

Red Stack

Magazin

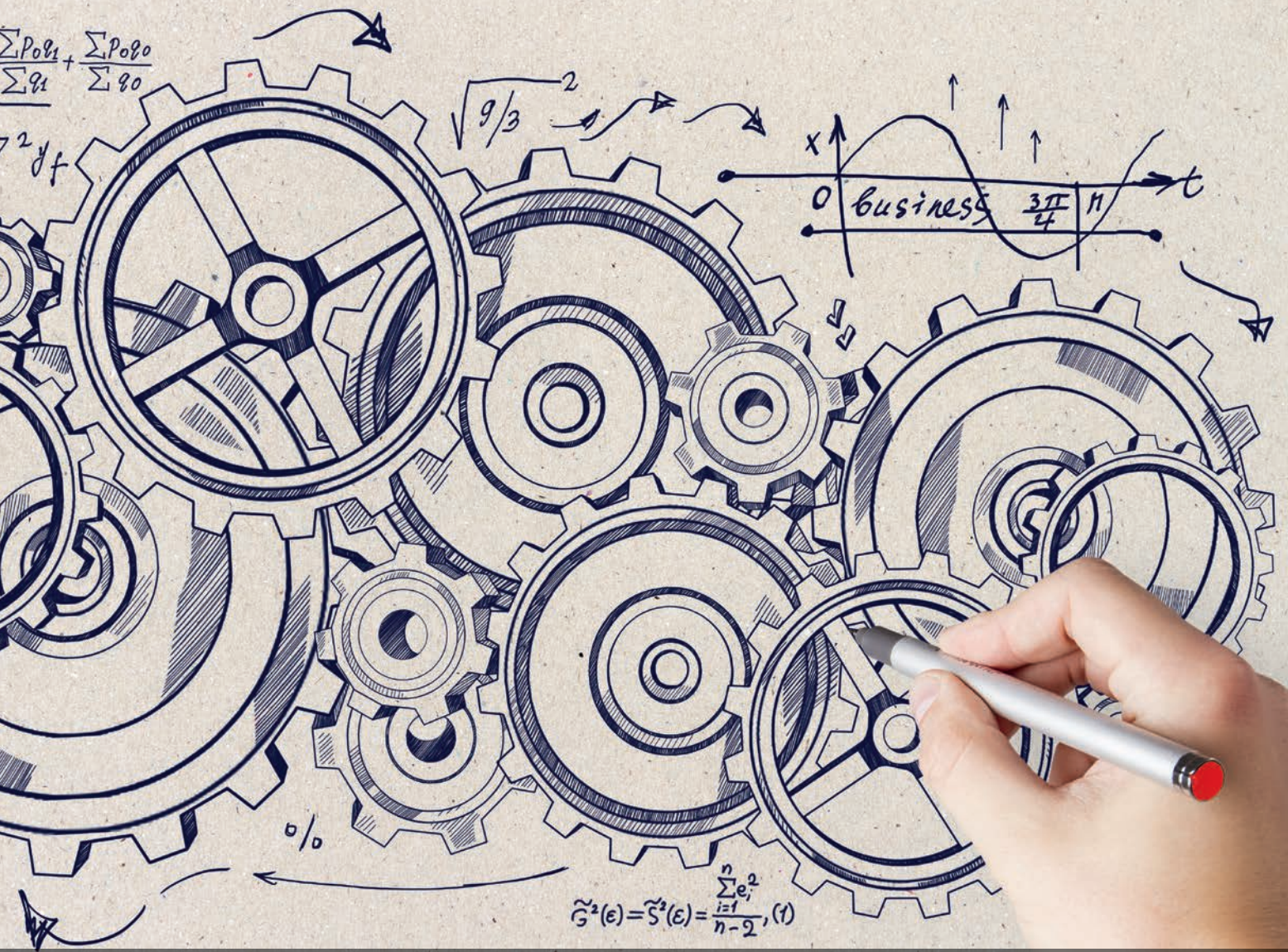
DOAG

SOUG
swiss oracle
user group

AOUG
AUSTRIAN ORACLE USER GROUP

Jetzt inklusive BUSINESS NEWS

Engineered Systems



Aus der Praxis

Virtualized Exadata – die vier ersten produktiven Jahre

Im Interview

Juan Loaiza, Executive Vice President, Mission-Critical Database Technologies, Oracle



Business News

Agile Unternehmenskultur und KI-Commerce

CloudLand

Pre-Event

19. März 2020
in Brühl bei Köln





Jan-Peter Timmermann
Vorstand Infrastruktur & Middleware

Liebe Mitglieder, liebe Leserinnen und Leser,

wie schreibt man hier eine Einleitung für dieses Heft... Als ich damit angefangen habe, stand ich persönlich vor der Entscheidung „Ich brauche ein neues Auto“. Es kam mir gar nicht in den Sinn, mein Auto durch verschiedene Hersteller zusammenzustellen. Ich habe mich für eine Konfiguration entschieden, die aufeinander abgestimmt ist; alle Komponenten, die verbaut worden sind, können sehr gut miteinander. Warum habe ich das gemacht? Warum mache ich das nicht im Server-Umfeld?

Diese Ausgabe des Red Stack Magazin beschäftigt sich ausgiebig mit dem Thema Engineered Systems. Vielleicht finden auch Sie die eine oder andere Antwort, die bisher noch nicht beantwortet werden konnte.

Ein Hinweis zur aktuellen Ausgabe. Ab sofort ist das bisher als separates vierteljährlich erscheinende Magazin Business News in das Red Stack Magazin integriert. Inhaltlich verändert sich nichts, nur der Seitenumfang pro Ausgabe wurde aufgrund der statt vier- nun sechsmaligen Erscheinungsweise reduziert. Die Business News widmet sich diesmal dem Thema „Agile Unternehmenskultur“ und betrachtet unter anderem auch die Auswirkungen der Digitalisierung auf den Einzel- und Großhandel.

Ich wünsche Ihnen viele wertvolle Erkenntnisse beim Lesen dieser Ausgabe und wünsche Ihnen noch einen guten Start in das neue Jahr 2020.

Ihr Jan-Peter Timmermann



Training

Training

MUNIQSOFT

TRAINING

ORACLE®
Silver Partner

20 Jahre Oracle-Datenbankschulungen von Experten
Effizient und kundenorientiert

Offene Kurse demnächst

DBA I	23.03.-27.03.2020	1.990,- € netto
PL/SQL II	23.03.-27.03.2020	1.990,- € netto
DB Security I und II	30.03.-03.04.2020	1.990,- € netto
Neuerungen 12c - 19c	30.03.-03.04.2020	1.990,- € netto
APEX I	20.04.-24.04.2020	1.990,- € netto
B&R mit RMAN	20.04.-24.04.2020	1.990,- € netto
Postgres SQL	27.04.-29.04.2020	1.590,- € netto

☎ 089 679090-40

Website: www.munisoft-training.de

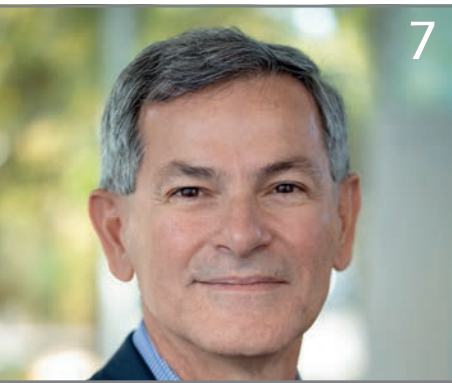
Tipps: www.munisoft-training.de/tipps

Schulungszentrum

Munisoft Training GmbH
Grünwalder Weg 13a
82008 Unterhaching/München

Mehr Oracle Schulungstermine unter
munisoft-training.de

Auf Anfrage bieten wir auch gerne individuelle Inhouse Schulung an



7

Interview mit Juan R. Loaiza



37

GraphQL als weitere Option beim Design und der Implementierung eines API



61

Datenbankadministratoren sehen sich heute fünf wesentlichen Herausforderungen gegenüber



