten, auf diverse Arten errechnet werden: Die einfachste Methode ist eine einfache Division, sodass bei einem Treffer die Relevanz des Dokuments mit zu-

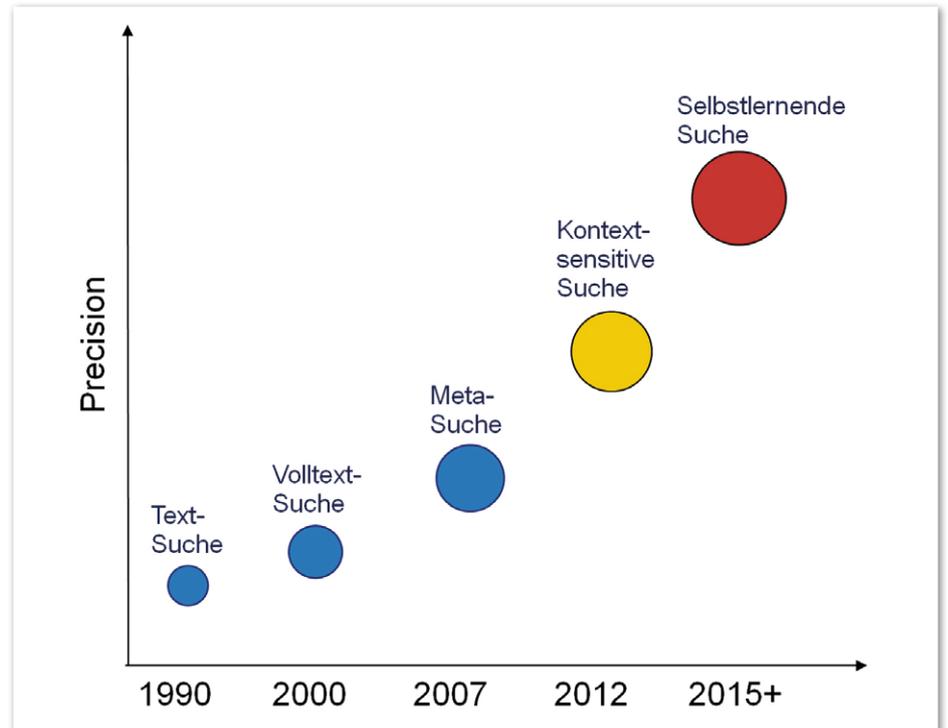


Abbildung 2: Training und Anwendung eines Klassifikationsmodells durch überwachtes Lernen

nehmender Länge linear abnimmt. Bessere Ergebnisse liefert das „tf-idf“-Maß, das statt der Länge des Dokuments die Häufigkeit des am häufigsten genutzten Worts als Divisor verwendet. Oft werden diese Ergebnisse für eine feinere Abstufung noch logarithmiert, durch Hauptachsen-Transformationen mit einem Schwellwert versehen oder progressiver beziehungsweise degressiver gemacht.

### Text-Klassifikation

Gerade im Bereich der Textsuche und -analyse ergibt sich eine Vielzahl von Faktoren, die für eine Texteinordnung, -suche oder -bewertung eine Rolle spielen können, aber stets nur in Kombination mit anderen Faktoren ein nutzbares Resultat liefern. Die Abstimmung dieser Faktoren aufeinander spielt hierbei eine entscheidende Rolle und kann häufig nur bis zu einem bestimmten Grad manuell vorgenommen werden.

Die Klassifizierung von Texten ist einer der häufigsten Anwendungsfälle für die Textanalyse. Hierbei soll ein unbekannter Text einer oder mehreren vorher definierten Gruppen von Texten zugeordnet werden. Die Ergebnisse sehen wir täglich, etwa wenn Google Werbung anzeigt, die zum einen auf uns persönlich angepasst ist, zum anderen aber auch auf dem In-

halt der Seite basiert, die wir gerade ansehen. Oder wenn unser Google-Gmail-Konto uns automatisch vorschlägt, in welche Unterordner es eingehende Mails aufgrund ihres Inhalts einsortieren würde. In Support-Centern werden Classifier häufig verwendet, um Mails nach ihrer Thematik vorzusortieren, damit sie schneller bei der richtigen Person landen. Und im Bereich der Textsuche werden Classifier verwendet, um für eine Suche zu bestimmen, in welchem Kontext die Suche überhaupt stattfinden soll.

Suchen wir beispielsweise bei Google nach „Apache Tomcat“, finden wir seitenweise Treffer zum gleichnamigen Applikationsserver. Wir finden keine Ergebnisse aus dem Bereich des U.S. Militärs, dessen Luftstreitkräfte sowohl über das Grumman F-14 Tomcat Kampfflugzeug als auch den Boeing AH-65 Apache Kampfhubschrauber verfügen. Google schließt aus der Kombination dieser beiden Begriffe, dass der Kontext „IT“ und nicht „Militär“ ist, und gewichtet Resultate höher, die der Kontextgruppe „IT“ zugeordnet sind.

Ein Classifier ist also ein Bemessungssystem, in dem eine Entität anhand ihrer Attribute ein oder mehrere Labels erhält. Er kann in einfachster Form durch ein regelbasiertes System manuell abgebildet

werden, in dem hinter jeder Ausprägung Regeln hinterlegt sind.

Solche Systeme werden meist mithilfe von Business Rule Engines wie Apache Drools abgebildet. Begibt man sich in den Bereich der maschinengestützten Lernverfahren, das sogenannte „Machine Learning“, unterscheidet man grob zwei Lernformen: überwachtes und unüberwachtes Lernen. In beiden Fällen wird ein Set von Texten benötigt, mit denen das System lernen kann. Dieses als „Corpus“ bezeichnete Trainings-Datenset durchläuft den Prozess der Textextraktion und wird dann in eine Dokument-Term-Matrix zerlegt, also eine Tabelle, die die Häufigkeit einzelner Wörter im selben Text aufzeigt. Diese Matrix kann nach Bedarf transformiert werden, um etwa für jedes Wort zu bestimmen, wie oft es mit jedem anderen Wort im Schnitt im gleichen Text vorkommt.

Diese Matrizen bilden also die Grundlage für den zu trainierenden Classifier. *Abbildung 2* stellt den Trainings- und Ausführungsprozess vereinfacht dar. Überwachtes und unüberwachtes Lernen unterscheiden sich hier lediglich in dem Punkt, dass beim überwachten Lernen die Labels, also die Gruppen in die klassifiziert werden soll, vorgegeben sind, während beim unüberwachten Lernen häufig nur die Anzahl der Gruppen, die gebildet werden sollen, vorgegeben wird und der Algorithmus die Definition der Gruppen selbst optimiert. Das exakte Verfahren zur Erstellung des Classifiers aus dem Corpus hängt vom verwendeten Lernalgorithmus ab.

Bekannte Klassifikationsverfahren des überwachten Lernens sind der Bayes-Klassifikator, das Perzeptron (neuronalen Netz) und die Support Vector Machine (SVM). Verfahren des unüberwachten Lernens sind zumeist Segmentierung/Clustering-Verfahren.

### Text Mining oder Volltextsuche

Volltextsuche ist mittlerweile ein gestandenes Verfahren mit etablierten Technologien, während gerade mit dem Big-Data-Hype immer neue Data- und Text-Mining-Werkzeuge für skalierbare Plattformen wie „Spark on Mesos“ oder „Hadoop“ auftauchen. Beide Verfahren ähneln sich in der Aufbereitung der Daten, ihre Funktion ist jedoch auf ihren pri-

mären Anwendungszweck ausgelegt. Bei der Volltextsuche heißt dies, für einen Suchbegriff möglichst rasch eine nach Relevanz bewertete und sortierte Liste mit Antworten zu liefern, während die Textanalyse beziehungsweise das Text Mining das Ziel verfolgt, einen Zusammenhang zwischen den Texten herzustellen, und daher meist mit Machine-Learning-Verfahren verknüpft ist.

In der Praxis gehen beide Ansätze allerdings meist Hand in Hand. Nehmen wir zum Beispiel eine Textsuchmaschine, so ist die Definition von Synonymen und Ähnlichkeiten zwischen Begriffen ein wichtiger Aspekt. Diese Synonyme und Wortbeziehungen können je nach Kontext unterschiedlich sein. Während das englische Wort „Spring“ (Frühling) in einem generischen Kontext nichts mit dem Begriff „Struts“ (Säulen) zu tun hat, haben diese beiden Begriffe im IT-Kontext durch einst sehr verbreitete Java-Frameworks eine starke semantische Beziehung. Die intelligente Suchmaschine eines IT-Unternehmens sollte bei einer Suche nach „Java“ also auch Texte finden, in denen zwar das Wort „Java“ nicht vorkommt, jedoch Begriffe wie „Servlet“, „Spring“ oder „Hibernate“.

An dieser Stelle kommen Text-Mining-Verfahren ins Spiel, die auf Basis eines großen Text-Corpus aus dieser IT-Firma bereits die Beziehungen zwischen allen Begriffen erlernen und in einer indizierten Synonymtabelle mit entsprechenden Korrelationsfaktoren ablegen. Unabhängig von der Suche erstellt ein solches Verfahren in selten ausgeführten Batchläufen, die aufgrund ihrer Rechenanforderung oft viele Stunden oder Tage dauern können, die für die Suche zur Laufzeit benötigten indizierten Informationen, um Suchanfragen schnell und präzise beantworten zu können.

### Ein Wort zur Infrastruktur

Aufgrund ihres sehr unterschiedlichen Profils laufen Anwendungen für Text Mining und Textsuche meist auch auf technologisch getrennten Plattformen. Mit dem Big-Data-Hype werden für das Text Mining skalierbare, offene Plattformen wie Hadoop und Spark immer populärer, besonders wenn die Durchlaufzeit für die Batchläufe zeitkritisch ist.

Die Textsuche wird meist in relationalen Datenbanken wie Oracle Text in der

Oracle-Datenbank oder tsearch2 in der PostgreSQL-Datenbank ausgeführt, dort sind diese ein bewährter Bestandteil. Die zuvor erstellten Modelle werden hierzu in ein relationales, tabellarisches Schema übertragen. Speziell dafür ausgelegte Text-Index-NoSQL-Datenbanken wie Solr oder Elasticsearch werben hier mit deutlich besserer Skalierbarkeit zu geringeren Kosten, besitzen jedoch nicht die Allrounder-Funktionalitäten einer relationalen Datenbank. Hier ist abzuwägen, ob die erwartete Indexgröße die Leistungsfähigkeit eines einzelnen Datenbanksystems übersteigen würde, was aber meist erst bei einem Index über viele Millionen Dokumente der Fall ist.

Bei all dem Wirbel um die Arbeit mit unstrukturierten Daten und den vielen neuen Produkten in diesem Umfeld sollte man sich immer vor Augen führen, dass es sich hier keinesfalls um ein neues Thema handelt. Letzten Endes läuft weiterhin alles auf einen Daten-Parser hinaus, der entsprechend seinem Reifegrad bei der Textverarbeitung viele Dinge von selbst übernimmt, etwa mathematische Verfahren zur Hauptkomponenten-Analyse und Clustering sowie auf an textuelle Datenformate angepasste Indizes; alles Verfahren, für die uns seit Ende der 1990er-Jahre bereits Werkzeuge zur Verfügung stehen, die derzeit allerdings unter dem Hype-Begriff „Big Data“ einen zweiten Frühling erleben.



Christopher Thomsen  
christopher.thomsen@opitz-consulting.com

# Die Oracle-Datenbank jenseits von Entity-Relationship-Modellierung – vorhandenes Wissen repräsentieren und neues mit RDF Graph generieren

Karin Patenge, ORACLE Deutschland B.V. & Co. KG

Die Oracle-Datenbank vereint in einzigartiger Weise unterschiedliche Datenmodelle. So finden relationale Daten neben JSON- oder XML-Dokumenten ihren Platz ebenso wie raumbezogene Daten oder unstrukturierte Inhalte wie Multi-Media-Dateien. Sogar die spaltenbasierte Datenhaltung im Arbeitsspeicher wird seit der Einführung der In-Memory-Option mit Version 12.1.0.2 durch die Datenbank unterstützt. Weniger bekannt, aber dafür im Informationszeitalter umso spannender ist, dass auch ein Datenmodell speziell für Begriffe („Ressourcen“) und deren Beziehungen („Relationships“) untereinander vorgehalten wird. Termini, die in diesem Zusammenhang immer wieder fallen, sind „Semantische Netze“, „Resource Description Framework“ (RDF), „Linked Data“ oder „Ontologien“. Die Datenbank stellt dafür das Feature „RDF Graph“ innerhalb der Option „Spatial and Graph“ bereit.

Wissens- und Informations-Zusammenhänge zu repräsentieren, bewegt sich sehr häufig außerhalb der Grenzen einer relationalen Datenmodellierung. Denn in einem Entity-Relationship-Modell (ER) müssen Entitäten, deren Attribute und die Beziehungen zwischen den Entitäten vorab definiert werden. In den weiteren Modellierungsschritten werden daraus Tabellen, Spalten mit zugeordneten Datentypen und Foreign Key Constraints, die über ein SQL-Skript in der jeweiligen Datenbank angelegt sind. Änderungen an der Applikation, die darauf abzielen, zusätzliche Informationen oder Zusammenhänge aufzunehmen, müssen im Datenmodell relativ aufwändig nachgezogen werden.

Eine Alternative ist die Repräsentation von Informationen und Informations-Zusammenhängen als Graphen in einem Knoten- und Kantenmodell. Die Knoten stellen dabei Begriffe oder beliebige Ressourcen dar. Über die Kanten werden Zusammenhänge zwischen diesen abgebildet. *Abbildung 1* veranschaulicht das Graphenmodell

in Form eines einfachen Anwendungsbeispiels, eines Familienverbands. Die Knoten sind die einzelnen Mitglieder, die Kanten zeigen, wie sie miteinander verwandt sind.

## Die Oracle-Datenbank als Triple Store

Knoten und Kanten sind in der Datenbank als sogenannte „Triples“ gespeichert. Das

sind Aussagen, bestehend aus Subjekt („Subject“), Prädikat („Property“) und Objekt („Object“), für die der Datentyp „SDO\_TRIPLE\_RDF\_S“ zum Einsatz kommt (*siehe Listing 1*).

Eine zu speichernde Aussage gemäß *Abbildung 1* ist nun „Maria (s) isMotherOf (p) Mia (o)“. Für die einzelnen Triple-Teile gilt:

```
-- Table with triples
create table family_rdf_data (
  id integer generated as identity (start with 1 increment by 1) primary key,
  triple sdo_rdf_triple_s
)
/

-- Create model
execute sem_apis.create_sem_model(
  'family',          -- Model name
  'family_rdf_data', -- Table name
  'triple'          -- Column name
)
/
```

Listing 1: Anlegen einer Triple-Tabelle sowie Registrieren eines semantischen Modells (Graph)

- Ein Subjekt (s) ist entweder ein Uniform Resource Identifier (URI) oder ein leerer Knoten („blank node“)
- Ein Prädikat (p) ist immer ein URI
- Ein Objekt (o) kann ein URI, ein Literal oder auch ein leerer Knoten sein

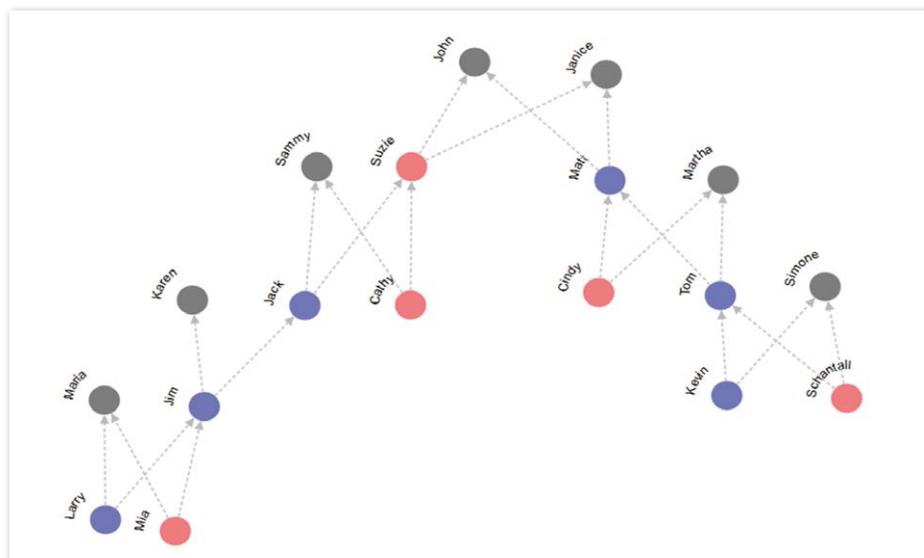


Abbildung 1: Ausschnitt aus einem Familienverband als Graphenmodell.

```

-- Maria ist die Mutter von Mia.
SQL> INSERT INTO family_rdf_data VALUES (
  SDO_RDF_TRIPLE_S(
    'family',
    '<http://www.example.org/family/Maria>',
    '<http://www.example.org/family/motherOf>',
    '<http://www.example.org/family/Mia>'))
/

-- Schantall ist (vom Typ) weiblich.
SQL> INSERT INTO family_rdf_data VALUES (
  SDO_RDF_TRIPLE_S(
    'family',
    '<http://www.example.org/family/Schantall>',
    '<http://www.w3.org/1999/02/22-rdf-syntax-ns#type>',
    '<http://www.example.org/family/Female>'))
/

-- Cathy hat die Größe von 5.8 (Fuß).
SQL> INSERT INTO family_rdf_data VALUES (
  SDO_RDF_TRIPLE_S('family',
    '<http://www.example.org/family/Cathy>',
    '<http://www.example.org/family/height>',
    '"5.8"^^xsd:decimal'));

SQL> commit
/

-- Extrahieren der einzelnen Triple-Bestandteile aus der Spalte TRIPLE
SQL> select id,
  f.triple.get_model() m,      -- Semantisches Modell
  f.triple.get_subject() s,   -- Subjekt
  f.triple.get_property() p,  -- Prädikat
  f.triple.get_object() o    -- Objekt
from family_rdf_data f
/

```

Listing 2

Listing 2 zeigt beispielhaft drei Triples.

### Neues Wissen generieren

Das Speichern von Triples ist aber nur die halbe Sache. Wirklichen Mehrwert bringt das regelbasierte Ableiten von neuen Informationen. Nehmen wir die zwei folgenden Aussagen an:

- Matt ist der Vater von Tom
- Tom ist der Vater von Kevin

Dann kann daraus geschlossen werden, dass auch folgende Aussage gilt: Matt ist ein Großelternteil von Kevin. Diese Aussage selbst ist nicht explizit in den Triples enthalten, aber durch eine selbstdefinierte Regel ohne Weiteres ableitbar. Dazu wird zunächst eine sogenannte „Rulebase“ als Container für Regeln in Bezug auf das Family Model angelegt. Hinzu kommt dann eine GrandParent-Regel (siehe Listing 3). Listing 4 zeigt die Struktur jeder Rulebase.

Für die weitere Verwendung muss die Regelbasis noch indiziert werden. Es wird ein sogenanntes „Entailment“ angelegt (siehe Listing 5). Nun ist der Moment gekommen, die Triples zu durchsuchen und daraus neue Informationen abzuleiten. Auf Basis der zuvor definierten Regel soll mit einem zusätzlichen Filter herausgefunden werden, wer Großvater welchen Enkelkinds ist. Dazu dient ein „SELECT“-Statement mit der Tabellen-Funktion „SEM\_MATCH“ (siehe Listing 6). Tabelle 1 zeigt das Ergebnis in Tabellenform, Abbildung 2 grafisch insgesamt neun abgeleitete Großvater-Enkelkind-Beziehungen.

Der Prozess des Ableitens von neuen Beziehungen heißt auch „Inferencing“. Weiterführende Funktionen in „Spatial and Graph“ verknüpfen das in Form von Triples gespeicherte Wissen mit relationalen Daten. Dazu kommt der „SEM\_RELATED“-Operator zum Einsatz.

Ein ebenfalls sehr anschauliches Beispiel ist als „Oracle By Example“-Tutorial (OBE) in der „Oracle Learning Library“ (siehe „[http://www.oracle.com/webfolder/technetwork/tutorials/obe/db/11g/r1/prod/datamgmt/nci\\_semantic\\_network/nci\\_Semantics\\_les01.htm](http://www.oracle.com/webfolder/technetwork/tutorials/obe/db/11g/r1/prod/datamgmt/nci_semantic_network/nci_Semantics_les01.htm)“) verfügbar. Es zeigt Schritt für Schritt, wie domänenspezifisches Wissen – konkret über onkologische und andere Erkrankungen – in Form von Triples modelliert und gespeichert

ist sowie mit Diagnosen in einer Patientenkartei verknüpft werden kann. Die Gesamtheit dieser Triples sind auch als Thesaurus des National Cancer Institute (*siehe „<https://ncit.nci.nih.gov/ncitbrowser>“*) bekannt (*siehe Listing 7*). Ein anderer häufig verwendeter Begriff für domänen-spezifisches Wissen ist „Ontologie“. Dazu in Beziehung gebracht, werden relational modellierte Daten über PatientInnen. Diese sind in der Tabelle „PATIENTS\_DATA“ vorgehalten (*siehe Listing 8*).

Mithilfe eines semantischen Index auf der „DIAGNOSIS“-Spalte lässt sich dann beispielsweise ermitteln, welche Patientin beziehungsweise welcher Patient an einer Unterklasse der Krankheit „Bone Sarcoma“ (Osteosarkom) leidet (*siehe Listing 9*).

Die auf Daten anwendbaren Fragestellungen, die als Graphen modelliert sind, können vielfältiger Natur sein. Die wohl bekanntesten beschäftigen sich mit der Analyse von Beziehungen in sozialen Netzwerken:

1. Wer ist mit wem über wie viele Knoten vernetzt?
2. Wer ist die einflussreichste Person in einem Netzwerk?
3. Wo gibt es eine Häufung von Individuen mit ähnlichen Interessen?

Das sind drei der möglichen Fragestellungen. Antworten darauf geben Graphen-basierte Algorithmen.

### Graphen-basierte Analysen jenseits einer relationalen Datenbank und SQL

Heutige und zukünftige IT-Architekturen orientieren sich immer mehr an immens und schnell wachsenden Datenmengen, die aus unterschiedlichsten Quellen zur Verfügung stehen. Es sind also flexible und skalierbare Architekturen gefragt, die in kurzen Zeiteinheiten und an unterschiedlichen Orten sehr viele Daten aufnehmen, filtern, prozessieren und analysieren können.

Das führt zu Architektur-Komponenten, die jenseits eines idealerweise einheitlichen Unternehmens-Data-Warehouse zu finden sind. Hier kommen zunehmend Technologien wie Hadoop, NoSQL oder Event Processing zum Einsatz, die häufig im Zusammenhang mit dem Thema „Big Data“ genannt werden.

```
SQL> exec sem_apis.create_rulebase('family_rb')
/

SQL> Insert into mdsys.semr_family_rb
values (
'grandparent_rule',
'(?x :parentOf ?y) (?y :parentOf ?z)',
NULL,
'(?x :grandParentOf ?z)',
SEM_ALIASES (
SEM_ALIAS('','http://www.example.org/family/')
)
)
/
```

Listing 3: Rulebase anlegen und mit Regeln füllen

Name	Null	Type
RULE_NAME	NOT NULL	VARCHAR2(30)
ANTECEDENTS		VARCHAR2(4000)
FILTER		VARCHAR2(4000)
CONSEQUENTS	NOT NULL	VARCHAR2(4000)
ALIASES		RDF_ALIASES

Listing 4

```
exec sem_apis.drop_entailment('rdfs_rix_family')
/

SQL> begin
sem_apis.create_entailment(
'rdfs_rix_family',
sem_models('family'),
sem_rulebases('RDFS','family_rb'));
end;
/

-- Statistiken
SQL> exec sem_apis.analyze_entailment('rdfs_rix_family')
/
```

Listing 5: Regelbasis indizieren

```
SQL> select x "Großvater", y "Enkelkind"
from table(sem_match(
'(?x :grandParentOf ?y . ?x rdf:type :Male)',
sem_models('family'),
sem_rulebases('RDFS','family_rb'),
sem_aliases(sem_alias('','http://www.example.org/family/'),
null))
/
```

Listing 6: Abfrage mit Inferencing

```
SQL> select count(*) from nci_rdf_data;

COUNT(*)
-----
408997
```

Listing 7: Anzahl der Triples im NCI-Thesaurus

```
describe patients_data
/
-- Ergebnis:
Name                               Null?      Type
-----
ID                                   NOT NULL   NUMBER
NAME                                             VARCHAR2 (35)
AGE                                               NUMBER
DIAGNOSIS                                       VARCHAR2 (200)
DIAG_LENGTH                                       NUMBER
```

Listing 8: Triple-Tabelle für den NCI-Thesaurus

```
select distinct diagnosis, sem_distance(123)
from patients_data
where sem_related(
  diagnosis,
  '<http://www.w3.org/2000/01/rdf-schema#subClassOf>',
  '<http://www.mindswap.org/2003/nciOncology.owl#Bone_Sarcoma>',
  sem_models('nci'),
  sem_rulebases('owlprime'), 123) = 1
order by sem_distance(123) asc
/
```

Listing 9: Semantische Analyse von Diagnosedaten im Zusammenhang mit dem NCI-Thesaurus

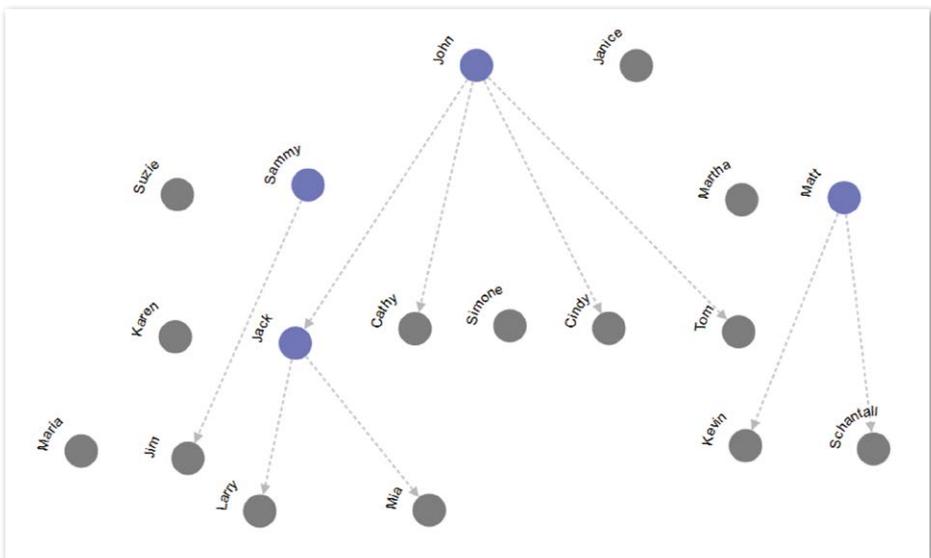


Abbildung 2: Abgeleitete Großvater-Enkelkind-Beziehungen

Großvater	Enkelkind
http://www.example.org/family/Jack	http://www.example.org/family/Larry
http://www.example.org/family/Jack	http://www.example.org/family/Mia
http://www.example.org/family/John	http://www.example.org/family/Cathy
http://www.example.org/family/John	http://www.example.org/family/Cindy
http://www.example.org/family/John	http://www.example.org/family/Jack
http://www.example.org/family/John	http://www.example.org/family/Tom
http://www.example.org/family/Matt	http://www.example.org/family/Kevin
http://www.example.org/family/Matt	http://www.example.org/family/Schantall
http://www.example.org/family/Sammie	http://www.example.org/family/Jim

Tabelle 1

Eine eigene Kategorie innerhalb der NoSQL-Datenbanken sind die sogenannten „Graph-Datenbanken“. Oracle hat mit der NoSQL-Datenbank und deren Aufsatz „Graph“ eine solche im Portfolio. Dieses Produkt ist zudem integraler Teil eines anderen Produkts – Oracle Big Data Spatial and Graph.

Die nachfolgenden Ausführungen beschreiben, wie in der Oracle-NoSQL-Datenbank ein Graph bestehend aus Knoten und Kanten mit zusätzlichen Attributen angelegt wird, die ein soziales Netzwerk abbilden, und wie dieses Netzwerk mit bereits vorgefertigten Funktionen im Hinblick auf die einflussreichsten Personen analysiert wird. Der für die Analyse verwendete Algorithmus ist „Page Rank“. Er wird unter anderem verwendet, um die Link-Popularität einer Web-Ressource (ein Link, ein Dokument etc.) zu ermitteln.

Das Code-Beispiel wurde in Anlehnung an das Posting „Identifying Influencers with the Built-In Page Rank Analytics in Oracle Big Data Spatial and Graph“ (siehe „[https://blogs.oracle.com/bigdataspatialgraph/entry/identifying\\_influencers\\_with\\_the\\_built](https://blogs.oracle.com/bigdataspatialgraph/entry/identifying_influencers_with_the_built)“) im Blog „Adding Location and Graph Analysis to Big Data“ erstellt und auf der Oracle Big Data Lite Virtual Machine v4.2.1 getestet. Dazu kommt eine für Graphen spezifische Skriptsprache zum Einsatz (siehe Listing 10 und Abbildung 3).

Oracle Big Data Spatial and Graph besitzt eine Engine, die parallel und In-Memory Graph-Analysen wie beim Page Rank ausführt, und zwar sowohl auf der Oracle-NoSQL-Datenbank als auch auf Apache HBase. Grundlage dafür ist PGX, ein sogenanntes „Graph Analysis Framework“ (siehe „<http://www.oracle.com/technetwork/oracle-labs/parallel-graph-analytics/overview/index.html>“), über das mehr als dreißig gängige Graphen-Algorithmen für die Bereiche „Classical“, „Community Detection“, „Path Finding“, „Ranking“, „Recommendation“ und „Structure Evaluation“ implementiert sind, die mit 107 und mehr Knoten und Kanten umgehen können. Anstelle von SQL wird über APIs wie Java, Gremlin oder Python auf die Daten zugegriffen.

**Fazit**

Die in ihrem Ursprung relationale Oracle-Datenbank hat sich zu einem multi-modalen Datenbanksystem entwickelt. Die Erweiterung als Graph-Datenbank, die Implementie-

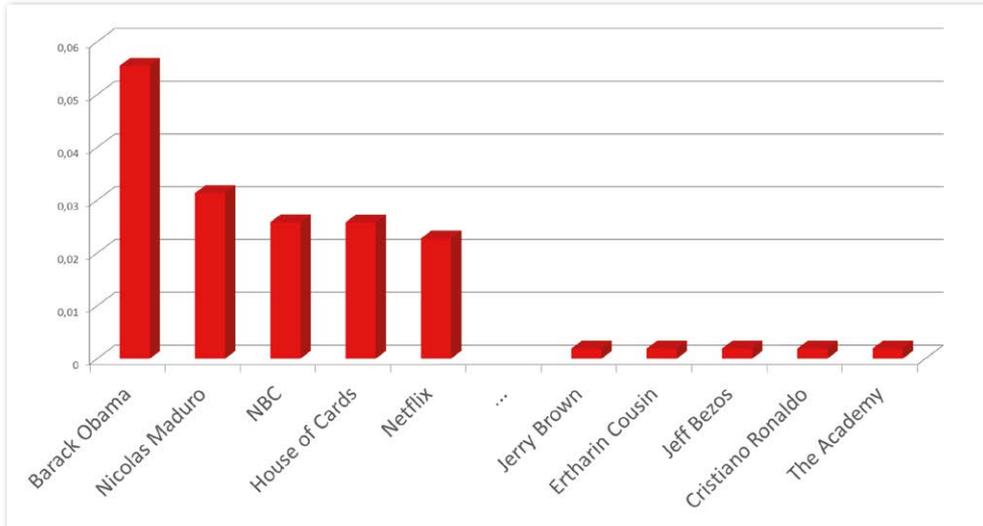


Abbildung 3: Ergebnis der Einfluss-Analyse im Netzwerk mit Page Rank

Die Unterstützung Objekt-relationaler Datentypen sowie die Unterstützung Dokument-zentrischer, semi-strukturierter Daten wie XML oder JSON sind zum Teil längst schon Realität und Bestandteil vieler Anwendungen.