74

Immer öfter im „Werkzeugkoffer“ moderner Unternehmen zu finden : Agile Führung

Einleitung

- 3 Editorial
- 5 Timeline
- 6 Aus der Ferne betrachtet: Ein Dutzend Jahre sind in der IT eine halbe Ewigkeit
- 7 „Oracle wird sein Angebot an Datenbanken weiterhin auf allen Plattformen unterstützen und verbessern: auf generischer Hardware, auf kundenspezifischen Systemen und in der Cloud.“
Interview mit Juan Loaiza

Datenbank

- 10 Exadata X8M – The Next Big Thing?
Interviews mit Frank Schneede
- 61 Performant unterwegs in heterogenen Datenbanken
Ales Zeman
- 70 Senkung der Betriebskosten durch Migration auf Open Source? Ja, aber ...
Grégory Steulet

Engineered Systems

- 16 Hilfreiche Tools für den täglichen Exadata-Betrieb (Teil 1)
Michael Schulze
- 22 Virtualized Exadata – die vier ersten produktiven Jahre...
Daniele Massimi
- 29 ODA mit Lizenzmangel
Marco Mischke

Development

- 34 Die neue Oracle-Forms-Version 12.2.1.4 und aktuelle Trends
Frank Hoffmann
- 37 Brainwash your Developers – Richtiges Java für PL/SQLer
Wolf G. Beckmann
- 42 Bessere APIs dank GraphQL?
Markus Lohn

- 58 Agil unterwegs in Infrastruktur-Projekten, eine sinnvolle Alternative?
Casimir Schmid
- 64 Standardkonform oder doch Oracle-proprietär?
Jürgen Sieben
- 68 Der DOAG Projektpreis 2019 geht an fabel!
Marcos López

Infrastruktur

- 47 Konsolidierung und Resource Management
Stefan Seck

APEX

- 52 Ein APEX Workflow-Tool für Citizen Developer
Michael Weinberger

Agile Unternehmenskultur

BUSINESS NEWS

- 75 Agile Vorgehensmodelle in Serviceorganisationen
Carsten Wiesbaum & Ingo Sobik
- 79 Erfolgreiche Kommunikation in agilen Projekten ist eine Frage der Beziehung
Carsten Firus
- 84 Führung im Kontext der agilen Arbeit
Daniel Reinol

KI-Commerce

BUSINESS NEWS

- 90 Der Handel und die Auswirkungen der Digitalisierung
Jens Horstmann

Intern

- 93 neue Mitglieder
- 93 Termine
- 94 Impressum
- 94 Inserenten

Timeline

02. Dezember 2019

Das Regionaltreffen München/Südbayern findet in München statt. Zum Jahresabschluss geht es noch mal um das Thema Datenbank-Entwicklung. Dietmar Neugebauer & Wolfgang Taschner begrüßen die Teilnehmer und berichten über Neues von der DOAG. Ulrike Schwinn, Oracle, referiert zum Thema Oracle als Multi-Model-Datenbank. Stefanie Bartl, Opitz Consulting, spricht über nützliche, aber oft vergessene SQL- und PL/SQL-Features.

06. Dezember 2019

Der DOAG-Vorstand kommt in Berlin zu seiner letzten Sitzung im Jahr 2019 zusammen und plant die Aktivitäten für das Jahr 2020. Auf der Agenda stehen die Planung 2020 und die Vorbereitung der Delegiertenversammlung im April. Den größten Zeitanteil nimmt die Öffnung der DOAG ein. Der Vorstand hat Fokus-Themen der nächsten Jahre herausgearbeitet. Diese werden nun von den Communities weiter heruntergebrochen. Ziel ist, die Fokus-Themen verstärkt in die Veranstaltungen und Medien einzubringen.

13. Dezember 2019

Das DOAG Datenbank-Webinar zum Thema "Transportmöglichkeiten mit Cascading/Cascaded Standby DBn, Active Data Guard Far Sync und Redoroutes" mit Marc Wagner, dbi services, findet statt.

20. Dezember 2019

Die DOAG-Geschäftsstelle beendet das Jahr 2019 und schließt das Berliner Büro über die Feiertage. Fried Saacke, DOAG-Vorstand und Geschäftsführer, blickt mit seinem Team auf ein sehr erfolgreiches Jahr zurück.

02. Januar 2020

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

10. Januar 2020

Das DOAG Datenbank-Webinar zum Thema "Flashback-Technologien und wie man sie anwendet" mit Referent Marco Mischke, Robotron Datenbank Software, steht an.

23. bis 24. Januar 2020

Auf der sechsten DOAG 2020 Noon2Noon in München steht die Frage „Oracle und VMware - Liebe auf den ersten Blick?“ im Mittelpunkt. Auf der praxisorientierten Veranstaltung besteht für die 38 Teilneh-

mer von 12 bis 12 Uhr die Möglichkeit, unter fachkundiger, neutraler Anleitung selbst Hand anzulegen und damit das Optimum aus der Datenbank und auf der virtuellen Umgebung herauszuholen. Ziel ist es, ein tieferes Wissen über den Betrieb von Oracle-Datenbanken auf VMware zu erlangen, um so den Infrastrukturkollegen zu helfen, das optimale Setup für die Guests festzulegen. Natürlich wird dabei auch auf das Thema Lizenzierung eingegangen.



23. Januar 2020

Die DOAG-Geschäftsstelle trifft sich zum Kick-off 2020. Fried Saacke berichtet über die Ergebnisse des vergangenen Jahres und stimmt auf die vor uns liegenden Herausforderungen ein. Hierzu startet Mylène Diacquenod mit einem Impulsvortrag. Change in vieler Hinsicht ist das Gebot des Jahres. Gemeinsam werden die zu verwirklichenden Ziele und Visionen erarbeitet. Anschließend geht es zur Bewährung in vier Gruppen in die unterirdischen Escape-Rooms eines ehemaligen Fernmeldebunkers in Berlin-Mitte.

12. bis 13. Februar 2020

Im Berliner Expertenseminar zum Thema „Gestaltung moderner User Interfaces mit APEX 19.2“ mit Moritz Klein und Carsten CzarSKI lernen die 19 Teilnehmer die unterschiedlichen Komponenten zur Gestaltung von Dialogen kennen. Alle Konzepte werden von den Teilnehmern in Hands-on Sessions selbst ausprobiert.

20. Februar 2020

Zum vierten Mal findet der DOAG Forms Day, diesmal in Frankfurt am Main, statt. Neben Fachvorträgen zum Thema "Oracle Forms" gibt es bereits am Vorabend Gelegenheit für ein kleines Networking.

27. Februar 2020

Die zweite Auflage des NetSuite User Days steht in Hamburg auf dem Programm. Matthias Runte, der stellvertretene NetSuite-Themenverantwortliche der DOAG, moderiert die Veranstaltung, die neben Vorträgen auch die Gelegenheit bietet, gegenüber den anwesenden NetSuite-Vertretern Wünsche und Kritik zu äußern.

Aus der Ferne betrachtet:

Ein Dutzend Jahre sind in der IT eine halbe Ewigkeit



Günther Stürner

Gäbe es heute eine Exadata, wenn damals, vor einem guten Dutzend Jahren die damaligen Größen der Plattenhersteller auf eine Idee von Oracle eingegangen wären? So sicher bin ich mir da nicht.

Aber der Reihe nach.

Wir saßen bei einem großen und wichtigen Kunden und er eröffnete uns, dass die Oracle-Datenbank für seine Bedürfnisse – es handelte sich um ein Data-Warehouse-System – völlig ungeeignet sei. „Viel zu langsam. Der Vorstand reißt uns den Kopf ab, weil alles so lange dauert. Wir brauchen einen radikalen Wechsel. Oracle fliegt raus usw...“ Wir kamen kaum zu Wort bei all den Vorwürfen und Schuldzuweisungen.