SQL als Standard-Abfragesprache wurde und wird in geeigneter Weise erweitert, um direkt in der Datenbank auf all diese Daten in einheitlicher Art und Weise zuzugreifen und sie in ihren Zusammenhängen zu analysieren. Aber die Anforderungen und Technologien entwickeln sich weiter. Daher lohnt es sich, den eigenen Blick über die bekannten und bewährten Technologien im Umfeld von Data Warehouses hinaus zu richten. NoSQL, Hadoop & Co. lassen sich gut in bestehende IT-Infrastrukturen integrieren oder als Cloud Services nutzen. Bereits in naher Zukunft werden kaum mehr Wege daran vorbeiführen. Daher gilt es, sich bald das neue Wissen zu erschließen und es bestmöglich für das eigene Unternehmen einzusetzen.

```
//
// Connect to the Oracle NoSQL Database
//
server = new ArrayList<String>();
server.add("bigdatalite:5000");
cfg = new PgNosqlGraphConfigBuilder() \
    .setDbEngine(DbEngine.NOSQL) \
    .setName("connections") \
    .setStoreName("kvstore") \
    .setHosts(server) \
    .addEdgeProperty("lbl", PropertyType.STRING, "lbl") \
    .addEdgeProperty("weight", PropertyType.DOUBLE,
"1000000") \
    .setMaxNumConnections(1).build();

//
// Get an instance of OraclePropertyGraph
//
opg = OraclePropertyGraph.getInstance(cfg);
opg.clearRepository();

//
// Use the parallel data loader API to load a
// sample property graph in flat file format with
// a degree of parallelism (DOP) 2
//
vfile="../../data/connections.opv"
efile="../../data/connections.ope"
opgdl=OraclePropertyGraphDataLoader.getInstance();
opgdl.loadData(opg, vfile, efile, 2);

// Read through the vertices
opg.getVertices();

// Read through the edges
opg.getEdges();

//
// Use In-Memory Graph Analysis Framework to run
// the built-in analytical function, PageRank, to
// identify influencers
//
```

Listing 10



Karin Patenge

karin.patenge@oracle.com

<http://oracle-spatial.blogspot.com>

```
analyst = opg.getInMemAnalyst();
rank = analyst.pagerank(0.0001, 0.85, 100).get();
// maximum error for terminating the iteration: e =
0.0001
// damping factor: d = 0.85
// maximum number of iterations: max = 100

rank.getTopKValues(5);

// Identify person at the given vertex
v1=opg.getVertex(11);
v2=opg.getVertex(601);
v3=opg.getVertex(421);

System.out.println("Top 3 influencers: \n " +
v1.getProperty("name") + \
"\n " + v2.getProperty("name") + \
"\n " + v3.getProperty("name"));

// Top: Barack Obama
```

# SPARC-Server in hyperkonvergenten VM-Infrastrukturen

Bert Miemietz, OSL Gesellschaft für offene Systemlösungen mbH

Webzentrierte Applikationen und Cloud-Services setzen Maßstäbe, denen sich das eigenbetriebene Rechenzentrum trotz typischerweise klassischer Applikationsstacks heute stellen muss. Die notwendige Schnelligkeit, Flexibilität und Effizienz ist allein mit Software-definierten Infrastrukturen erreichbar. Hyperkonvergente Systeme gelten dabei als besonders fortschrittlich. Der Artikel gibt einen kurzen Überblick über mögliche Architekturen, bemüht sich um eine realistische Bewertung der Einsatzszenarien und zeigt an einem Beispiel auf, dass moderne SPARC-Systeme auch in hyperkonvergenten Systemen eine höchst nutzbringende Rolle spielen können.

Was Branchenexperten und Analytiker bereits vor Jahren prognostizierten, ist inzwischen erlebbare Realität: Angesichts zunehmender Möglichkeiten, alles als Dienst aus der Cloud beziehen zu können („Everything as a Service“), gerät das Rechenzentrum im Eigenbetrieb unter Druck. Mit der Notwendigkeit, weiterhin klassische Applikationen und Dienste bereitzustellen, mit der hundertprozentigen Kontrolle über die Daten sowie über die Verfahren zu ihrer Verarbeitung und Verwertung (damit einhergehend auch größerer Flexibilität in der Anpassung an Geschäftsmodelle) und nicht zuletzt mit dem Datenschutz gibt es aber auch in Zukunft starke Argumente für das eigene oder per Outsourcing betriebene Rechenzentrum. Wer damit erfolgreich sein will, muss sich jedoch auch an durch die Cloud gesetzten Standards messen. Dabei geht es in Bezug auf Bereitstellung und Betrieb von RZ-Infrastrukturen primär um:

- Flexibilität, Geschwindigkeit und Verfügbarkeit
- Reduzierte Komplexität
- Integration
- Kostensenkung

Die reine, meist mit dem Argument „Konsolidierung“ eingeführte Server-Virtualisierung genügt, für sich allein genommen, diesen Anforderungen bei Weitem nicht mehr.

## Das Lösungsversprechen

Auf der Suche nach Lösungen wird früher oder später klar, dass es nicht nur um Technologien, sondern vorrangig um veränderte Organisation und Prozesse geht, und dass schlussendlich bestimmte Strukturen und Aufgabenfelder sogar wegfallen. So verwundert es nicht, wenn gerade größere Einheiten eine stärkere Affinität zur Private Cloud mit OpenStack zeigen: Neben möglicherweise nachvollziehbaren technischen Gründen steht auch im Raum, dass vielgliedrige Organisationsformen (etwa Storage, SAN, Server, Netzwerk, Backup, Monitoring) hier ein neues Zuhause finden können, gegebenenfalls um den Preis geringfügiger Transformationen.

Wesentlich stärker wirksam wird der Sog der Cloud bei kleineren Einheiten. Eine durchgreifende Vereinfachung von Organisation und Prozessen ist weniger ein Politikum; Reduzierung der Komplexität, schnelle Ergebnisse und einfache Lösungen stehen klar im Vordergrund. Gestützt wird dies durch eine unterschwellige Frustration über die zunehmende Komplexität hochverfügbarer und agiler Infrastrukturen und über ein gerade für kleinere Anwender fragwürdiges Preis-Leistungsverhältnis vor allem in der Storage-Infrastruktur (RAID, SAN). Dies wird spätestens spürbar, wenn es um höhere Performance oder höhere Verfügbarkeiten geht, etwa in Disaster-Recovery-Konfigurationen.

Als Lösungsmöglichkeit für reduzierte Komplexität werden das Software-definierte Rechenzentrum (SDDC) sowie speziell konvergente und hyperkonvergente Infrastrukturen gehandelt. Doch was verbirgt sich dahinter? Wie so häufig ist trotz inflationärer Verwendung der Begriffe die Definitionslage nicht einheitlich, es kristallisiert sich aber in etwa folgendes Verständnis heraus: Konvergente Systeme können – vereinfacht gesagt – als die Bündelung mehrerer IT-Komponenten in einer ganzheitlichen, optimierten IT-Lösung verstanden werden (Vorkonfektionierung und Abstimmung). Beispielhaft seien hier die Vblocks von VCE, UCS von Cisco/NetApp und die Private Cloud Appliance (früher Virtual Compute Appliance) von Oracle [1] genannt, deren herausragendes Merkmal der aus Xsigos I/O Director hervorgegangene Oracle Fabric Interconnect ist. Dabei handelt es sich um eine Appliance, die intern ein konvergentes Netzwerk (LAN+SAN) zum Anschluss der Compute-Nodes ermöglicht und gleichzeitig die Brücke zur Anbindung an die Außenwelt schlägt. Von Oracle als „Converged System“ positioniert, markiert es zugleich den Übergang zur nächsthöheren Konvergenzstufe, den hyperkonvergenten Systemen, ohne dass dabei in jedem Fall eine klare Abgrenzung möglich wäre.

„Hyperkonvergenz“ bezeichnet eine Software-zentrierte Architektur, die Mas-

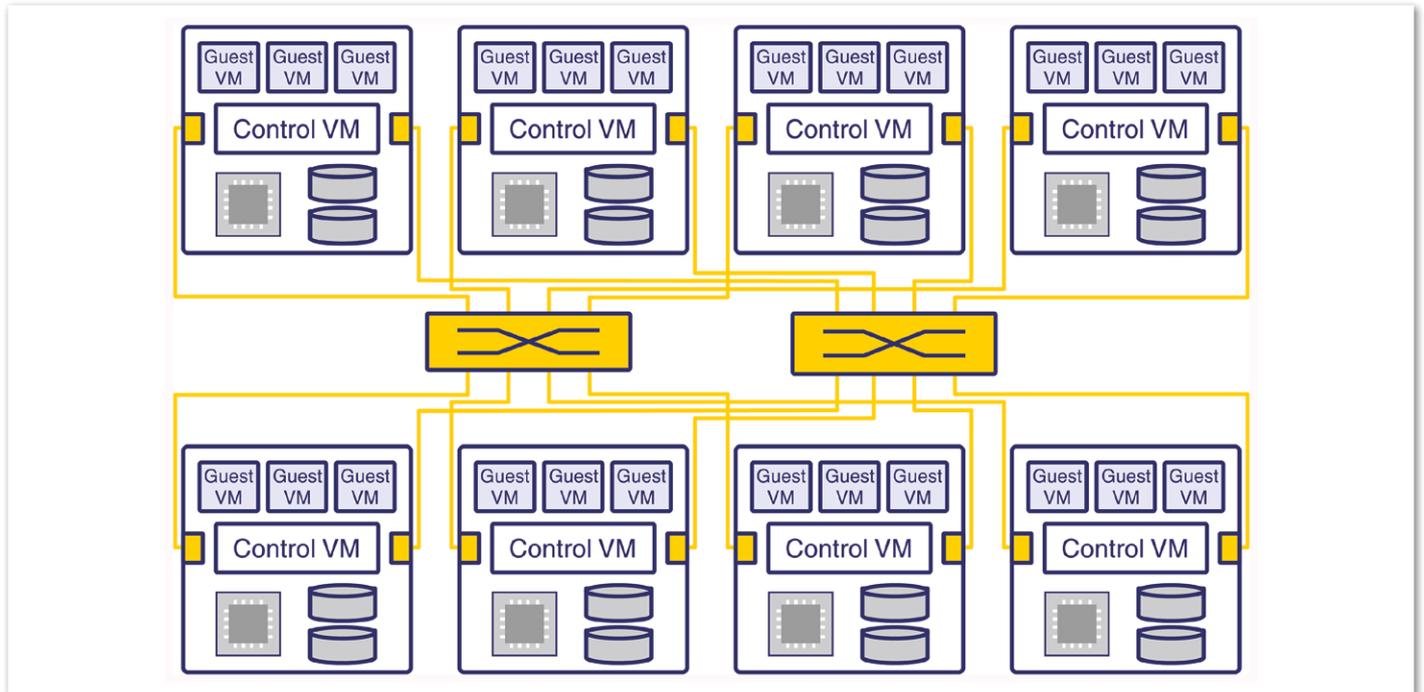


Abbildung 1: Beispiel für eine Hyperkonvergente Infrastruktur nach dem Brick-Modell

senspeicher, Netzwerk und Server-Virtualisierung in einem von Grund auf integrierten Paket implementiert und die Verwaltung als ein einziges System ermöglicht.

Die entscheidenden Merkmale sind also Softwarezentrierung und die Darstellung als ein einziges System. Daneben zeichnen sich alle in diese Kategorie einzuordnenden Systeme durch ein mehr oder weniger konvergentes Netzwerk aus. Schaut man näher auf die Architektur der

heute verfügbaren hyperkonvergenten Systeme, so gibt es derzeit am Markt zwei unterschiedliche Ansätze.

### Das Brick-Modell

Der Grundgedanke ist bestechend einfach: Alle Ressourcen (Compute, Network, Storage) sind in einem Baustein (Brick oder Building Block) zusammengefasst – im Prinzip nicht anders als in einem PC oder Industrie-Standardserver. Kapazität, Performance und Verfügbarkeit skalieren dann über das

Vernetzen mehrerer Bricks entsprechend dem Bedarf und die Software sorgt dafür, dass alles wie ein einziges System funktioniert (siehe Abbildung 1).

Diesem Konzept folgen beispielsweise Nutanix, SimpliVity und VMware mit „EVO:RAIL“. Inspiriert wurde diese Entwicklung durch Architekturen für Web-Dienste, wie sie Google und Amazon anbieten, zum Teil adaptiert man auch vergleichbare Technologie-Konzepte wie das NDFS als Neu-Interpretation des Google-Filesys-

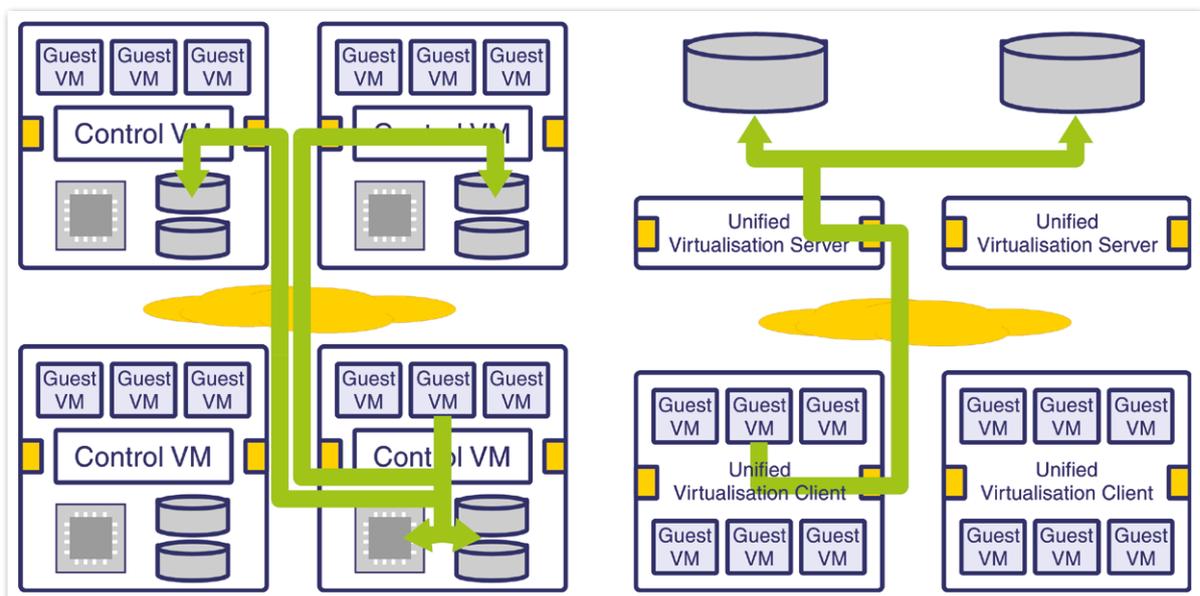


Abbildung 2: Die Lösung des Replikationsproblems ist im Brick-Modell (links) deutlich anspruchsvoller als im Client-Server-Modell (rechts)

	Nutanix	SimpliVity	VMware	OSL
Produktbezeichnung	NX Series	OmniCube	EVO:RAIL	Unified Virtualisation Environment
Architektur	Brick	Brick	Brick	Client/Server
Minimale Anzahl Server	–	–	–	1
Minimale Anzahl Compute-Nodes	3	1	4	1
Maximale Anzahl Compute-Nodes	- / typ. 64	8/32	32	120
Skalierungsstufe (Nodes)	1	1	4	1
Compute-Node-Connectivity	1GE/10GE	1GE/10GE	10GE	1GE/10GE/IB
Hypervisors	V-Sphere, Hyper-V, KVM	V-Sphere, KVM	V-Sphere	KVM, Xen, VirtualBox
Interne Storage-Protokolle	NFS, SMB, iSCSI	NFS	VSAN	RSIO, iSCSI, SMB, NFS
Externe Storage-Anbindung	NFS, iSCSI	NFS, iSCSI	NFS, iSCSI	FC, FCoE, NFS, iSCSI
Storage-Architektur	VSA per Node	VSA per Node	VSphere	UVS+RSIO

Tabelle 1

tems. Aber natürlich ist der Betrieb von virtuellen Maschinen oder gar OLTP-lastigen Anwendungen etwas deutlich Anderes als der von Web-Diensten. So werben die Hersteller denn auch mit einer Vielzahl von technologischen Feinheiten, angefangen von ausgeklügelten Replikations-Mechanismen und verteilten Datei-Systemen über spezielle Lösungen für den Zugriff auf Metadaten bis hin zu ausgefeilten Caching-Konzepten und natürlich den massiven Einsatz von SSDs. Einige dieser Technologien bedeuten in der Praxis einen tatsächlichen Zugewinn, wie es SimpliVity mit seinen Lösungen für verteilte Standorte demonstriert. In vielen anderen Punkten wird deutlich, dass das Öfteren Probleme gelöst werden, die spezifisch aus der gewählten Architektur folgen.