Es gibt meist nicht den einen Grund für ein Problem. Bei 600 Megabyte pro Sec Durchsatz – im Maximum(!) – war das Primärproblem aber offensichtlich.

Den Grauhaarigen unter uns wird das nicht gänzlich unbekannt vorkommen. Die Daten, die von der Platte zum Server wollten, stellten sich ordentlich in der Schlage an, der DB-Server langweilte sich zu Tode, das „Drähle“ zwischen Platten und Server glühte, die Endanwender hatten übersäuerte Mägen vom vielen Filterkaffee und die Datenbank war an allem schuld.

So konnte es nicht weitergehen. Eine Idee war es, ein Stück Software auf den Plattensystemen zu installieren, das im Zusammenspiel mit dem Datenbank-Server die Anzahl der Daten, die transportiert werden müssen, reduzieren sollte. Datenbank-Intelligenz auf den Platten (heute bekannt als SMART-Scan). Coole Idee. Es funktionierte auch perfekt. Nur die EMCs der da-

maligen Zeit verspürten keine große Lust, Oracle SW zu lizenzieren und auf ihren Systemen zu installieren.

Das war ein herber Rückschlag, der umso mehr schmerzte, weil just zu jener Zeit einige neue Spieler im Datenbank-Umfeld auftauchten und sich anschickten, den Data-Warehouse-Markt mit „Datenbank-Maschinen“ zu revolutionieren. Greenplum und Netezza, um zwei wichtige Protagonisten zu nennen, hatten ihre große Zeit – damals.

Wie schrieb Juan Loaiza, einer der Datenbank- und Exadata-Entwicklungsmanager bei Oracle kürzlich in einem Artikel? „Take a good idea and make it better.“ Genau so muss man sich das bei der Exadata vorstellen. Die Konkurrenten hatten gezeigt, dass die Kundschaft gewillt ist, solche Gesamtsysteme zu kaufen, und für die Datenbankentwickler war es mehr als verlockend, einen Server zu konzipieren, der alles das hat, was man sich als „Datenbanker“ wünscht.

Die Vorarbeiten waren gemacht, die Software-Architektur war festgelegt, jetzt brauchte es nur noch den Schritt hin zur Hardware, der mit HP, einem der damals wichtigsten Partner, ohne große Diskussionen feststand. Die Exadata V1 war da und sorgte für mächtigen Wirbel.

Heute ist die Exadata ein fester Bestandteil des Oracle-DB-Universums und unverzichtbar für viele Kunden, aber auch unverzichtbar für Oracle. Die Exadata als Grundlage für die Oracle-DB-Cloud macht den radikalen Unterschied zwischen den Cloud-Anbietern und mit der Exadata X8M ist ein Quantensprung gelungen, der alles Bisherige, inklusive des nur wenige Monate älteren Bruders Exadata X8 (ohne M), in einen dunklen Schatten stellt.

Jedem, der mit einer Exadata arbeiten möchte, sei der Autonomous Database Service (ADW oder ADT) empfohlen – gibt es auch als kostenfreie Variante.

Exadata for free! Es gab schon schlechtere Ideen.

Kommentare, Anregungen, Lob – und wenn es sein muss, auch Tadel – an: guenther.stuerner@dbms-publishing.de

Richtigstellung

In der Ausgabe 5/2019 ist uns leider ein Fehler unterlaufen. Im Artikel „SQLXPLAIN – Helfer bei der Performance-Analyse“ von Stefan Seck wurden bei den *Abbildungen 1 und 2* falsche Bilder

abgedruckt. Wir möchten dies korrigieren und bieten den Artikel von Stefan Seck mit den korrekten beiden Abbildungen unter diesem Link zum Download an: www.doag.org/go/Stefan_Seck



„Oracle wird sein Angebot an Datenbanken weiterhin auf allen Plattformen unterstützen und verbessern!“

Günther Stürner, Datenbank-Berater und Coach, dbms-publishing, unterhielt sich mit Juan Loaiza, Executive Vice President, Mission-Critical Database Technologies bei Oracle, über die Entwicklung, die Technik und den Einsatz von Oracle-Exadata-Systemen

Können Sie uns bitte eine kurze Zusammenfassung Ihrer Zeit bei Oracle und Ihrer derzeitigen Zuständigkeiten in der Oracle-Datenbankentwicklungsgruppe geben?

Seit fast 20 Jahren bin ich bei Oracle für die Entwicklung und das Produktmanagement von unternehmenskritischen Datenbanktechnologien verantwortlich. Ich habe 1988 bei Oracle angefangen und persönlich an vielen verschiedenen Bereichen der Oracle-Datenbank gearbeitet, darunter Management, Performance, Hochverfügbarkeit, Engineered Systems und Cloud-Datenbanken.

Oracle Exadata erblickte im Jahr 2008 das Licht der Welt. Dies war eine große strategische Veränderung für Oracle von einem reinen Software-Unternehmen zu einem Software-Unternehmen, das auch für die Bereitstellung von Hardware verantwortlich ist. Ich nehme an, das war das Ergebnis vieler interner, vielleicht sogar kontroverser Diskussionen?

Ja, wir hatten natürlich viele interne Diskussionen zu diesem Thema. Es war in der Tat eine große Veränderung für Oracle, in den Hardwaremarkt einzusteigen. Ursprünglich arbeiteten wir mit HP zur Herstellung der Hardware zusammen, was weni-

ger strittig war. Kurz danach erwarb Oracle allerdings Sun Microsystems und begann mit der Herstellung eigener Hardware. Der Eintritt in den Hardwaremarkt war für Oracle schon ein gewisses Risiko, im Nachhinein jedoch eine ausgezeichnete Entscheidung, da dies unseren Übergang zum Cloud-Anbieter erleichterte.

Die Kunden waren beeindruckt, als sie die ersten Ergebnisse von Oracle Exadata sahen. Larrys Botschaft war nicht nur Marketing, sondern Realität. Doch obwohl die Maschine außergewöhnliche Ergebnisse zu liefern schien, äußerten die Kunden die Befürchtung, dass Exadata sie noch stärker an Oracle binden würde.

Exadata war noch nie eine Lock-in-Plattform. Für Kunden ist es sehr einfach, ihre Datenbanken auf Exadata zu migrieren, und es ist für sie genauso einfach, ihre Datenbanken von Exadata zu migrieren. In beiden Richtungen sind keine Änderungen an den Anwendungen erforderlich. Kunden werden sich an die hohe Leistung und Verfügbarkeit von Exadata gewöhnen und möchten möglicherweise nicht mehr langsamer und mit längeren Ausfallzeiten auf einer anderen Plattform arbeiten. Ich wür-

de dies mit echten Lock-in-Plattformen wie Mainframes vergleichen, bei denen der Kunde proprietäre Software verwendet, die nicht auf anderen Plattformen funktioniert und daher seine Anwendungen neu schreiben muss, um sie auf eine andere Plattform zu verschieben. Cloud-Anbieter wiederholen jetzt diesen Lock-in-Effekt.

Ein weiterer häufig geäußelter Punkt war, dass ein Oracle-Exadata-System nur für die größten Kunden sinnvoll ist.

Bei der ersten Auslieferung von Exadata verkauften wir nur ein vollständiges Rack-System, das auf große Workloads beschränkt war. Anschließend haben wir elastische Exadata-Konfigurationen eingeführt, mit denen Kunden die Anzahl der benötigten Rechen- und Speicherserver konfigurieren können. Außerdem lassen sich auf einem einzelnen Exadata-System viele Datenbanken betreiben, sodass auch kleinere Kunden es verwenden können, da sie keine Exadata für jede Datenbank benötigen. Mit unserer auf Exadata in der öffentlichen Cloud laufenden Oracle Autonomous Database können Kunden nun jede gewünschte Datenbankgröße konfigurieren, auch sehr kleine. Wir haben kürzlich eine neue Stufe der Oracle Autonomous Database eingeführt, mit der Kunden Exadata kostenlos nutzen können.

Nach elf Jahren und zehn Releases von Oracle Exadata sind das Konzept und die Architektur noch immer dieselben. Eine Unmenge an Verbesserungen, aber immer noch dieselbe Architektur. Das ist in der IT-Branche sehr selten.

Die Hardware-Architektur ist konzeptionell sehr einfach: Ein Scale-Out-Grid von Rechenservern zum Ausführen von Datenbanken sowie ein Scale-Out-Grid von Smart-Storage-Servern zum Speichern und Verarbeiten von Daten und das alles verbunden mit Hochgeschwindigkeitsleitungen, basierend auf Großserien-Hardware auf dem neuesten Stand der Technik, wobei der gesamte Stack vollständig für die Datenbank optimiert ist. Mit dieser Architektur können wir auf jede Ebene skalieren und dabei Engpässe vermeiden. Die grundlegende Hardware-Architektur bleibt zwar konsistent, wir werden aber auch weiterhin fantastische Innovationen hinzufügen, wie den persistenten Speicher in Oracle Exadata X8M. Die wahre Magie von Exadata ist auch die Software, die sich in die Datenbank einfügt und diese auslagert. Diese Software wurde seit der Erstveröffentlichung von Exadata grundlegend geändert und verbessert.

Wie profitieren Kunden von der Tatsache, dass alle Oracle-Exadata-Generationen dieselbe Architektur verwenden?

Kunden können problemlos neue Exadata-Hardware-Generationen übernehmen und ihre vorhandenen Exadata-Konfigurationen mit neuer Hardware erweitern. Die neuen Versionen der Exadata-Software können sowohl auf neuer Hardware als auch auf älteren Generationen von Exadata-Hardware ausgeführt werden. Wir haben viele Kunden, die zwei und drei Hardware-Generationen in einem einzigen System betreiben. Dies gibt Kunden die Möglichkeit, Kapazitäten einfach nach Bedarf hinzuzufügen, ohne vorhersehen zu müssen, was sie in Zukunft benötigen.

Wenn wir uns die Oracle-Exadata-Zeitleiste ansehen, ist das Jahr 2019 schon etwas Besonderes. Sie haben im April eine neue Generation veröffentlicht – Oracle Exadata X8 – und im September Oracle

Exadata X8M. Warum erschienen innerhalb von sechs Monaten zwei Systeme? Wie unterscheiden sich die beiden Geschwister?

Es gibt zwei wesentliche Unterschiede zwischen den Oracle Exadata X8M- und den Oracle Exadata X8-Systemen. Der erste Unterschied besteht darin, dass wir von einer internen InfiniBand-Netzwerkstruktur zu einer 100-Gbit-Ethernet-Struktur übergegangen sind, auf der das RDMA Over Converged Ethernet (RoCE)-Protokoll ausgeführt wird. Beim ersten Entwurf zu Exadata haben wir versucht, Mainstream-Ethernet als Fabric zu verwenden, aber es war zu langsam und es fehlten verlustfreie Protokolle und RDMA-Protokolle. Mittlerweile verfügt Ethernet über all diese Funktionen, sodass wir zu einem verbreiteteren Fabric wechseln konnten. Der zweite große Unterschied ist, dass Exadata X8M Dauerspeicher einführt, eine neue Technologie, die einen viel schnelleren Zugriff auf gespeicherte Daten ermöglicht. Wir haben eine Vielzahl neuer Software entwickelt, um den Dauerspeicher zu nutzen und mithilfe von RDMA-Protokollen einen Zugriff auf Daten auf Speicherservern mit sehr geringer Latenz zu ermöglichen. Der Grund für die Einführung von Exadata X8M sechs Monate nach Exadata X8 ist, dass die neuen Netzwerktechnologien und Dauerspeichertechnologien zum Zeitpunkt der Auslieferung von Exadata X8 noch nicht für den unternehmenskritischen Einsatz bereit waren. Wir benötigten weitere sechs Monate, um sicherzustellen, dass diese neuen Technologien in puncto Qualität und Stabilität für äußerst unternehmenskritische Workloads bereit waren.

In den letzten Jahren war Oracle Exadata eine erfolgreiche Maschine beim Kundenstamm von Oracle, aber Exadata hat sich auch als ein wichtiger Teil der Oracle-Cloud-Strategie etabliert. Können Sie bitte umreißen, wie Oracle die Exadata-Plattform für die eigenen Cloud-Angebote nutzt?

Ja, Exadata ist seit langem eine sehr erfolgreiche Technologie im On-Premises-Bereich. Heute nutzen über 80 Prozent der größten Banken, Telekommunikationsunternehmen und Handelsunternehmen sowie Tausende kleine Unternehmen Exadata. Exadata ist ein wesentlicher Teil von Oracles Cloud-Strategie. Kunden können ein Exadata-System in der Oracle Cloud buchen und alle Vorteile nutzen, die sie von Exadata bei der Bereitstellung On-Premises kennen. Zusätzlich bekommen sie eine Vielzahl an Cloud-Automatisierung für die Administration ihrer Datenbanken, ein dynamisches Abonnement anstatt des Systemkaufs und Oracle-Experten erledigen die Administration der Infrastruktur. Kunden, die möchten, dass Oracle alle Datenbankoperationen automatisiert und verwaltet, bieten wir Oracle Autonomous Database an, das ebenfalls auf Exadata läuft.

Bei Betrachtung der heutigen Rechenzentren, sehen wir viele virtualisierte Standard-Server. Auf einer Vielzahl von ihnen laufen Oracle-Datenbanken. Weshalb sollten sie auf eine Exadata-Umgebung wechseln? Was sind für sie die größten Vorteile?

Der einfache Grund, weshalb Tausende von Unternehmenskunden mittlerweile ihre unternehmenskritischen OLTP- und Warehouse-Datenbanken auf Exadata betreiben, sind die wesentlich bessere Performance, Verfügbarkeit und Verwaltung zu niedrigeren Kosten. Es gibt viele einzigartige Softwarefunktionen in Exadata wie die Möglichkeit von Anfragenverlagerung in den Speicher für Analysezwecke und die Möglichkeit zum Ausfüh-

ren von RDMA nutzenden Datenbankkommunikationsprotokollen, die auf keiner anderen Plattform verfügbar sind. Oracle hat Hunderte von Technikern, die annähernd zwei Jahrzehnte an Erfahrung mit dem Schreiben von Exadata-Software und der Optimierung der gesamten Plattform haben. Viele Kunden waren anfangs skeptisch, ob die Vorteile der Wahrheit entsprachen, da sie zu gut klingen, um wahr zu sein. Aber nachdem sie die Technologie ausprobiert und übernommen hatten, konnten sie für sich selbst prüfen, dass die Vorteile sehr wohl wahr sind, und sie haben ihre Nutzung von Exadata stetig gesteigert.

Wenn ein Kunde von Oracle Exadata begeistert ist... Welche Einsatzmöglichkeiten hat ein solcher Kunde heute? Er kann es kaufen und in seinem eigenen Rechenzentrum betreiben, aber was ist sonst noch möglich?

Kunden können beim Einsatz von Exadata aus vier Optionen wählen. Die erste ist die traditionelle Option, Exadata zu kaufen und selbst zu betreiben. Die zweite Möglichkeit besteht darin, Exadata in der Oracle Cloud zu abonnieren. Die dritte Möglichkeit besteht darin, Oracle Autonomous Database in der Oracle Cloud zu betreiben und unter der Haube Exadata zu verwenden. Die vierte Option ist, Oracle Cloud Exadata in einem eigenen Rechenzentrum bereitzustellen. Wir nennen dies Oracle Exadata Cloud beim Kunden.