Überhaupt ist das Brick-Modell nicht so ganz neu: Bereits vor zwanzig Jahren gab es einen ähnlichen Hype um das „Massive Parallel Processing“ und die Umsetzung dieses Ansatzes für die kommerzielle IT. IBM SP2 und vor allem Pyramid „Meshine“ wiesen vergleichbare Architekturmerkmale zu den heutigen hyperkonvergenten Systemen auf und kämpften bereits mit den schwierigeren Seiten dieses Konzepts:

- **Storage-Redundanz**

Im klassischen Rechenzentrum stehen den Herausforderern extrem skalierbare, performante und verfügbare Konkurrenten (Enterprise-Storage) gegenüber. Gegen diese Spezialisten mit allein durch Software umgewidmeter

Universal-Hardware anzutreten, ist ein ehrgeiziges Unterfangen, die Erfolge bleiben derzeit auch auf ausgewählte Segmente beschränkt. Hauptproblem ist das Schaffen performanter Redundanzen, was zumeist über das komplette Spiegeln größerer „Data-Chunks“ geschieht. RAID-5 erweist sich für die Inter-Node-Kommunikation nach wie vor als zu aufwändig, heute genauso wie zu Pyramids Zeiten. Auch mit „1:n“-Beziehungen stellt sich weiterhin die Frage, wie mit dem Ausfall mehrerer Knoten umgegangen wird, etwa bei einer Störung der Stromversorgung. Zudem belastet die Replikation auch die Netz-Anbindung vermeintlich unbeteiligter Knoten (siehe Abbildung 2). Es gibt also eine Vielzahl von Schwierigkeiten im Storage-Kontext – die Lösungsangebote unterscheiden sich je nach Hersteller.

- **I/O-Performance**

Hier hängen die Ergebnisse stark von den Lastprofilen ab. Was als Architektur für Big Data und Web-Applikationen passt, bringt bei klassischen OLTP-Anwendungen möglicherweise Nachteile. Die Brick-Architekten reagieren darauf mit dem Einsatz von SSDs. Spätestens die Replikation frisst jedoch bei etlichen Anwendungen einen Großteil des Gewinns wieder auf, sodass die Performance klassischer Architekturen mit konventionellen RAID-Systemen und Festplatten weiterhin die Referenz bleibt.

- **Fat Nodes**

Wie bindet man besonders leistungsfähige Compute- oder Storage-Nodes ein? Die Gründe dafür können vielfältig sein, die Lösungsangebote sind es derzeit (noch) nicht.

- **Backup**

Wo greift man in einer vollsymmetrischen, verteilten Storage-Architektur sinnvoll ein Backup ab, ohne dass es zu punktuellen Überlastungen kommt? Gerade für große Datenbanken ist dies eine nicht zu unterschätzende Herausforderung.

- **DR-Tauglichkeit**

Um für ein Disaster Recovery gewappnet zu sein, sind besondere Konfigurationen erforderlich. Im Vergleich zu Metro-Clustern beziehungsweise Hochverfügbarkeits-Konfigurationen von Enterprise-RAID-Systemen verliert manche hyperkonvergente Brick-Lösung in Sachen „Einfachheit“ wieder etwas von dem ihr eigenen Charme.

Generell spiegelt sich die Verwendung von Standard-Hardware im Moment noch nicht in attraktiven Preisen wider, sodass sich die Wachstumsschritte zwischen den Ausbaustufen für manchen Anwender als (zu) groß erweisen. Neben den bekannten architekturenspezifischen Besonderheiten dürfte dies ein weiterer Grund dafür sein, dass ohne (heute noch nicht konkret erkennbare) Revolutionen technisch motivierte Anwendungsfälle für das Brick-Mo-

dell begrenzt bleiben könnten. Eine starke Position nehmen diese Lösungen jedoch schon heute bei VDI-Konzepten ein.

**Das Client-Server-Modell**

Der deutsche Speicher-Virtualisierungs- und Cluster-Spezialist OSL geht als derzeit einziger Anbieter einen komplett anderen Weg und organisiert sein „OSL Unified Virtualisation Environment“ in einem Client-

Server-Modell. Der Unified Virtualisation Server (UVS) liefert virtualisierte Storage- und Netzwerk-Ressourcen, Cluster- und VM-Control-Dienste; als Skalierungsfaktor wirkende Compute-Nodes (Unified Virtualisation Clients) docken an diese an. Im Storage-Backend kann sowohl mit UVS-internen Platten (wie SSD) als auch mit extern angeschlossenen RAID-Systemen gearbeitet werden, die Compute-Nodes sind HCI-

typisch über ein konvergentes Netzwerk angeschlossen. Das System wartet dabei mit einer hohen Integration von Speicher-Virtualisierung und Netzwerk-Technologie auf, das RSIO-Protokoll erlaubt insbesondere eine hochskalierbare und einfache Bereitstellung von virtuellen Block-I/O-Devices für die UVCs und die Gastsysteme. Im Vergleich zum Brick-Konzept punktet das Client-Server-Modell mit der Fähigkeit,

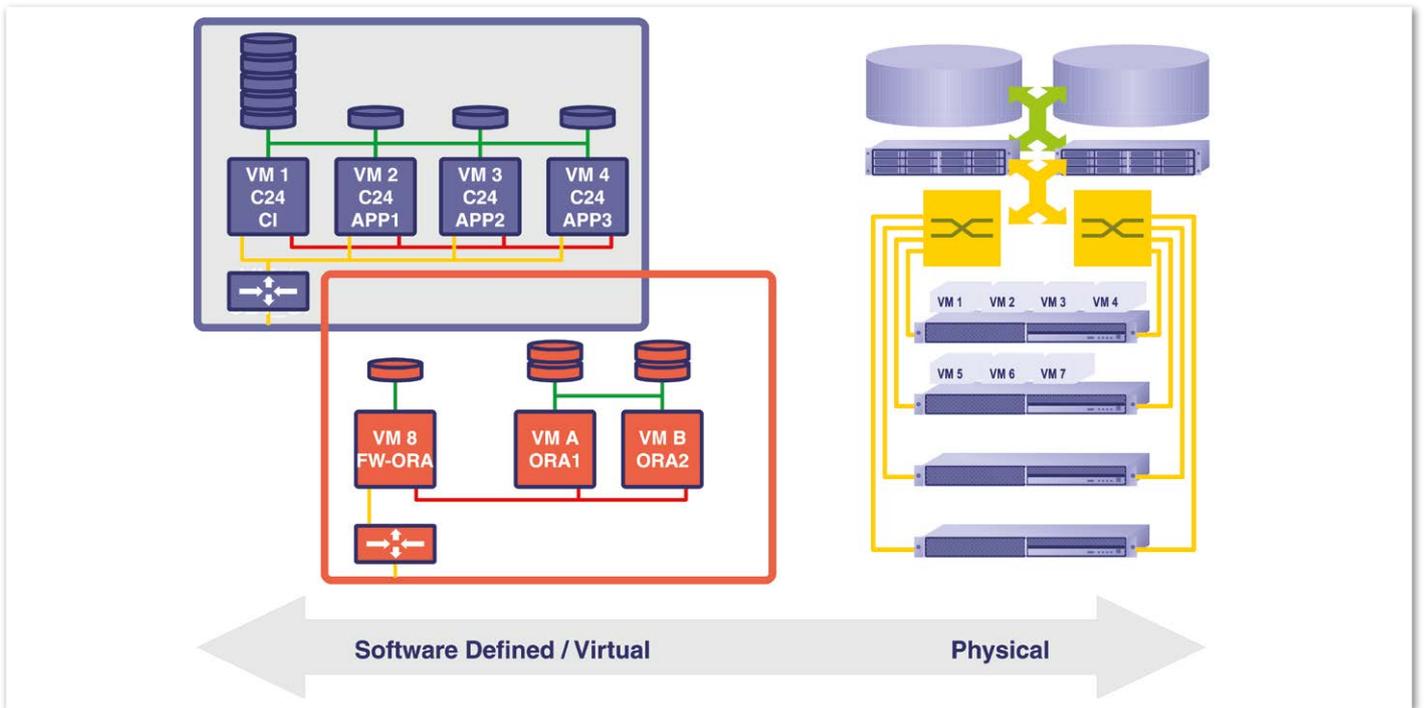


Abbildung 3: Neue Infrastrukturen werden im Virtualisierungs-Layer per Software auf einer immer gleichen physischen Hardware-Konfiguration erzeugt

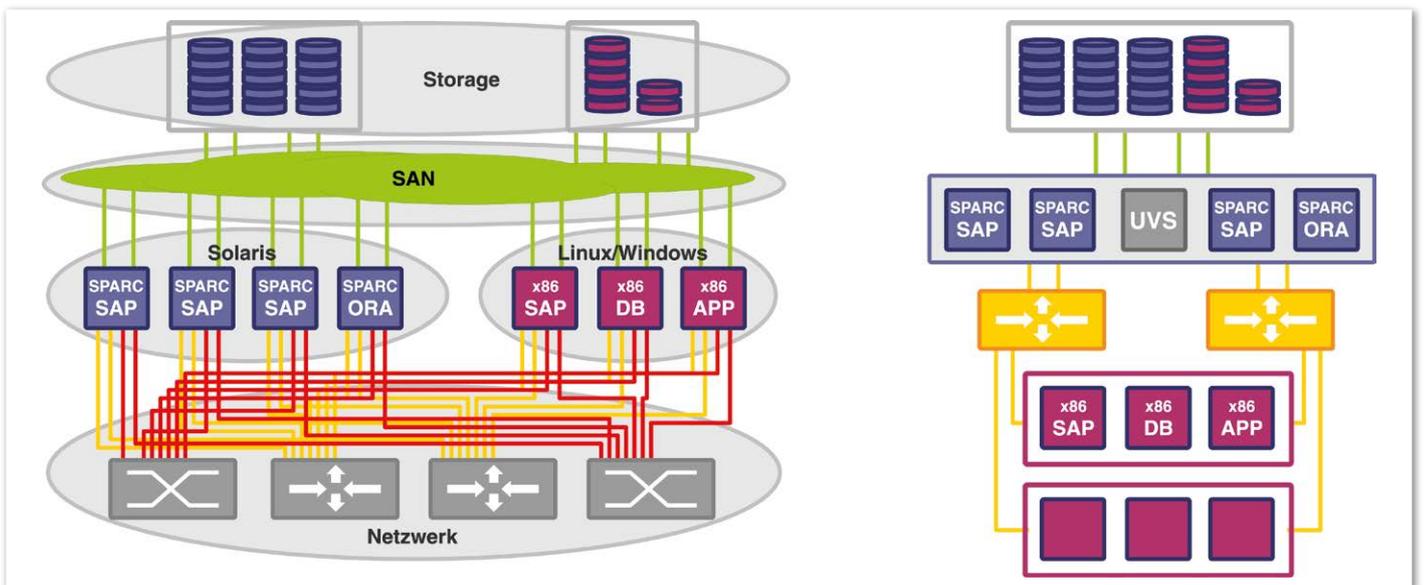


Abbildung 4: Hyperkonvergente Systeme (rechts konsolidiertes SPARC-System plus x86-basierte UVC) bedeuten im Vergleich zu einer klassischen Landschaft (links) nicht nur eine schlanke und aufgeräumte Infrastruktur, sondern integrieren vielfältige administrative Aufgabenfelder in einem vereinfachten und einheitlichen Prozess

(vorhandene) leistungsfähige und preisgünstige (Spezial-) Systeme einzubinden. Das gilt für Compute-Nodes, aber noch viel stärker für Speichersysteme. So ist hier tatsächlich ein sehr breites Spektrum von Speicher- und Serversystemen integrierbar, ausdrücklich auch vorhandene Komponenten – ein nicht unwesentlicher Faktor für Flexibilität und Kostensenkungen. *Tabelle 1* gibt zu beiden Architektur-Modellen einen Überblick über ausgewählte Produkte und deren Implementierungsdetails [2, 3].

### Das Software-definierte Rechenzentrum (SDDC)

Allen vorgestellten Lösungen ist gemein, dass sie komplette – auch sehr unterschiedliche RZ-Infrastrukturen per Software darzustellen vermögen. Das beschleunigt die Provisionierung, senkt durch eine bessere Auslastung Kosten, reduziert Fehlerquellen und vereinfacht Administration sowie interne Abläufe erheblich. Hosts und Netzwerk-Komponenten realisieren sich in entsprechenden Software-Objekten, wie die nachfolgende, am OSL UVE orientierte Grafik in *Abbildung 3* zeigt.

Eine Besonderheit am gewählten Beispiel ist, dass auf der Block-I/O-Seite durch das Zusammenspiel von Speicher-Virtualisierung und RSIO der klassische Administrations-Stack von LUNs über das SAN bis zum Block-Device der nutzenden OS-Instanz entfällt. Für alle vorgestellten Lösungen gilt jedoch, dass hyperkonvergente Systeme neue Aufgaben und Architekturen per Software und ohne Änderungen am physischen Setup umsetzen können.

### SPARC-Server und x86-VM-Infrastrukturen

Das Software-zentrierte Client-Server-Konzept des OSL UVE erlaubt es auch, dass SPARC-Systeme (auch einzelne LDOMs) als Unified Virtualisation Server für x86-basierte Compute-Nodes (UVC) agieren können [7]. Das ist interessanter, als es auf den ersten Blick erscheinen mag. Für den Betrieb von unternehmenskritischen Applikationen (beispielsweise SAP und/oder Oracle-Datenbanken) ist zumeist eine leistungsfähige und hochverfügbare Infrastruktur (Storage, Server, Netzwerk) vorhanden. Wenn diese – quasi nebenbei – eine x86-VM-Infra-

struktur treiben kann, übertragen sich diese Performance- und Verfügbarkeitsattribute ohne erhebliche Mehrkosten auch auf die VM-Welt.