Wenn wir uns die Feature-Liste der Oracle-Datenbank ansehen, ist es erstaunlich, dass einige Features, bei denen es sich anscheinend um klassische Enterprise-Edition-Features handelt, nur in einer Oracle-Exadata-Umgebung verfügbar sind. Heißt das, dass die frü-



Zur Person: Juan Loaiza

Juan Loaiza ist Executive Vice President für geschäftskritische Datenbanktechnologien bei Oracle; er berichtet an den Oracle Chairman und CTO Larry Ellison. Loaiza ist verantwortlich für die Leitung der Produktstrategie, Produktentwicklung und des Produktmanagements für die Transaktionsverarbeitungs- und technischen Systemtechnologien in der Public Cloud und on-premises. Loaiza hat einen BS- und MS-Abschluss in Informatik vom Massachusetts Institute of Technology (MIT). Im Jahr 1988 verließ er das MIT-Doktorandenprogramm, um sich dem Oracle Database Engineering-Team anzuschließen und ist seitdem ein Innovator im Bereich der Datenbanktechnologien.

here Vorzeige-Edition – die Enterprise Edition – in den nächsten Jahren an Boden verliert oder, noch etwas provokativer ausgedrückt: Haben lokale und Nicht-Oracle-Exadata-Systeme eine Zukunft?

Oracle wird sein Angebot an Datenbanken weiterhin auf allen Plattformen unterstützen und verbessern: auf generischer Hardware, auf kundenspezifischen Systemen und in der Cloud. Trotzdem ist es klar, dass der IT-Markt im Allgemeinen schnell zu einem Cloud-Modell übergeht. Daher konzentriert sich Oracle wie andere Anbieter stark (aber nicht ausschließlich) auf die Cloud und die Exadata-Plattform, die als Basis unserer Cloud dient.

Ich bin ein großer Anhänger der In-Memory-Datenbank. Das ist eine wirklich starke Technologie. Aber warum sollte jemand Oracle Exadata verwenden, wenn er mit einer klassischen Server-Umgebung zusammen mit In-Memory eine hohe Performance erzielen kann? Und eine zweite Frage: Ist es sinnvoll, ein Oracle-Exadata-System zusammen mit der In-Memory-Datenbank zu betreiben?

Die Verwendung von Oracles In-Memory-Technologien auf Exadata bietet viele Vorteile. Oracle Exadata implementiert Oracles Spaltenformate und Verarbeitung auf der Speicherebene und im Flash-Speicher und verwendet auf den Datenbankservern zusätzlich In-Memory-On-DRAM. Durch die Implementierung der spaltenorientierten Technologie in Exadata-Flash in Verbindung mit DRAM profitieren Kunden von der In-Memory-Verarbeitung für wesentlich größere Datensätze. Darüber hinaus bietet Exadata eine wesentlich höhere Verfügbarkeit für In-Memory-Workloads, indem In-Memory-Strukturen serverübergreifend dupliziert werden. Dies ähnelt konzeptionell dem Schutz von Speichersystemen vor Festplattenausfällen durch Implementierung von Spiegelung über Festplatten und Flash-Laufwerke. Insgesamt ist In-Memory auf Exadata schneller, bietet eine höhere Kapazität und ist besser verfügbar, verglichen mit allen anderen Plattformen.

Die Ankündigung der Oracle Autonomous Database sorgte für viel Wirbel in der Presse. Welche Idee steckt hinter diesem Angebot?

Oracle Autonomous Database ist der nächste große Wandel in der Datenbankbranche. Dies ist vergleichbar mit selbstfahrenden Autos als der kommende große Umbruch für die Automobilindustrie. Die jüngsten Fortschritte beim maschinellen Lernen haben es uns ermöglicht, die Datenbank selbstverwaltend, selbstsichernd und selbstreparierend zu gestalten. Auf diese Weise entfällt der Wartungsaufwand beim Betrieb einer unternehmenskritischen Datenbank, sodass sich Datenbankfachleute auf die Steigerung des Geschäftswerts konzentrieren können und nicht auf mühsame Routinewartung.

Vielen Dank, Juan. Ich habe nur noch eine weitere Frage. Die Oracle Autonomous Database ist so etwas wie der „Heilige Gral“ der Datenbanktechnologie. Können Kunden in Zukunft autonome Funktionen oder sogar eine vollständige ADB-Umgebung in ihren eigenen Rechenzentren erwarten?

Die Oracle Autonomous Database ist von Haus aus eine Cloud-Technologie, da sie nicht nur die Datenbank, sondern den gesamten Stack einschließlich Datenbank-, Infrastruktur-, Netzwerk- und Lebenszyklusoperationen automatisiert. Wir planen, Oracle Autonomous Database On-Premises verfügbar zu machen, und zwar zusätzlich zu unserer Gen2 Oracle Exadata Cloud auf der Kundenplattform im Jahr 2020.



The Next Big Thing

Exadata X8M – The Next Big Thing?

Frank Schneede, Oracle Deutschland

Von der Exadata Database Machine gibt es seit der Oracle Open World 2019 ein neues Modell, die Exadata X8M, die eine komplette Neukonzeption der über lange Jahre bewährten Architektur darstellt. Mit neuem internen Netzwerk, persistentem Memory und neuer Virtualisierungstechnologie wartet die neue Exadata mit beeindruckenden Performancekennzahlen auf. Ist sie jetzt „The Next Big Thing“, auf das die Kunden schon lange gewartet haben? Dieser Artikel beschreibt die neue Architektur anhand des 2-Sockel-High-Capacity-Modells sowie die darin liegenden Möglichkeiten für Neu- und Bestandskunden.

Seitdem die Exadata im Jahr 2008 erstmalig vorgestellt worden ist, wurde sie regelmäßig an aktuelle Hardwaregenerationen angepasst. Allein durch das „Schneller, Höher, Weiter“ der Aktualisierung und Erweiterung der Hardware kann jedoch keine Lösung mit den exponentiell wachsenden Anforderungen in der IT schritthalten. Daher hat es mit jedem Modellwechsel auch umfangreiche funktionale Erweiterungen der eingesetzten Software sowohl in der **Exadata Storage Server Software** als auch in der Oracle-Datenbank gegeben. Nur so war es möglich, die Spitzenposition als leistungsfähigste Plattform für den Betrieb von

Oracle-Datenbanken bis heute zu bewahren. *Abbildung 1* zeigt die Exadata-Evolution über die Jahre in Schlagworten.

Im Rahmen der Exadata-Weiterentwicklung wurden und werden natürlich auch neue Technologien evaluiert und für die Exadata adaptiert, wenn diese für den Anwender einen Nutzen darstellen. So wurde bereits länger über den Einsatz von persistentem Memory und schnellem Ethernet in der Exadata spekuliert. Mit dem Announcement von der diesjährigen OOW wurde aus der Spekulation Gewissheit. Drei wesentliche Änderungen kennzeichnen die Exadata Database Machine X8M:

- 100Gb/s Ethernet als internes Netzwerk mit RoCE (RDMA over Converged Ethernet) Protokoll
- Intel® Optane™ Persistent Memory Modules im Storage Server
- KVM als Virtualisierungslösung

Exadata X8M-2 Hardware-Änderungen

Wie das Exadata-Modell X8-2 setzt die X8M-2 in den Datenbankservern auf aktuelle „**24-Core 2.4GHz Intel® Xeon® (Cascade Lake)**“-Prozessoren, auch deren