Eine Schlüsselrolle spielt dabei die enorme Leistungsfähigkeit aktueller SPARC-Systeme (wie T5 und M10), die noch vor fünf Jahren unvorstellbare Konsolidierungsmöglichkeiten eröffnete. So findet man heute zum Teil auch extreme Konstellationen, in denen ein halbes Server-Rack auf nur noch eine Höheneinheit schrumpft. Herausragende Faktoren sind dabei:

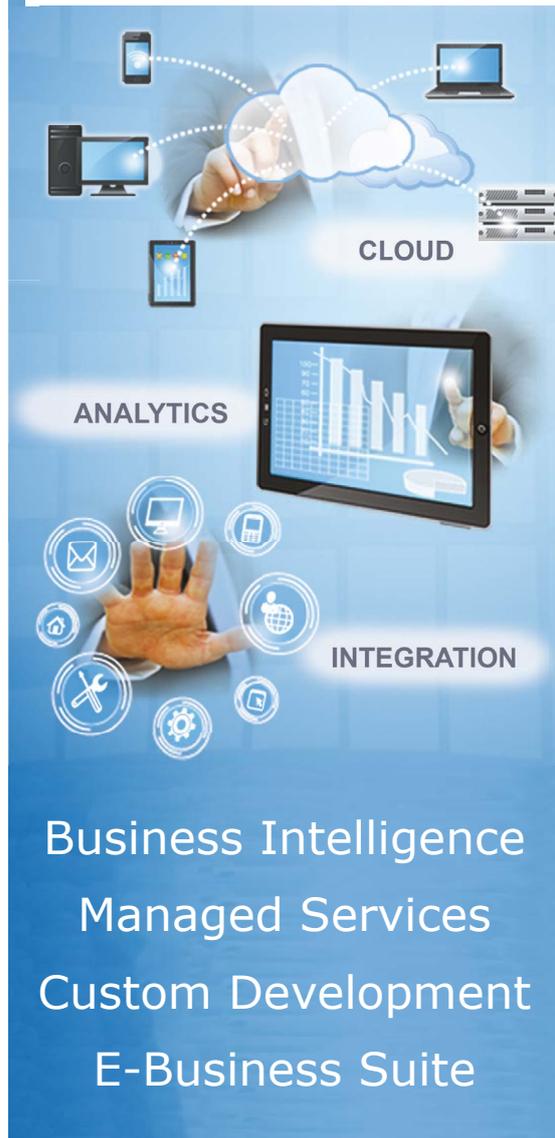
- Die große Zahl von Cores und Threads in einem System
- Die gegenüber früheren Systemen deutlich erhöhte Taktrate
- Die Reife von LDOMs als überlegene, feingranulare Virtualisierungs-Technologie
- Überzeugende Möglichkeiten zum I/O-Sharing (Direct I/O und SR-IOV)

Hinzu kommt: Für kleinere und mittelgroße Rechenzentren ist heute der Verzicht auf ein SAN realistisch. Moderne Enterprise-RAID-Systeme verfügen mindestens über vier FC-Ports (oft deutlich mehr), an die sich zwei oder mehr konsolidierte SPARC-Server problemlos direkt und redundant anschließen lassen. So ist auf einfachste Weise ein beeindruckendes Spektrum an Performance und Verfügbarkeit abzudecken. Zudem verfügen die SPARC-Server über ausreichend Performance, um in einer LDom einen Unified Virtualisation Server laufen zu lassen, der seinerseits über das Unified Virtualisation Network eine komplette x86-VM-Infrastruktur bedienen kann (*siehe Abbildung 4*).

Hyperkonvergente Systeme mit ihrer Fähigkeit, komplette Infrastrukturen wie ein einziges System zu managen, bringen so auf natürliche Weise zusammen, was sonst typischerweise in dedizierten Projekten mit dedizierter Hardware und dedizierten Teams dargestellt wird. Dieser organisatorische Vorteil ist der ausschlaggebende Punkt bei der überlegenen Agilität einer hyperkonvergenten Infrastruktur (*siehe Abbildung 5*).

### Fazit

Hyperkonvergente Infrastrukturen (HCI) erlauben bereits heute radikale Vereinfachungen und zugleich eine erhebliche Steigerung



Business Intelligence  
 Managed Services  
 Custom Development  
 E-Business Suite

Danke für Ihren Besuch bei der DOAG 2015!

2015  
**DOAG**  
 Konferenz + Ausstellung

**Apps Associates GmbH**

Flughafenring 11  
 D-44319 Dortmund

Tel.: +49 231 22 22 79-0

www.appsassociates.com

**ORACLE®** Platinum Partner

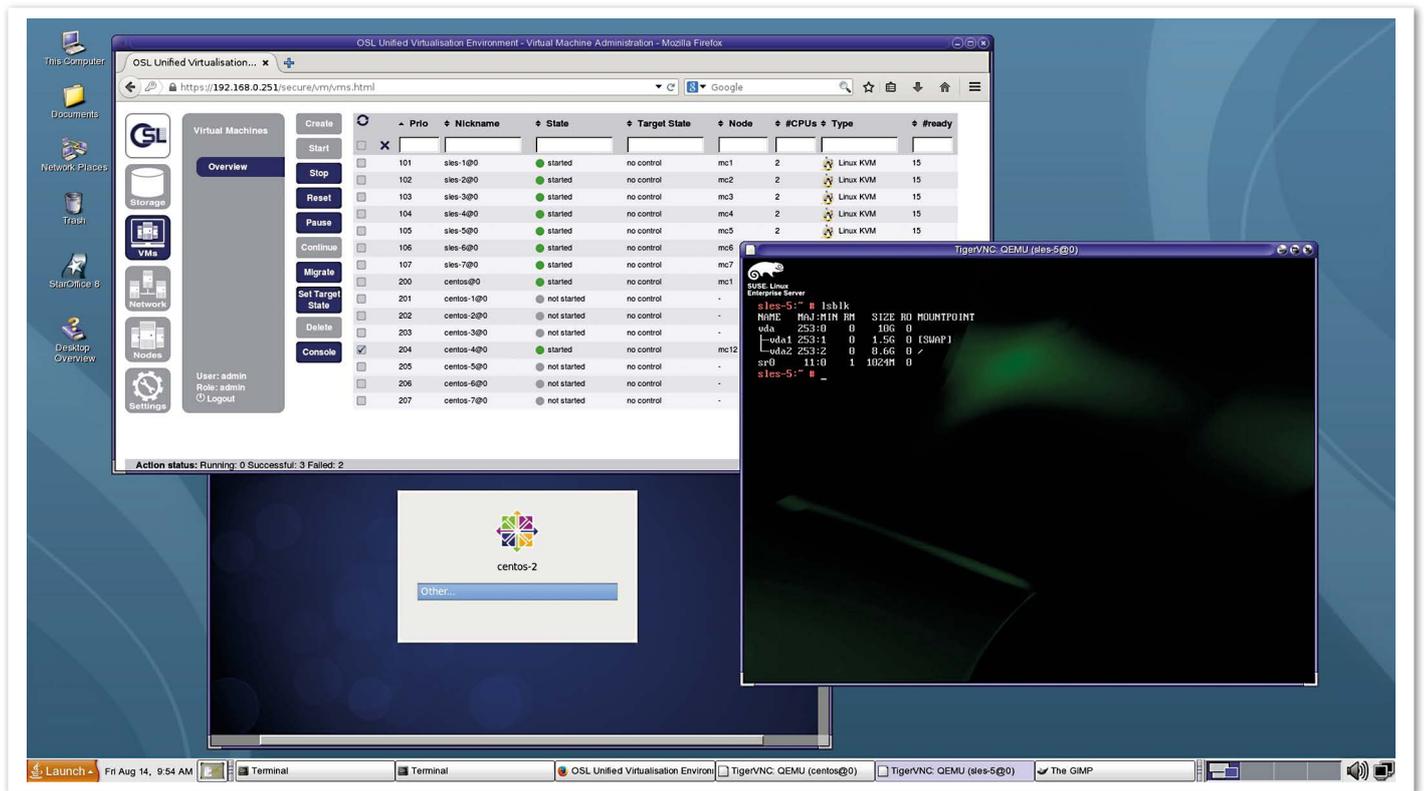


Abbildung 5: Desktop mit der UVE-WebGUI nebst Konsolen virtueller Maschinen

von Geschwindigkeit, Flexibilität und Verfügbarkeit. Dort, wo SPARC-Systeme und die dazugehörige Storage-Infrastruktur das Rückgrat Datenbank- und SAP-orientierter Rechenzentren bilden, können sie mit ihrer Leistungsfähigkeit und ihren herausragenden Verfügbarkeitseigenschaften eine neue Funktion übernehmen: Sie eignen sich auch als zentrale Server für hyperkonvergente, x86-basierte VM-Infrastrukturen. Ganz gleich, ob Brick- oder Client-Server-Modell, erlauben hyperkonvergente Infrastrukturen so die Transition vielgliedriger

Organisationsformen (Solaris, Linux, Windows, VDI, Datenbank, SAP, Backup, Netzwerk, Storage, SAN) zu einem schlanken, integrierten RZ-Prozess.

### Quellen und weiterführende Links

- [1] <http://www.oracle.com/technetwork/server-storage/private-cloud-appliance/overview/index.html>
- [2] <http://www.datacenterzombie.com/hyper-converged-infrastructure-comparison>
- [3] [http://www.osl.eu/solutions/virtualisation/infrastructure\\_g.html](http://www.osl.eu/solutions/virtualisation/infrastructure_g.html)
- [4] <http://www.nutanix.com/resources/reference-architecture>

[5] [https://www.simplivity.com/wp-content/uploads/SIM\\_DS\\_FINAL\\_4.pdf](https://www.simplivity.com/wp-content/uploads/SIM_DS_FINAL_4.pdf)

[6] <https://www.vmware.com/de/products/evorail>

[7] [http://www.osl.eu/news/uv\\_s\\_sparc/uv\\_s\\_sparc\\_g.html](http://www.osl.eu/news/uv_s_sparc/uv_s_sparc_g.html)

Bert Miemietz  
bert.miemietz@osl-it.de

## Carsten Czarski ist DOAG-Botschafter 2015

Der begehrte Preis „DOAG-Botschafter“ geht in diesem Jahr an Carsten Czarski, Senior Leitender Systemberater bei Oracle. Czarski hat sich bei der DOAG insbesondere durch hohes übergreifendes Engagement verdient gemacht. Der erfahrene Systemberater arbeitet seit mehr als 14 Jahren bei Oracle, seit 2002 ist er der DOAG verbunden – und hat seitdem keine DOAG-Konferenz verpasst.

Von Beginn an hat Czarski sich in besonderem Maße in der DOAG engagiert: Als Speaker auf Konferenzen und Community Days ebenso wie als Wissens- und Vertrauensträger, der Anerkennung und volle Unterstützung aus allen Communities genießt. „Innerhalb der DOAG ist Carsten Czarski im Bereich Datenbankentwicklung als DER Wissensträger bekannt. Dieses Wissen stellt er

den Mitgliedern seit mehr als zehn Jahren auf den unterschiedlichsten Plattformen zur Verfügung“, erläutert Dietmar Neugebauer, DOAG Vorstandsvorsitzender, die Entscheidung des Komitees.



# Data Guard auf der ODA

David Hueber, dbi services ltd.

Ziel aller IT-Systeme ist es, für die Endbenutzer und/oder Kunden einen Service zu erbringen. Je nachdem, wie kritisch dieser Dienst ist, werden „Service Level Agreements“ geschlossen, die von der IT-Infrastruktur gewährleistet werden müssen. Damit wird die System-Verfügbarkeit zu einem Schlüsselbegriff beim Aufbau von produktiven IT-Systemen. Die Oracle Database Appliance (ODA) unterstützt mehrere solcher hochverfügbaren Lösungen – eine davon ist Data Guard.

Verfügbarkeit besteht auf drei Ebenen:

- Verfügbarkeit der Hardware
- Verfügbarkeit des Dienstes
- Verfügbarkeit der Daten

Man könnte dieser Liste noch die Verfügbarkeit der Website hinzufügen. Die Verfügbarkeit der Hardware ist im Grunde die Kapazität der zugrunde liegenden Hardware, physische Ausfälle zu unterstützen beziehungsweise zu überdauern. Das reicht von Themen wie „Netzwerk“ über elektrische Ausfälle bis zu Festplattenfehlern. Um diesen Bereich abzudecken, wurde die ODA als hochredundante Architektur ausgelegt, die Lösungen wie Netzwerkbindung, redundante Stromversorgung, mehrere Controller und natürlich ASM-Redundanz umfasst.

Dienst- und Datenverfügbarkeit bei der Oracle-Datenbank schützen im Wesentlichen vor Instanz- und Datenbank-Abstürzen (siehe Abbildung 1). Für die Verfügbarkeit der Dienste integriert die ODA mehrere Out-of-the-Box-Lösungen:

- RAC One Node
- RAC

Diese sind direkt in OAKCLI integriert (siehe Listing 1). Nach der Beantwortung von sechs Fragen und rund dreißig Minuten Verarbeitungszeit erhält man eine voll funktionsfähige Zwei-Knoten-Real-Application-Cluster-Datenbank.

Eine ODA umfasst eine ASM-Redundanz mit normaler oder hoher Redundanz, aber es bleibt ein gemeinsam genutzter Speicher zwischen beiden Knoten, der ausfallen kann

und noch anfälliger für eine logische Korruption ist. Wenn die ODA mit der Oracle Database Enterprise Edition läuft, bedeutet das, dass Data Guard ohne zusätzliche Kosten zur Verfügung steht. Von daher kann

man eine Data-Guard-Konfiguration zwischen zwei ODAs einrichten, um die Verfügbarkeit der Daten und sogar Dienste zu erhalten, wenn man einen Fast Start Failover konfiguriert. Leider hat die ODA die Data-

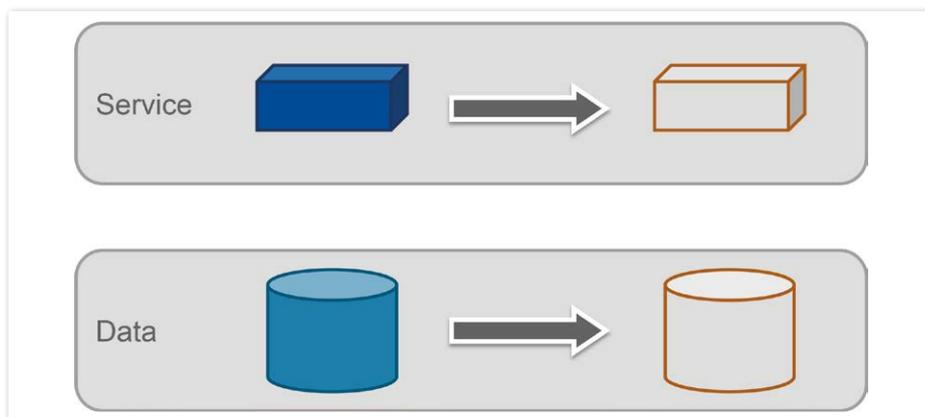


Abbildung 1: Typen der Hochverfügbarkeit

```
Bitte wählen Sie eine der folgenden Optionen für die Datenbankbereitstellung [1 ... 3]:
1 => EE: Enterprise Edition
2 => RACONE
3 => RAC
3
Ausgewählter Wert ist: RAC
```

Listing 1

```
[dhu-oda1]# oakcli create dbhome
Bitte geben Sie das Benutzerpasswort für den Root ein:
Bitte geben Sie das Benutzerpasswort für den Root nochmals ein:
...
...
```

Listing 2

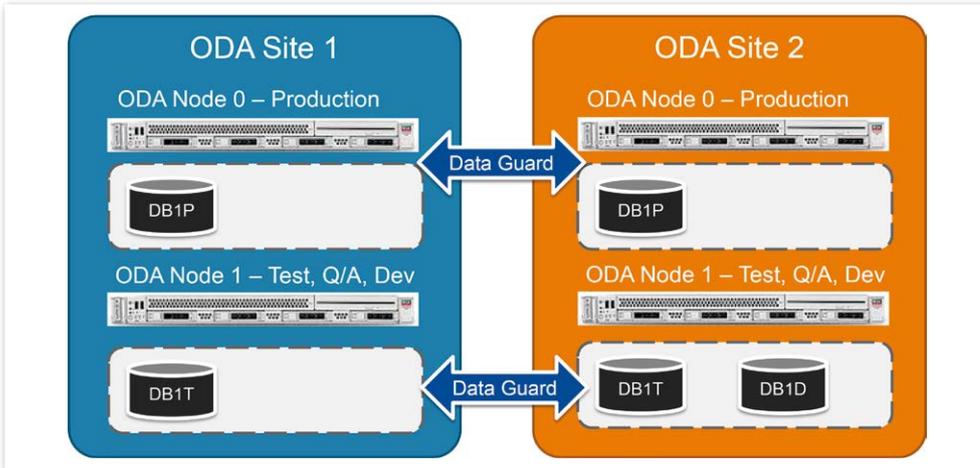


Abbildung 2: ODA-Data-Guard-Implementierung

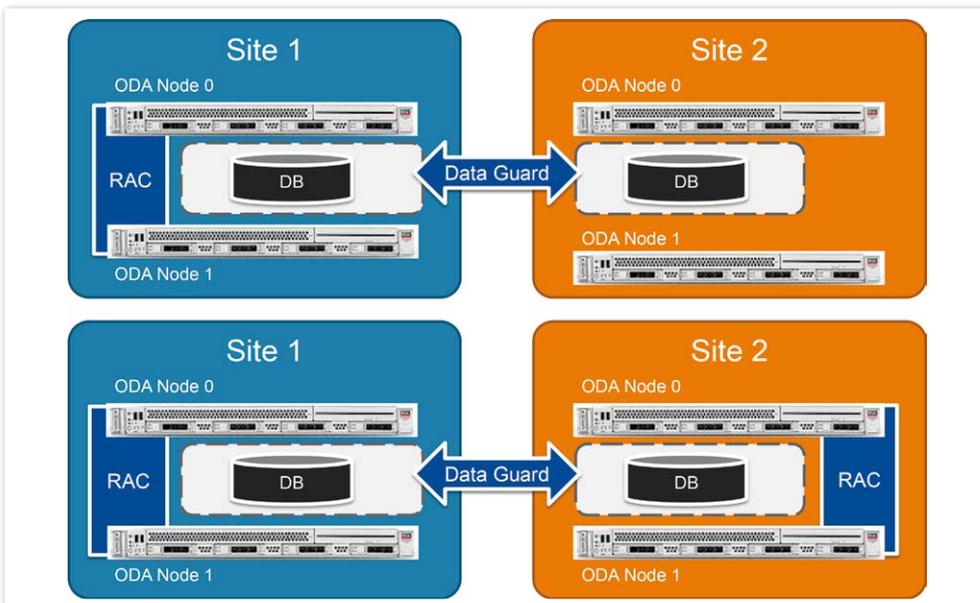


Abbildung 3: Maximal-Available- und Maximum-Availability-Architektur

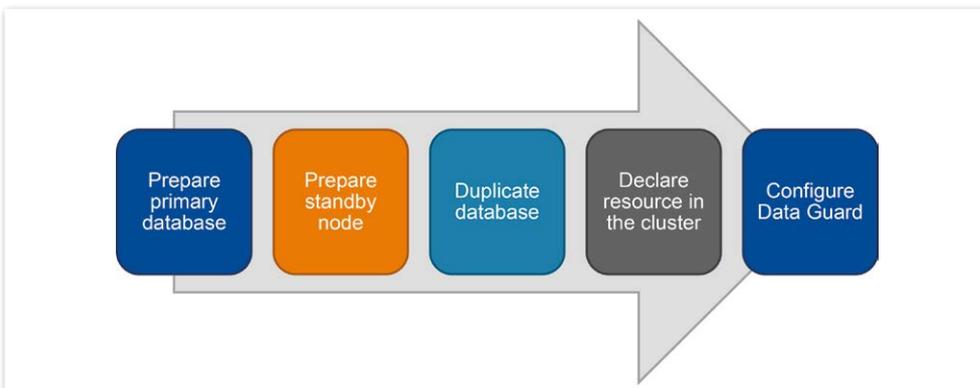


Abbildung 4: Prozess-Überblick

```
[root@dbi-oda2 snaps]# acfsutil snap create -w DBITEST /u02/app/oracle/oradata/datastore/

acfsutil snap create: Snapshot-Vorgang ist abgeschlossen.
```

Listing 3

Guard-Einrichtung in OAKCLI (noch?) nicht integriert. Das bedeutet, dass der DBA dies noch manuell erledigen muss.

Es ist zu beachten, dass Data Guard (oder Data Guard alleine) nicht alle Fälle abdeckt. Wie immer sind einige Analysen erforderlich, um die optimal geeignete Lösung auszuwählen. Aber aus Erfahrung des Autors decken die verschiedenen Möglichkeiten, um Data Guard und ODA zu implementieren, etwa 75 bis 90 Prozent der Fälle ab.

### ODA- und Data-Guard-Implementierung

Das Einrichten von Data Guard zwischen zwei ODAs trägt zu einer Antwort für eine Verfügbarkeitslösung bei, kann jedoch auch einen anderen Punkt lösen, der nicht unterschätzt werden sollte, nämlich die Verfügbarkeit einer Prüf-Umgebung. Eine brandneue ODA in der Produktion ist sehr gut, aber man möchte vielleicht Patches oder neue Versionen von Oracle oder der Anwendungen testen. Die ODA-Architektur ermöglicht eine Implementierung, die beide Fragen durch die Bereitstellung von Knoten für die Produktion beantwortet und Data Guard zwischen den ODAs einrichtet (siehe Abbildung 2).

Grundsätzlich sind mehrere Implementierungslösungen, Test- oder Entwicklungsdatenbanken möglich. In der Tat lässt sich dank der In-Memory-Option, die in der ODA vollständig integriert ist, auch die zweite ODA für eine BI-Umgebung nutzen. Man könnte auch Lösungen wie beispielsweise Fast Maximal Available (AMA) oder Maximum Availability Architecture (MAA) wählen (siehe Abbildung 3).

Die Einrichtung einer Data-Guard-Konfiguration kann grundsätzlich durch ein Verfahren zusammengefasst werden (siehe Abbildung 4). Die Vorbereitung des Primärdatenbank-Schritts verläuft wie bei jeder anderen Umgebung. Man sollte alle Parameter wie „FORCE LOGGING“ oder „DG\_BROKER\_CONFIG\_FILE1/2“ konfigurieren.

Auf der Standby-Seite gibt es, ausgehend von einer brandneuen ODA, ein bisschen mehr zu erledigen:

- Erstellen eines „ORACLE\_HOME“  
Falls nicht der Standard verwendet wird
- Erstellen der Ordner-Struktur  
Seit Version 12 sind Datenbanken auf ACFS-Volumes gespeichert. Diese müssen für die neue Datenbank eingerichtet sein.

```
[root@dbi-oda2 ~]# oakcli create dbstorage -db DBITEST
INFO: 2015-06-08 00:00:31: Beachten Sie die Protokolldatei /opt/oracle/oak/log/
...
...
INFO: 2015-06-08 00:02:36: Speicherstruktur für die Datenbank ,DBITEST' erfolgreich eingerichtet
INFO: 2015-06-08 00:02:36: Legen Sie für die Datenbank DBITEST die folgende Verzeichnisstruktur fest
INFO: 2015-06-08 00:02:36: DATA: /u02/app/oracle/oradata/datastore/.ACFS/snaps/DBITEST
INFO: 2015-06-08 00:02:36: REDO: /u01/app/oracle/oradata/datastore/DBITEST
INFO: 2015-06-08 00:02:36: RECO: /u01/app/oracle/fast_recovery_area/datastore/DBITEST

SUCCESS: 2015-06-08 00:02:36: Speicher für die Datenbank erfolgreich eingerichtet: DBITEST
```

Listing 4

```
(SID_DESC =
  (GLOBAL_DBNAME = DBITEST_SITE2.it.dbi-services.com )
  (ORACLE_HOME = /u01/app/oracle/product/12.1.0.3/dbhome_1 )
  (SID_NAME = DBITEST)
)
```

Listing 5

```
RMAN>duplicate target database for standby from active database dore-
cover nofilenamecheck;
...
...
archived log file name=/u01/app/oracle/fast_recovery_area/datastore/
DBITEST/DBITEST_SITE2/archivelog/2015_04_28/o1_mf_1_2081_bmynm39y_
.arc
thread=1 sequence=2081
media recovery complete, elapsed time: 00:00:03
Finished recover at 28-APR-2015 11:27:07
Finished Duplicate Db at 28-APR-2015 11:27:15
```

Listing 6

```
oracle@dbi-oda2:$ srvctl add database -db DBITEST_SITE2
-oraclehome $ORACLE_HOME
-dbtype SINGLE
-role PHYSICAL_STANDBY
-spfile /u02/app/oracle/oradata/datastore/.ACFS/snaps/DBITEST/DBITEST_
SITE2/spfileDBITEST.ora
-pwfile /u01/app/oracle/product/12.1.0.2/dbhome_1/dbs/orapwDBITEST -db-
name DBITEST
-startoption mount
-stopoption immediate
-instance DBITEST
-node dbi-oda2
-acfspath "/u01/app/oracle/oradata/datastore,/u02/app/oracle/oradata/
datastore,/u01/app/oracle/fast_recovery_area/datastore"
```

Listing 7

- **Vorbereiten von „SPFILE“ für eine Standby-Datenbank**  
Eingeben eines temporären statischen Eintrags in „LISTENER“

Sobald das Duplikat erstellt wurde und die Standby-Datenbank funktionsfähig ist, be-

stehen die letzten Schritte darin, die neue Datenbank im Cluster der Standby-ODA zu erklären und die Data-Guard-Konfiguration zu erstellen.

Standardmäßig verfügt jede neu installierte ODA bereits über ein „ORACLE HOME“ des neuesten verfügbaren Re-

lease (derzeit 12.1.0.2.3). Wird ein separates „ORACLE HOME“ für die Standby-Datenbank benötigt, kann man es mit „OAKCLI“ installieren (siehe Listing 2).

Beim Wechsel der ODA von Version 2.10 auf 12.1.2.0.0 kommt es zu einer umfassenden Änderung der Architektur. Der Datenbank-Speicher wechselt von „Raw ASM“ zu „ACFS“. Dies hat Einfluss auf die Art und Weise, wie die Speicherung für die Standby-Datenbank vorbereitet werden muss. Für Nicht-CDB-Datenbanken gibt es im Grunde drei Standard-ACFS-Volumes:

- **Redo- und Steuerdatei-Volumes**  
Gemountet unter „/u01“
- **Daten-Volumes**  
Gemountet unter „/u02“
- **Wiederherstellungs-Volumes**  
Gemountet unter „/u01“

Bei jedem dieser Volumes ist eine Struktur für jede Datenbank vorgesehen. Zudem ist das Daten-Volume Snapshot-basiert, was bedeutet, dass alle Datendateien in einem ACFS-Snapshot gespeichert sind. Bis Version 12.1.2.1.0 musste man diese Struktur manuell mit „mkdir“, „chown“, „chmod“ und „acfsutil“ erstellen. Die Redo- und Reco-Struktur werden einfach mit „mkdir“ angelegt und die Zugriffsrechte wie für andere Datenbanken. Für das Daten-Volumen ist ein Befehl erforderlich, um den neuen Snapshot zu erstellen (siehe Listing 3). Seit der Version 12.1.2.2.0 ist der gesamte Prozess in „OAKCLI“ integriert und funktioniert jetzt ziemlich einfach (siehe Listing 4).

Wie bei jeder Data-Guard-Einrichtung muss die Standby-Datenbank in „NO-MOUNT“ gestartet werden, bevor das Duplikat laufen kann. Dafür ist ein „SPFILE“ erforderlich. Die einfachste Möglichkeit ist die Verwendung einer der primären Datenbanken sowie die folgende Anpassung:

- *db\_unique\_name*  
Wird nicht auf der Primär-Datenbank festgelegt, die mit „OAKCLI“ erstellt wurde.
- *control\_files*  
Der Parameter, der Namen und Pfad der Steuerdatei(en) definiert, muss aus „SPFILE“ gelöscht werden. Da alle Oracle Managed Files (OMF) auf Basis der ODA sind, lässt sich der Name der Steuerdatei(en) nicht vorab definieren. Durch Löschen des/der Parameter(s) wird/werden er/sie von RMAN während des Duplizierens mithilfe des OMF-Mechanismus neu erstellt.
- *db\_create\_file\_dest, db\_create\_log\_file\_dest, db\_create\_log\_file\_dest2*  
Diese Parameter definieren, wo Steuerdatei(en), Redo-Logs und Datendateien erstellt/gespeichert werden. Die ersten beiden sollten bereits ordnungsgemäß konfiguriert sein. Es könnte sich jedoch lohnen, „f db\_create\_log\_file\_dest2“ festzulegen, um einen Mirror von Redo-Logs und Steuerdateien zu erhalten. Um Leistungsverluste zu vermeiden, benutzt man einen Pfad auf den SSD-Festplatten.

Die Erstellung der Standby-Datenbank erfolgt üblicherweise mit einem Duplikat der aktiven Datenbank. Das bedeutet, dass sowohl auf die Ziel- als auch auf die Hilfs-Datenbank mit „TNS“ zugegriffen werden muss. Da sich die Standby-Datenbank an diesem Punkt nur im „NOMOUNT“-Zustand befindet, ist ein statischer Eintrag im Listener erforderlich, um darauf zuzugreifen.

Der Zuhörer wird vom Netz-Benutzer verwaltet und die Datei „listener.ora“ befindet sich unter „\$GRID\_HOME/network/admin“. Im Abschnitt „SID\_LIST\_LISTENER“ muss ein Eintragstyp hinzugefügt werden (siehe Listing 5). Dieser kann entfernt werden, sobald die Duplizierung abgeschlossen ist. Es ist zu beachten, dass Data Guard in 12c den Namen des DGB-Dienstes verwendet. Das bedeutet, dass der traditionelle „DGMRGL-Eintrag“ nicht mehr erforderlich ist. An diesem Punkt kann man seinen bevorzugten RMAN-Befehl verwenden (siehe Listing 6).

Sobald die Standby-Datenbank erfolgreich erstellt ist, muss sie im Cluster deklariert werden, um die Vorteile von Oracle Restart nutzen zu können. Dieser Vorgang erfolgt als Oracle-Benutzer durch „SRVCTL ADD DATABASE“ (siehe Listing 7) mit folgenden Optionen:

- *db\_unique\_name*
- *ORACLE\_HOME*
- Datenbank-Typ
- Datenbank-Rolle
- *SPFILE* -Standort
- Speicherort der Passwortdatei
- *db\_name*
- Startmodus, „OPEN“ ist Active Guard, anderweitig lizenziert „MOUNT“
- Stopp-Modus
- *instance\_name*
- Running-Knoten
- ACFS-Volumes

Das Erstellen der Data-Guard-Konfiguration ist ähnlich wie bei jeder anderen Data-Guard-Konfiguration und erfolgt mithilfe von „DGMRGL“. Man wählt den Übermittlungsmodus der Logs und den für die Bedürfnisse und die Erstellung am besten geeigneten Schutzmodus (siehe Listing 8).

Der Punkt ist erreicht, an dem Data Guard voll funktionsfähig ist und eine Umstellung durchgeführt werden kann. Listing 9 zeigt den Status, den die Datenbanken auf beiden ODAs jetzt haben. Die Umstellung verläuft geradlinig (siehe Listing 10). Bei der erneuten Überprüfung der Datenbanken sind diese korrekt eingestellt (siehe Listing 11). Ab jetzt ist die frühere Primär-Datenbank definiert als „PHYSICAL\_STANDBY“ und wird im „MOUNT-Modus“ gestartet, während die neue Primär-Datenbank als „PRIMARY“ deklariert ist und im OPEN-Modus gestartet wird.

### Tipps und Tricks

Auf einer ODA, die noch mit Version 2.10 läuft, wird die Duplizierung für den Standby mit einem Fehler fehlschlagen (siehe Listing 12). Der Blick in die „alert.log“-Datei der Datenbank offenbart ein Zugriffsproblem (siehe Listing 13). Dieses Problem ist auf einen Zugriffsfehler der Oracle-Binärdateien zurückzuführen, der im MOS-Hinweis 1.084.186.1 beschrieben wird. Die Lösung ist die Verwendung von „setasmgidwrap“ für „ORACLE HOME“ (siehe Listing 14).

In einer AMA-Architektur ist die Primär-Datenbank eine RAC, während die Standby-Datenbank eine einzelne Instanz ist. Das bedeutet, dass die Primär-Datenbank über zwei Redo-Logs-Threads und die Standby-Datenbank nur über einen einzigen verfügt. Damit eine solche Data-Guard-Einrichtung jedoch ordnungsgemäß funktioniert, sind für die Standby-Datenbank zwei Standby-Redo-Logs-Threads erforderlich.