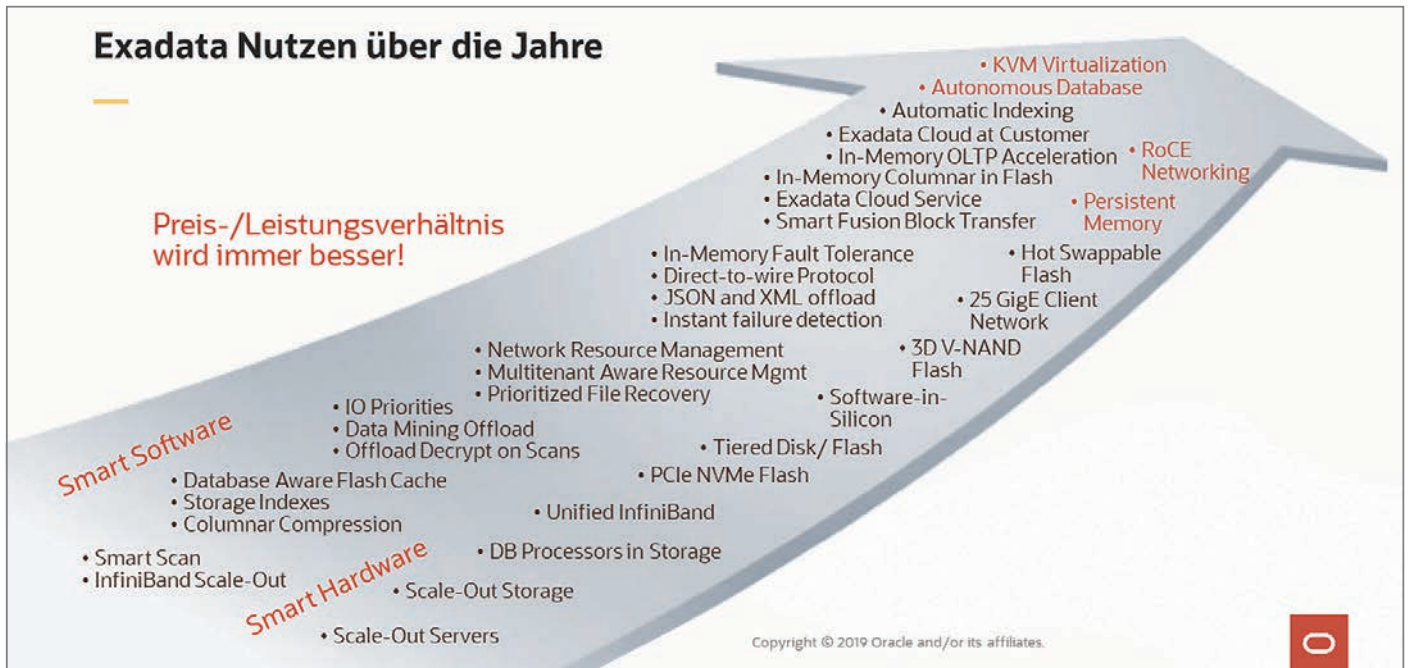


Abbildung 1: Exadata HW- und SW-Evolution (Quelle: © Oracle)

Ausstattung mit Hauptspeicher (Standard 384GB, erweitert 768GB pro DB Server) ist identisch geblieben. Neu ist der Einsatz der **QSFP28 100GbE RDMA Ethernet Card** anstelle der bisher verwendeten InfiniBand-Adapter für das interne Netzwerk.

Als Switches für die Kommunikation zwischen Datenbank und Storage-Servern werden nun „**Cisco Nexus 9336C-**

FX2“-Switches verwendet, über die das neue RoCE-Protokoll läuft.

Die Storage-Server verfügen im High-Capacity-Modell über **14TB - 7.200 RPM Disk Drives** und **6.4TB Flash Accelerator F640 NVMe Cards**, der Netzwerkadapter ist entsprechend auch die **QSFP28 100GbE RDMA Ethernet Card**. Zwölf neue **128 GB Intel® Optane™ DC Per-**

sistent Memory Modules bilden im Modell X8M eine neue Storage Tier mit insgesamt 1,5TB persistentem Speicher, die in jedem Storage Server zur Verfügung stehen. Der bisher eingesetzte Hauptspeicher ist mit 12 x 16GB-DDR4-DIMM-Modulen bestückt und im Gegensatz zum Exadata-Modell X8-2 nicht erweiterbar. *Abbildung 2 zeigt die Veränderung der*

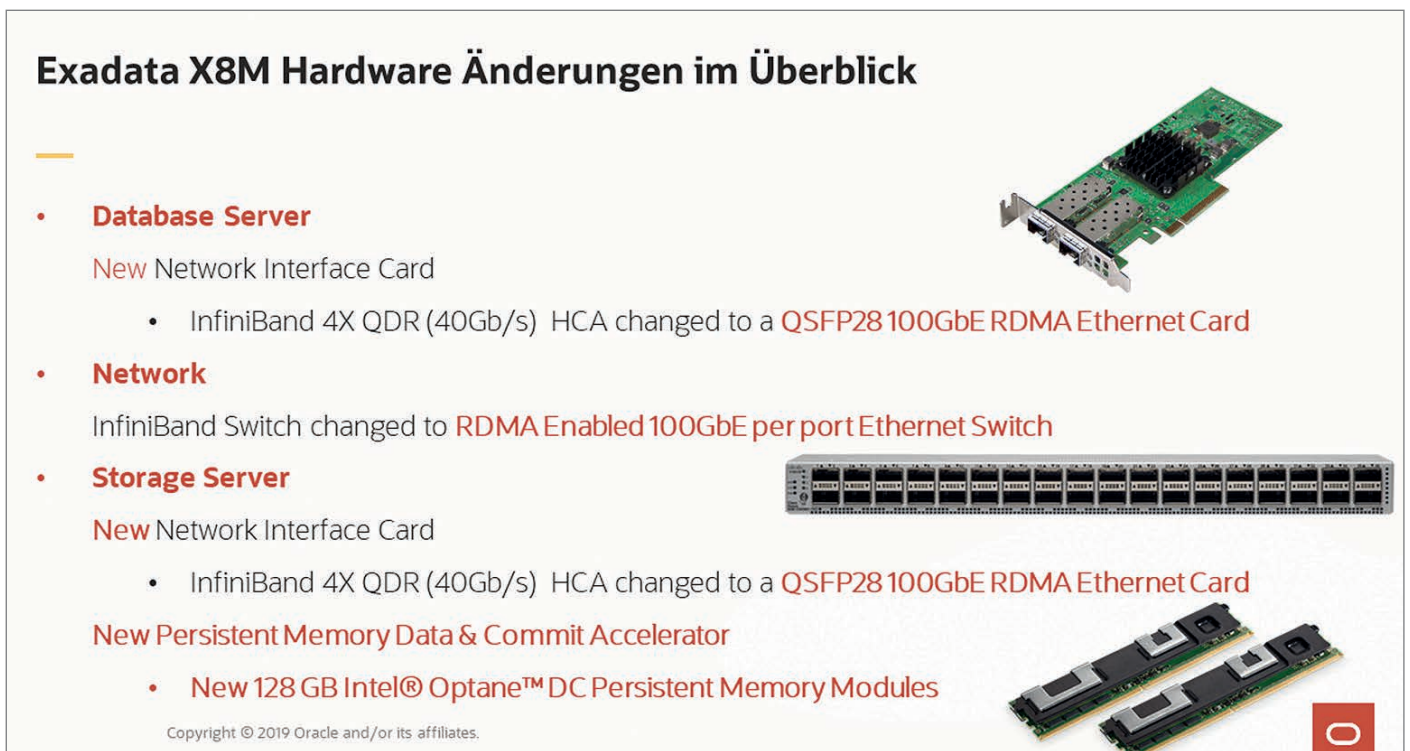


Abbildung 2: Exadata X8M Hardware-Änderungen (Quelle: © Oracle)

Hardware-Ausstattung der Exadata X8M-2 im Vergleich zur Exadata X8-2.