### Umstellung auf Oracle 11g

Bei einem 11g-Data-Guard unterscheidet sich das Verhalten der Datenbank-Ressourcen während einer Umstellung etwas. Listing 15 zeigt den Zustand vor der Umstellung und Listing 16 danach. Wie wir sehen, sind die Ressourcen nur teilweise angepasst, was folgende Konsequenzen hat:

- Die neue Primär-Datenbank wird bei einem ODA-Neustart nur in „MOUNT“ gestartet und der 11g-Broker wird sie nicht öffnen
- Die Standby-Datenbank wird im „OPEN“-Modus gestartet, womit Active Guard aktiviert ist

Nach Überprüfung im Oracle-Support und Öffnen eines Service Request kam heraus, dass es nur zwei Lösungen/Workarounds gibt:

- Lizenzierung von Active Guard und Konfigurieren des Einrichtungsmodus „OPEN“ für beide Datenbanken
- Manuelles Ändern der Cluster-Ressourcen-Parameter nach jeder Umstellung oder Failover

### Fazit

Die Kombination von Data Guard und ODA ist definitiv eine interessante Lösung, um mehrere Anliegen wie beispielsweise Datenverfügbarkeit, Standort-Notfälle oder Test-Umgebungen zu lösen. Falls es seit der ersten Version zur Verfügung steht, sind es die erheblichen Verbesserungen mit Version 12c wert, alle Vorteile solcher Architekturen zu nutzen.

Alle weiteren Listings finden Sie online unter:

[www.doag.org/go/doagsoug/201506/listing](http://www.doag.org/go/doagsoug/201506/listing)



David Hueber  
david.hueber@dbi-services.com

# High Performance Spatial Computing mit Oracle 12c

Eva-Maria Kramer, Disy Informationssysteme GmbH

In der heutigen Zeit sind Datenbanken schon lange keine reinen Daten-Container mehr. Sie können weit mehr als nur schnelle Views und Daten für Drittsysteme bereitstellen. Immer schnellere Systeme und immer größerer Funktionsumfang ermöglichen es, komplexe Projekte komplett in der Datenbank zu bearbeiten. Dieser Artikel zeigt Tipps und Tricks beim Spatial Computing und spricht auch das Thema „High Performance“ an.

Anfang 2014 startete das Unternehmen der Autorin ein großes Kundenprojekt, in dem diverse Herausforderungen gemeistert werden mussten. Inhaltlich ging es um die Aufbereitung heterogener Geometrie- und Sachdaten zu einem konsistenten Gesamtdaten-Bestand, der für eine anschließende Weiterverarbeitung in Modellrechnungen benötigt wurde.

Das gesamte Projekt stand unter extremem Zeitdruck, sodass keine klassische Projekt-Durchführung möglich war. Die Daten mussten in sehr kurzer Zeit analysiert und bezüglich ihrer grundsätzlichen Eignung für die Fragestellung qualifiziert werden. Anschließend waren sie attributiv aufzuwerten, mit Blick auf Relevanz zu filtern und räumlich zu verschneiden. Heruntergebrochen in Teilschritte bedeutete dies über 250 Algorithmen. Damit das Projekt in der verfügbaren Zeit möglich war, musste auf getrennte Implementierungs-, Software-QS- und anschließende Berechnungs-Phasen verzichtet werden. Und selbst dann war der Zeitplan nur mit absoluter Top-Performance des Systems und kreativen Lösungen zu stemmen.

## Entscheidung für Oracle 12c und Solaris

Unter den genannten Umständen kam für die Berechnungen nur eine Lösung direkt in der Datenbank infrage. Damit stand eine schwierige Entscheidung an: Entweder auf die bewährte und vielfach erprobte Funktionalität von Oracle 11g setzen

oder es mit 12c im Release 1 versuchen. Letztendlich sprach die Erwartungshaltung an die bessere Rechenzeit dafür, mit 12c ins Rennen zu gehen. Die umfangreichen Optimierungen der gesamten Spatial-Funktionalitäten, und hier besonders die Spatial Vector Acceleration („SPATIAL\_VECTOR\_ACCELERATION = TRUE;“), gaben den entscheidenden Ausschlag.

Bei einem Projekt dieser Größenordnung ist es zu erwarten, dass man auf Fehler stößt, insbesondere wenn neue Software-Releases zum Einsatz kommen. Die in 12c gefundenen Probleme konnten alle durch Hotfixes sehr schnell gelöst oder durch einfache Workarounds umgangen werden.

Ein Beispiel für ein Problem war die Nutzung der Puffer-Funktion „SDO\_GEOM.SDO\_BUFFER“, die eine Puffer-Fläche um existierende Geometrien berechnet. Unter bestimmten Umständen lieferte diese Funktion falsche Ergebnisse. Ein anderes Beispiel war die häufig verwendete Funktion „SDO\_AGGR\_UNION“ zur räumlichen Vereinigung von Geometrien. Wurde diese in Schleifen verwendet, passierte es, dass die Prozedur ohne eine sprechende Fehlermeldung einfach abbrach.

Das zweite zentrale Element zur Einhaltung des Projektplans war das gewählte Server-Setup, auf dem die Datenbank installiert wurde. Für maximale CPU- und I/O-Performance sowie Stabilität setzte man auf Solaris mit ZFS als Dateisystem sowie Solaris Zonen. In der RAM-Disk wa-

ren sowohl die komplette Datenbank installiert als auch I/O-kritische Tablespace abgelegt. Weitere Tablespace mit normaler I/O-Last waren auf einem Storage abgelegt. Durch ZFS-LZ4-Kompression sowie Hyper-Threading standen dem Projekt somit 500 GB RAM und 60 Kerne zur Verfügung, weitere 1.500 GB auf dem Storage. Durch Snapshots, die auf ein zweites Solaris-System repliziert wurden, war keine Downtime für ein Backup des Systems erforderlich.

Dadurch konnte im Falle eines Fehlers sehr schnell und einfach auf den letzten bekannten Snapshot zurückgegangen werden. Der Zeitverlust durch den Rücksprung auf einen bis zu zwei Tage alten Snapshot wurde durch die schnellere Berechnung mehr als aufgewogen. Entsprechende Mechanismen ermöglichten es, die Oracle-Datenbank ohne Verzögerung in Betrieb zu nehmen – zunächst mit Performance-Einschränkungen; sobald alle Daten auf die RAM Drives migriert waren (ca. 4 Stunden nach einem System-Neustart), stand dann wieder die volle I/O-Performance zur Verfügung.

## Dauerhafte Nutzung aller verfügbaren Ressourcen

Um die maximale Rechenleistung zu erreichen, mussten alle verfügbaren CPU-Kerne durchgängig ausgelastet werden. Solange es berechenbare Schritte gab, sollte kein Thread ungenutzt bleiben. Letztendlich blieb man knapp unter diesem Ziel. Von 64 CPU-

Kernen wurden 60 für die Berechnung verwendet. Die restlichen vier blieben dem Betriebssystem und dem Datenbank-System vorbehalten, da es für die Performance von 12c nicht förderlich ist, wenn zu viele Threads aus Hintergrund-Prozessen auf den gleichen Kernen laufen.

Für die Ablaufsteuerung ist mit der Disy Spatial Workbench über die Jahre ein eigenes Framework in Java entwickelt worden. Diese verwaltet vollautomatisch alle verfügbaren Threads und kennt die noch offenen Aufgaben mit entsprechenden Abhängigkeiten. Solange rechenbereite Aufgaben vorliegen, werden diese in logisch sinnvoller Reihenfolge gestartet. Im Falle eines Fehlers wird die entsprechende Aufgabe protokolliert und übersprungen, sodass der Thread direkt freigegeben ist. Damit ist es möglich, vom Fehler unabhängige weitere Berechnungen direkt neu zu starten, ohne dass die Threads blockiert bleiben.

Als Trade-off dieses Ansatzes stand die von Oracle über SQL direkt nutzbare parallele Verarbeitung von Statements nicht zur Verwendung. Damit hätte man die Kontrolle über die Threads aus der Hand gegeben, was in Summe wahrscheinlich Rechenzeit gekostet, auf jeden Fall aber die Zahl der grauen Haare in die Höhe getrieben hätte.

### Beispiel aus der Praxis

Im Rahmen des Projekts zeigte sich wieder eindrucksvoll, dass auch das beste Setup nichts nützt, wenn die einzelnen Abfragen unperformant geschrieben sind oder der Optimizer überfordert ist. Gerade die Verarbeitung von räumlichen Daten bietet hier einige Herausforderungen. Im Folgenden ein Beispiel dafür, mit welchen Strategien man auch bei komplexen geometrischen Aufgaben mit 12c ans Ziel kommen kann.

Das Ergebnis der Modellierung waren sogenannte „Isophonen“, also Flächen (Geometrien) mit gleicher Lautstärke. Isophonen können als Flächen oder als Bänder vorliegen (siehe Abbildung 1). Die Flächen zeigen je Pegelklasse die Bereiche, in denen bestimmte Lautstärken erreicht sind (Mindest-Lautstärke). Die Bänder dagegen zeigen jeweils die betroffene Fläche eines Lautstärke-Intervalls an. Im Projekt mussten beide Arten der Darstellung erzeugt werden, sowohl Flächen als auch Bänder.

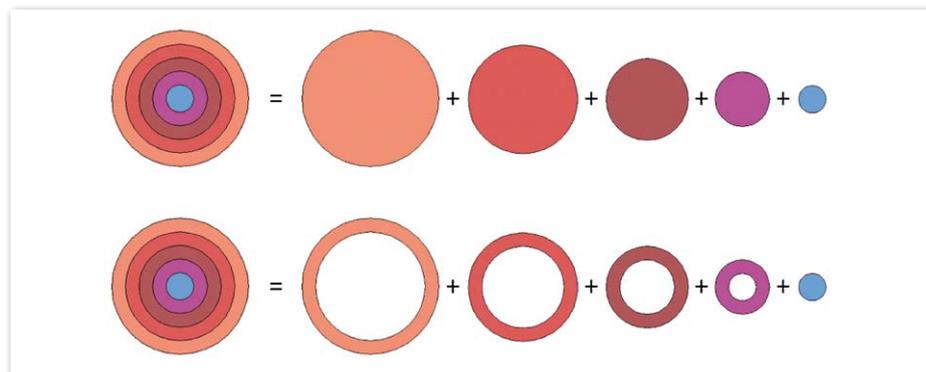


Abbildung 1: Isophonen-Geometrien als Flächen (oben) und Bänder (unten)

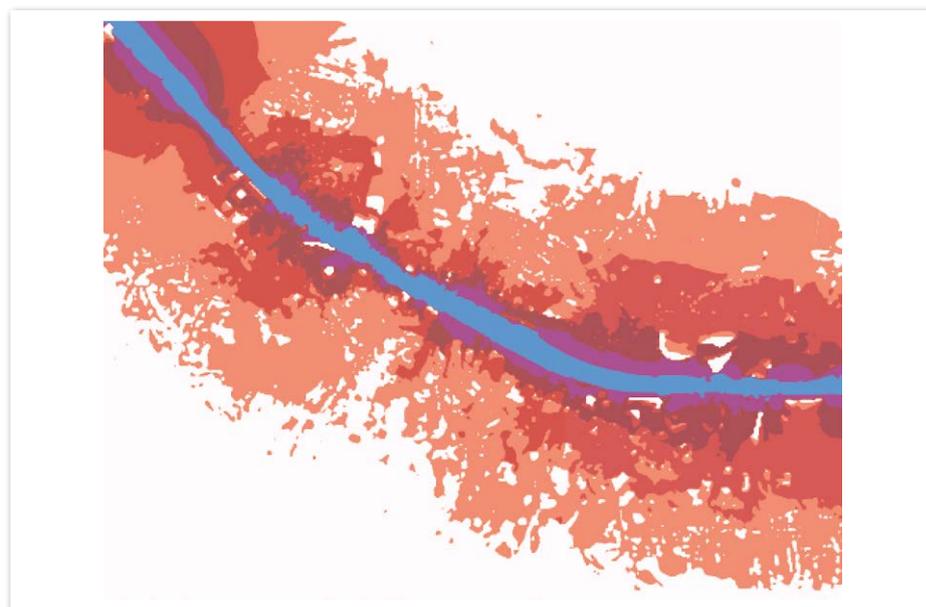


Abbildung 2: Komplexe Isophonengeometrien

Modellierte Isophonen zeichnen sich in der Regel durch extrem viele Stützpunkte aus, da die Geometrien häufig sehr groß und gleichzeitig sehr komplex sind (siehe Abbildung 2). Viele Stützpunkte bedeuten mehr Speicherbedarf, vor allem jedoch längere Laufzeiten bei räumlichen Abfragen beziehungsweise Operationen. Bei Bändern ist die Anzahl der Stützstellen durch die inneren Ringe nochmals deutlich größer als bei den Flächen.

Aus dem Projekt resultierten knapp 400.000 Isophonen-Flächen mit in Summe 95 Millionen Stützpunkten. Hieraus mussten fast 200 Millionen Isophonen-Bänder berechnet werden, die insgesamt über 150 Millionen Stützpunkte aufwiesen. Der erste Ansatz, um aus den Flächen-Geometrien Bänder-Geometrien zu generieren, war ein kurzes Statement (siehe Listing 1). Darin wurden von den Flächen-Geometrien alle im gleichen räumlichen Gebiet liegenden Flä-

chen mit höheren Lärmwerten abgezogen. Sachdaten sowie Identifier blieben erhalten.

Für sehr kleine Datenmengen ist ein solches Statement vertretbar, für größere ist dieser Ansatz nicht praktikabel. Selbst wenn das Statement nach Wochen erfolgreich abgeschlossen worden wäre (und man diese Zeit gehabt hätte), wären drei große Nachteile geblieben. Erstens muss die gesamte Konfiguration der Redo-Logs auf lang laufende Abfragen ausgelegt sein, mit allen Nachteilen, die man hierdurch in Kauf nehmen muss. Zweitens müsste im Falle eines Abbruchs (verlorene Connection, Stromausfall) nach dem Redo das gesamte Statement nochmals von vorne verarbeitet werden. Und drittens hat man mit solch einem Statement keinerlei Anhaltspunkt, wie lange die Verarbeitung noch dauert. Die erste Optimierung bestand darin, eins nach dem anderen zu tun (siehe Listing 2).

Die Berechnung erfolgte nicht mehr in einer einzigen Transaktion in der Datenbank. Vielmehr wurde in einer Schleife über alle Flächen iteriert. Jede berechnete Fläche wurde direkt der Ergebnis-Tabelle hinzugefügt. Damit bot sich die Möglichkeit, dass im Falle eines Abbruchs das Statement jederzeit für alle noch nicht verarbeiteten Geometrien wiederaufgenommen werden konnte. Schon bearbeitete Datensätze blieben erhalten.

Gleichzeitig war jederzeit transparent, wie viele Datensätze bereits verarbeitet wurden, und somit eine zeitliche Hochrechnung möglich. Die technischen Probleme des ersten Ansatzes waren damit adressiert. Für den engen Zeitplan war die Laufzeit des Statements jedoch immer noch nicht hinreichend. Die Lösung: Teile und herrsche (*siehe Listing 3*).

In einem ersten Schritt wurden alle innersten Geometrien, also die lautesten Flächen, direkt in die Ergebnis-Tabelle für Bänder geschrieben. Hier gab es keinen Unterschied von Fläche zu Band. Anschließend verarbeitete man die restlichen Geometrien mit einer doppelten Schleife. Der Optimizer von Oracle ist an dieser Stelle nicht in der Lage, den geometrischen Index für eine Vorselektion für die Funktion „SDO\_RELATE“ zu nutzen. Stattdessen musste eine Vorselektion über das vorberechnete minimal umgebende Rechteck (engl.: MBR) erfolgen, was ein Vielfaches an Rechenzeit einsparte. An dieser Stelle ein Tipp: In 12c gibt es gegenüber 11g eine verbesserte und schnellere Implementierung dieser Funktion: „SDO\_GEOM\_MBR“ statt „SDO\_GEOM.SDO\_MBR“.