Bereits mit der Exadata X8 wurden im April 2019 als wesentliche Änderung die sogenannten **Exadata Storage Server X8-2 Extended (XT)** eingeführt, die als kostengünstige Storage Server ins Exadata-interne Netzwerk eingebunden werden können, um zum Beispiel historische Daten direkt auf der Exadata im Zugriff zu haben. Diese Storage Server beinhalten keinen Flash-Speicher und benötigen keine „**Exadata Storage Server Software**“-Lizenz, was sie zu einer sehr günstigen Alternative zu extern anzubindenden Speichersystemen (zum Beispiel ZFS Storage Appliance) macht. **Exadata Storage Server X8M-2 Extended (XT)** gibt es auch für die neue Architektur, der einzige Unterschied zum Pendant der Exadata X8-2 besteht in der internen Netzwerkanbindung. Über persistenten Memory verfügen diese Server nicht, da für das Ansprechen des Memory die **Exadata Storage Server Software** zuständig ist.

Die Einstiegsvariante in die **Exadata-Welt ist das Exadata Database Machine X8M-2 High Capacity (HC) Eighth Rack**, in dem nur ein Sockel der verbauten Server mit einem Prozessor bestückt ist und die Anzahl der verbauten Flash-Module beziehungsweise Festplatten je Server halbiert ist. Die Datenbankserver lassen sich durch Hinzufügen der fehlenden Hardwarekomponenten zum Quarter Rack erweitern, ein Ausbau der HC Storage Server ist nicht vorgesehen. Kunden, die das Eighth Rack erweitern möchten, haben daher mit dem **Exadata X8M-2 Eighth Rack Storage Server HC** die Möglichkeit, das System sehr granular an gestiegene Leistungsanforderungen anzupassen. Auch im Eighth Rack der Exadata kommen insgesamt 1.5TB **Intel® Optane™ DC Persistent Memory Modules** je Storage Server zum Einsatz.

RDMA over Converged Ethernet (RoCE)

Bis einschließlich des nach wie vor aktuellen Modells Exadata X8-2 erfolgte die Kommunikation zwischen Storage und Datenbank über InfiniBand. Obwohl der Anwender ja eigentlich kaum Berührungspunkte mit dem internen Netzwerk der Exadata hat, haben die Kunden mit

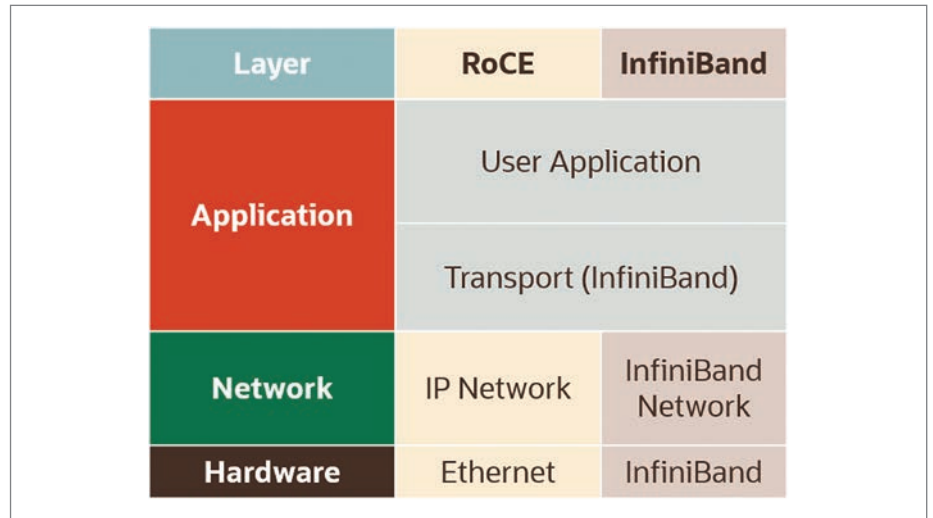


Abbildung 3: Exadata RoCE vs. InfiniBand (Quelle: Frank Schneede)

der InfiniBand-Technologie ein wenig „gefremdet“. Bislang stand jedoch kein adäquater Ersatz für InfiniBand zur Verfügung, mit dem eine ähnlich latenzarme und performante Netzwerkverbindung zu realisieren gewesen wäre. In Verbindung mit RDMA (Remote Direct Memory Access), dessen Nutzung in der InfiniBand-Architektur-Spezifikation definiert ist, war ein extrem zuverlässiger und performanter Betrieb des Exadata-Clusters möglich.

Die InfiniBand Trade Association (IBTA), ein Firmenkonsortium, in dem die größten Anbieter von Netzwerk-Karten (Broadcom, Intel, Mellanox) und Netzwerk Switches (Arista, Cisco, Juniper, Mellanox) zusammenarbeiten, hat für eine schnelle und latenzarme Datenübertragung netzwerkintensiver Anwendungen das **RoCE (RDMA over Converged Ethernet)**-Netzwerkprotokoll entwickelt. RoCE ist ein offener Standard, der auf Linux entwickelt wurde und schnelle und sichere RDMA-Zugriffe auf Ethernet ermöglicht. In der Exadata X8M kommt RoCE zum Einsatz; *Abbildung 3* zeigt, dass sich durch den Einsatz von 100GbE für die interne Kommunikation lediglich auf Hardware- und Netzwerkebene des 7-Schichten-Modells Änderungen ergeben!

Das bedeutet, dass große Teile des bewährten InfiniBand-Softwarestacks, der auf der Exadata X8M für die Kommunikation eingesetzt wird, weiterhin verwendet werden. RDMA sorgt für den direkten Zugriff eines Servers auf den Prozesspeicher eines anderen Servers unter Umgehung von OS und CPU und ist daher nach wie vor ein integraler Bestandteil der

Exadata-Architektur, ohne den folgende Funktionalitäten nicht möglich wären:

- Hoher Durchsatz bei geringer CPU-Last für die **Übertragung großer Datenmengen**
- **Direct-to-Wire**-Protokoll für schnelle OLTP-Kommunikation im Cluster
- **Smart Fusion Block Transfer** vermeidet Log-Writes beim Übertragen von Datenblöcken zwischen Knoten
- **Koordination von Transaktionen** zwischen Knoten
- **Low-Latency-Zugriff auf persistenten Memory** im Storage Server (nur mit X8M und 19c)
- **Low-Latency Database Log-Writes auf persistenten Memory** im Storage Server (nur mit X8M und 19C)

RDMA wird auf allen unterstützten Datenbankversionen (11.2.0.4 bis 19.4.0.0) verwendet, die Optimierungen in Bezug auf persistenten Memory setzen die aktuelle Datenbank-Version 19c voraus.

Für die Zuverlässigkeit einer Cluster-Lösung wie der Exadata ist das Erkennen der Verfügbarkeit und einwandfreien Funktion aller beteiligten Clusterknoten essenziell. Diese Funktion ist auch über RoCE sichergestellt, indem sogenannte Heartbeat-Meldungen zu den Knoten der Exadata geschickt werden. Normalerweise wird ein relativ langer Timeout benötigt, um nicht fälschlicherweise einen Knoten vom Cluster auszuschließen. Durch die Nutzung von RDMA kann ein Serverfehler sehr sicher und schnell festgestellt werden. Da RDMA direkt auf die

Hardware zugreift, erfolgt die Antwort des entfernten Ports sehr schnell, auch wenn die Software überlastet ist. Es werden vier RDMA Readrequests an den verdächtigen Server geschickt, wobei alle vier Kombinationen von Quell- und Zielports abgedeckt werden. Wenn alle vier RDMA Readrequests fehlschlagen, wird der Knoten vom Cluster ausgeschlossen. Der gesamte Vorgang geschieht vollständig automatisch und transparent.

Innerhalb des internen RoCE-Netzwerks wird durch geschickte Priorisierung der Netzwerk-Nachrichten dafür gesorgt, dass Latenz-sensitive Prozesse nicht durch Operationen mit hohen Bandbreitenanforderungen beeinträchtigt werden. Latenz-sensitive Prozesse sind zum Beispiel Commit-Operationen oder Cache Fusion, hoher Durchsatz wird unter anderem für Backups, Reporting oder Batch-Prozesse benötigt. **RoCE Class of Service (CoS)** ermöglichen es, dass jedes Netzwerkpaket auf mehreren Classes of Service mit getrennten Netzwerk-Puffern gesendet werden kann. Exadata wählt automatisch und transparent den besten Class of Service für jede Art von Nachricht von der Oracle-Datenbank.