Die Aufgabe der äußeren Schleife bestand darin, alle noch zu prozessierenden Flächen zu selektieren und das MBR zu erstellen. Die innere Schleife iterierte dann über alle Objekte, die das MBR der äußeren Schleife berührten. Es wurden also nur die Geometrien rechenintensiv miteinander verschnitten, die sich gemäß der Vorselektion anhand der räumlichen Lage auch potenziell überlagerten. Neben den Vorteilen der ersten Optimierung war mit dieser Anweisung auch das Laufzeit-Problem adressiert.

## Fazit

Mit diesem Setup wurde im letzten Jahr das genannte Projekt sehr erfolgreich abgeschlossen. Durch die Kombination der

```
CREATE TABLE Isophonen_Baender AS
SELECT o.Sachdaten,
       o.Pegel,
       SDO_GEOM.SDO_DIFFERENCE (o.Geom,
                                (SELECT SDO_AGGR_UNION (SDOAGGRTYPE (i.Geom, 0.05))
                                 FROM Isophonen_Flaechen i
                                 WHERE i.Sachdaten = o.Sachdaten AND i.Pegel > o.Pegel),
                                0.05) Geom
FROM Isophonen_Flaechen o;
```

Listing 1

```
DECLARE
BEGIN
  FOR c IN (SELECT Sachdaten, Pegel, Geom FROM Isophonen_Flaechen)
  LOOP
    INSERT INTO Isophonen_Baender (Sachdaten, Pegel, Geom)
    SELECT o.Sachdaten,
           o.Pegel,
           SDO_GEOM.SDO_DIFFERENCE (o.Geom,
                                    (SELECT SDO_AGGR_UNION (SDOAGGRTYPE (i.Geom, 0.05))
                                     FROM Isophonen_Flaechen i
                                     WHERE i.Sachdaten = o.Sachdaten AND i.Pegel > o.Pegel),
                                    0.05) Geom
    FROM Isophonen_Flaechen o
    WHERE c.Sachdaten = o.Sachdaten AND c.Pegel = o.Pegel;
  COMMIT WRITE NOWAIT;
  END LOOP;
END;
/
```

Listing 2

```
DECLARE
  v_band_geom MDSYS.SDO_GEOMETRY;
BEGIN
  INSERT INTO Isophonen_Baender (Sachdaten, Pegel, Geom)
  SELECT Sachdaten, Pegel, Geom FROM Isophonen_Flaechen WHERE Pegel =
75;
  COMMIT WRITE NOWAIT;

  FOR o IN (SELECT Sachdaten, Pegel, Geom, SDO_GEOM_MBR(Geom) Mbr FROM Iso-
phonen_Flaechen WHERE Pegel != 75)
  LOOP
    v_band_geom := o.Geom;
    FOR i IN (SELECT Geom FROM Isophonen_Flaechen
              WHERE Sachdaten = o.Sachdaten AND Pegel > o.Pegel
              AND SDO_RELATE(Geom, o.Mbr, ,mask=anyinteract`) = 'TRUE'
              ORDER BY Pegel desc)
    LOOP
      v_band_geom := SDO_GEOM.SDO_DIFFERENCE(v_band_geom, i.Geom, 0.05);
    END LOOP;

    INSERT INTO Isophonen_Baender (Sachdaten, Pegel, Geom)
    VALUES (o.Sachdaten, o.Pegel, v_band_geom);
    COMMIT WRITE NOWAIT;
  END LOOP;
END;
/
```

Listing 3

stabilen Datenbank-Version 12c und der enormen Verbesserungen ihrer Performance für räumliche Abfragen gegenüber der Version 11g mit dem gewähltem Hardware-Setup war man in der Lage, sämtliche Ergebnisse in Quality, in Budget und vor allem auch in Time abzuliefern.

Die von Oracle bereitgestellten räumlichen Funktionen (Locator sowie Spatial-Option) sind ausgereift und vollumfänglich, auch die mit 12c teilweise komplett neu geschriebenen Funktionen sind inzwischen stabil. Bei einigen Funktionen muss der Anwender sich jedoch über die Besonderheiten oder den offiziellen Funktionsumfang der Funktionen im Klaren sein. Der Umgang der gleichen Funktion mit 2D- und 3D-Daten ist zwischen den Versionen zum Teil unterschiedlich.

Bei der Implementierung von Abfragen oder PL/SQL-Routinen lohnt sich immer wie-

der einen Blick in die Dokumentation oder das Change-Log. Mit 12c wurde zum Beispiel das bereits erwähnte „SDO\_GEOM.SDO\_MBR“ durch „SDO\_GEOM\_MBR“ ergänzt, wobei Letzteres ein deutlich schnelleres Resultat liefert. Die neue Funktion „POINTIN\_POLYGON“ ist in der Lage, extrem schnell Punkte mit Polygonen zu korrelieren. Absolutes Highlight für die Berechnung war aber das „SPATIAL\_VECTOR\_ACCELERATION“-Flag. Die Geschwindigkeitsgewinne durch diese Funktion sind wirklich enorm.

Wenn der Projektzeitplan ein wenig lockerer gewesen wäre, hätte die Autorin große Freude daran gehabt, die diversen Berechnungen vergleichsweise auch auf einer Version 11g auszuführen, um den Unterschied zwischen 11g und 12c im Gesamten zu sehen. Für ihren Teil möchte sie 12c für Spatial-Processing-Projekte nicht mehr missen.

Die finalen Ergebnisse des Kundenprojekts und vor allem die Isophonen können online auf den Seiten des Eisenbahn-Bundesamtes eingesehen werden (*siehe* „<http://tinyurl.com/pvyt59d>“).



Eva-Maria Kramer  
eva.kramer@disy.net

## Wir begrüßen unsere neuen Mitglieder

### Persönliche Mitglieder

Ingo Jannick	Thilo Jäger
Marc Steinhaus	Uwe Hofmann
Kai Weber	Florian Giesecke-Uellner
Stefan Kopp	Sven Ruppert
Christine Wege	Ronald Stöcken
Joern Kleinbub	Ulrich Lickert
Christoph Pfrommer	Ivan Korac
Georghe Breazu	Thomas Rein
Jörg Vargel	

### Firmenmitglieder DOAG

Johannes Bayer-Albert, byteletics oHG  
Gerhard Lutz, Stadtwerke Augsburg Hold. GmbH  
Thomas Schlenker, Sage bäurer GmbH  
Oliver Pliquet, SupplyOn AG  
Roland Junge, Toll Collect GmbH

### Neumitglieder SOUG

Pierre-Jean Giraud, Giraud Adequations  
Daniel Meienberg, Diso AG  
Ivan Korac, Altran  
Mariusz Zuk, upc cablecom

# Heute: Undo

Gerd Volberg, OPITZ CONSULTING GmbH

Daten aus der Datenbank zu selektieren, zu ändern und dann zu speichern, ist in Oracle Forms das Einfachste der Welt. Aber was ist, wenn der Anwender vor dem Speichern einen einzelnen Datensatz auf den ursprünglichen Wert zurücksetzen möchte (*siehe Abbildung 1*)?

ID	First Name	Last Name	Phone Number	Hire Date
198	Donald	OConnell	650.507.9833	21.06.1999
199	Douglas	Grant	650.507.9844	13.01.2000
200	Jennifer	Whalen	515.123.4444	17.09.1987
201	Michael	Hartstein	515.123.5555	17.02.1996
202	Pat	Fay	603.123.6666	17.08.1997
203	Susan	Mavris	515.123.7777	07.06.1994

Abbildung 1: Initiale Daten

ID	First Name	Last Name	Phone Number	Hire Date
198	Donald	Sutherland	650.507.9833	21.06.1999
199	Douglas	Grant-Fay	650.507.9844	13.01.2000
200	Jennifer	Whalen	515.123.4444	17.09.1987
201	Michael	Harts	515.123.5555	17.02.1996
202	Pat	Fay	603.123.6666	17.08.1997
203	Susan	Mavris	515.123.7777	07.06.1994

Abbildung 2: Nach dem Update, vor dem Commit

```

PROCEDURE Undo IS
  V_Block  VARCHAR2 (30) := :SYSTEM.CURSOR_BLOCK;
  V_Field  VARCHAR2 (61);
  V_Item   VARCHAR2 (61);
BEGIN
  Validate (Item_Scope);
  IF :SYSTEM.RECORD_STATUS = 'CHANGED' THEN
    V_Field := Get_Block_Property (V_Block, FIRST_
ITEM);
    V_Item  := V_Block || '.' || V_Field;

    WHILE V_Field IS NOT NULL
    LOOP
      IF Get_Item_Property (V_Item, ITEM_TYPE) IN
        ('DISPLAY ITEM', 'CHECKBOX', 'LIST',
        'RADIO GROUP', 'TEXT ITEM')
        AND Get_Item_Property (V_Item, BASE_TABLE) =
,TRUE' THEN
        COPY (Get_Item_Property (V_Item, DATA-
BASE_VALUE), V_Item);
        END IF;
        V_Field := Get_Item_Property (V_Item, Nex-
tItem);
        V_Item  := V_Block || '.' || V_Field;
        END LOOP;
      END IF;
      Set_Record_Property (:SYSTEM.CURSOR_RECORD, :SYSTEM.
CURSOR_BLOCK,
        STATUS, QUERY_STATUS);
    END;
  
```

Listing 1

ID	First Name	Last Name	Phone Number	Hire Date
198	Donald	Sutherland	650.507.9833	21.06.1999
199	Douglas	Grant-Fay	650.507.9844	13.01.2000
200	Jennifer	Whalen	515.123.4444	17.09.1987
201	Michael	Hartstein	515.123.5555	17.02.1996
202	Pat	Fay	603.123.6666	17.08.1997
203	Susan	Mavris	515.123.7777	07.06.1994

Abbildung 3: Nach dem Undo

Eine neue Abfrage gegen die Daten- bank abzusetzen, kommt nicht infrage, dann wären alle Änderungen weg, die vor dem Commit gemacht wurden.

Es muss also ein echtes Undo ent- wickelt werden, der genau einen Da- tensatz auf den ursprünglichen Zu- stand zurücksetzt (*siehe Abbildung 2*). Die Prozedur in *Listing 1* ermöglicht dies. *Abbildung 3* zeigt das Ergebnis.



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



08.01.2016

**Webinar: Datenbank-Security**

Johannes Ahrends, Christian Trieb  
sig-database@doag.org

19.01.2016

**Regionaltreffen NRW (Primavera)**

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

21.-22.01.2016

**DOAG Noon2Noon RAC & Dataguard**

Robert Marz, Björn Rost  
robert.marz@its-people.de

25.01.2016

**Regionaltreffen Jena/Thüringen**

Jörg Hildebrandt  
regio-thueringen@doag.org

28.01.2016

**Regionaltreffen Stuttgart**

Jens-Uwe Petersen, Anja Zahn  
regio-stuttgart@doag.org



03.02.2016

**Regionaltreffen München/Südbayern**

Franz Hüll, Andreas Ströbel  
regio-muenchen@doag.org

08.02.2016

**Regionaltreffen Berlin/Brandenburg**

Michel Keemers  
regio-bb@doag.org

09.02.2016

**Regionaltreffen Hamburg/Nord**

Jan-Peter Timmermann  
regio-nord@doag.org

Weitere Termine und Informationen unter [www.doag.org/termine/calendar.php](http://www.doag.org/termine/calendar.php) sowie unter [www.soug.ch](http://www.soug.ch)

12.02.2016

**Webinar: Oracle 12c Standard Edition Two Single Tenant Database**

Johannes Ahrends, Christian Trieb  
sig-database@doag.org

23.-24.02.2016

**DevCamp 2016**

office@doag.org



08.03.2016

**JavaLand 2016**

DOAG Geschäftsstelle  
office@doag.org

11.03.2016

**Webinar: Datenbank-Security**

Johannes Ahrends, Christian Trieb  
sig-database@doag.org



10.-14.04.2016

**COLLABORATE 16**

collaborate.oaug.org

12.04.2016

**Regionaltreffen Hamburg/Nord**

Jan-Peter Timmermann  
regio-nord@doag.org

13.04.2016

**DOAG 2016 Exaday**

DOAG Geschäftsstelle  
office@doag.org

**Impressum**

**Herausgeber:**

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

SOUG Swiss Oracle User Group  
Im Gundelinger Feld/Bau 5  
Dornacherstrasse 192  
CH-4053 Basel  
Tel.: +41 (0)61 367 93 30  
[www.soug.ch](http://www.soug.ch)

**Verlag:**

DOAG Dienstleistungen GmbH  
Fried Saacke, Geschäftsführer  
[info@doag-dienstleistungen.de](mailto:info@doag-dienstleistungen.de)

**Chefredakteur (ViSdP):**

Wolfgang Taschner, [redaktion@doag.org](mailto:redaktion@doag.org)

**Redaktion:**

Fried Saacke, Julia Bartzik, Marina Fischer,  
Mylène Diacquenod, Marius Fiedler,  
Dr. Dietmar Neugebauer, Jan Peterskovsky

**Titel, Gestaltung und Satz:**

Alexander Kermas, DOAG Dienstleistungen GmbH

**Titel:** © Mimi Potter / fotolia.com

**Foto S. 41:** © Oracle / oracle.com

**Foto S. 45:** © stillkost / fotolia.com

**Foto S. 71:** © antonprado / 123rf.com

**Anzeigen:**

Simone Fischer, [anzeigen@doag.org](mailto:anzeigen@doag.org)  
DOAG Dienstleistungen GmbH  
Mediadaten und Preise finden Sie  
unter: [www.doag.org/go/mediadaten](http://www.doag.org/go/mediadaten)

**Druck:**

Druckerei Rindt GmbH & Co. KG  
[www.rindt-druck.de](http://www.rindt-druck.de)

**Inserentenverzeichnis**

Apps Associates LLC <a href="http://www.appsassociates.com">www.appsassociates.com</a>	S. 59
DBConcepts <a href="http://www.dbconcepts.at">www.dbconcepts.at</a>	S. 29
dbi services ag <a href="http://www.dbi-services.com">www.dbi-services.com</a>	S. 47
DOAG e.V. <a href="http://www.doag.org">www.doag.org</a>	U 2, U 3
Libelle AG <a href="http://www.libelle.com">www.libelle.com</a>	S. 25
MuniQsoft GmbH <a href="http://www.muniqsoft.de">www.muniqsoft.de</a>	S. 3
Softbase A/S <a href="http://www.softbase.com">www.softbase.com</a>	S. 35
Trivadis GmbH <a href="http://www.trivadis.com">www.trivadis.com</a>	U 4

# DevCamp 2016

23. - 24. Februar 2016 | Bonn

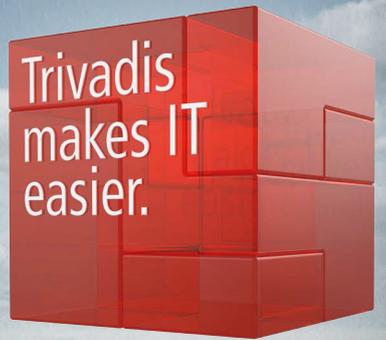


- ADF Fitness Center
- IoT
- Mobile

[devcamp.doag.org](http://devcamp.doag.org)

# Grenzenlose Möglichkeiten

Mit Big Data & Data Science.



Viele denken bei Big Data zunächst nur an Grossprojekte mit einem hohen finanziellen Aufwand. Trivadis hat bereits in zahlreichen kleinen Projekten bewiesen, dass Big Data auch bei kleinem Budget einen grossen Nutzen bringt. Trivadis kombiniert langjährige Erfahrung und Best Practices in Data Management, Analytics und Processing mit aktuellsten Big-Data-Technologien und hochentwickelten analytischen Verfahren. Meistern Sie mit Trivadis neue und aussergewöhnliche Anforderungen an Datenmenge, Komplexität und Performance und ziehen Sie so wirtschaftlichen Nutzen aus allen verfügbaren Daten. Sprechen Sie mit uns über Big Data.

[www.trivadis.com](http://www.trivadis.com) | [info@trivadis.com](mailto:info@trivadis.com)

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

**trivadis**  
makes IT easier. ■ ■ ■