Intel Optane Persistent Memory

Das „RDMA over Converged Ethernet (RoCE)“-Protokoll ist nur die eine Hälfte

der Medaille, wenn es um die gesteigerte Performance der neuen Exadata-Modelle geht. Ein weiterer integraler Bestandteil sind die **Intel® Optane™ DC Persistent Memory Module (PMEM)**, die als neue Speicherschicht in den Exadata-Storage-Servern genutzt werden. Die neue Speichertechnologie bietet eine bessere Leseperformance als Flash, und Schreiboperationen überstehen Stromausfälle im Gegensatz zu DRAM. Preislich liegen die neuen Speichermodule zwischen DRAM und Flash.

Herkömmlich angebunden, würden die PMEM-Module außer der Datenpersistenz kaum einen nennenswerten Performancevorteil bringen. In der **Exadata Database Machine X8M-2** sorgen ausgefeilte Optimierungen dafür, dass das Performancepotenzial des PMEM ausgeschöpft wird und die Integrität der Daten auf PMEM bei Ausfällen erhalten bleibt:

- RDMA (Direktzugriff auf den PMEM unter Umgehung von CPU und OS)
- Aufruf spezieller Anweisungen zum Flush von Daten aus dem CPU-Cache in das PMEM
- Vervollständigung oder Rückgängigmachung der Schreibvorgänge, die durch einen Crash unterbrochen wurden

Zum Vergleich: Eine Exadata X8 liefert für 8k-Leseoperationen auf Flash eine End-

to-End Latenz von ca. 200µs. Würde man eine Exadata X8M mit PMEM und eine Datenbankversion älter als Oracle 19c verwenden, würde die End-to-End-Latenz schon auf ca. 100µs sinken. Wenn man nun noch zusätzlich den Kontextswitch zwischen Datenbank-Software und Kernel/OS auf dem Datenbank-Server und Kernel/OS und der Storage-Server-Software auf dem Storage-Server mithilfe von RDMA vermeidet, so erhält man eine End-to-End-Latenz von weniger als 19µs! (siehe Abbildung 4)

Mit persistentem Memory und RDMA-Zugriffen erreicht die Exadata X8M im Vergleich zur Exadata X8 also eine Verbesserung der Latenz für die interne Kommunikation um Faktor 10 und erreicht so eine 2,5-fach verbesserte OLTP-Performance.

Der PMEM wird vollkommen transparent für den Anwender vor dem Flash Memory in die Speicherhierarchie integriert und angesprochen (siehe Abbildung 5). Dadurch verbessert sich das für „hot Data“ insgesamt nutzbare Datenvolumen um Faktor 10. Eine automatische Spiegelung des PMEM zwischen Storage-Servern sorgt für die notwendige Fehlertoleranz. Der Anwender braucht hierfür nichts zu tun!

Die für OLTP-Systeme wichtige geringe Latenz von Log Writes wird durch den **Exadata X8M Persistent Memory Commit Accelerator** sichergestellt. Schnelle

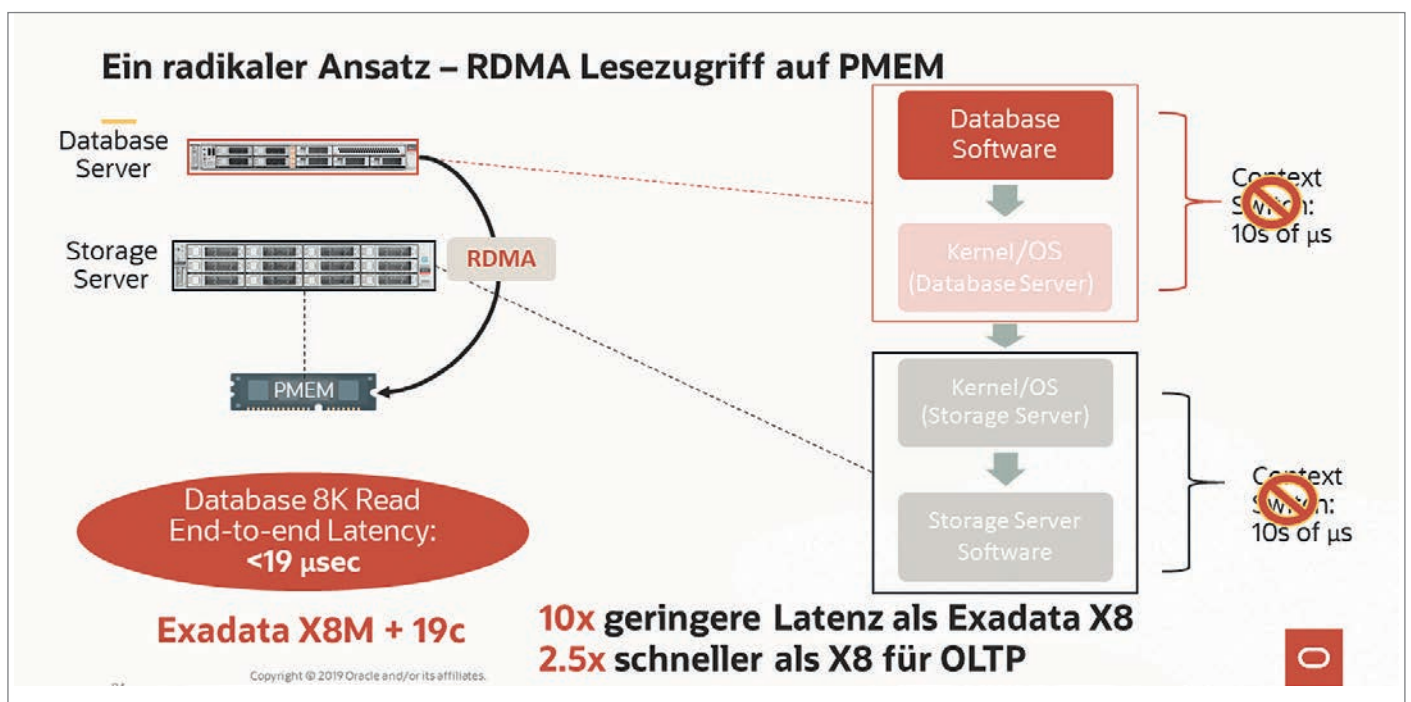


Abbildung 4: RDMA-Leseoperation auf persistentem Memory (Quelle: © Oracle)

Log Writes bedeuten schnellere Ausführung von Commit-Operationen, jede Verlangsamung des Log-Writer-Prozesses bedeutet, dass darunter die gesamte Datenbankperformance leidet. Unter Umgehung von Netzwerk und I/O-Software, Interrupts, Context Switches etc. schickt die Datenbank RDMA-Schreiboperationen an den PMEM mehrerer Storage-Server. Diese Schreiboperationen werden zeitverzögert auf Flash oder Disk heruntergeschrieben und beeinträchtigen somit nicht mehr die System-Performance.

KVM – die neue Virtualisierung

Bislang wurde auf den Zwei-Sockel-Modellen der Exadata die Virtualisierung mittels **Oracle VM (OVM)** umgesetzt, die auf dem Xen Hypervisor basiert. Da OVM das RoCE-Protokoll und PMEM nicht unterstützt, kommt auf der Exadata X8M-2 mit der bewährten KVM-Virtualisierung eine neue Lösung zum Einsatz, die viele Anwender bereits von der Oracle Database Appliance kennen dürften. An der Motivation für den virtualisierten Betrieb von Exadata hat sich mit der neuen Maschine nichts geändert, der primäre Einsatz für Virtualisierung auf der Exadata bleibt die Optimierung für Lizenzkosten, da eine vir-

tualisierte Exadata als Hard-Partitioning anerkannt wird. Das Setup erfolgt mithilfe des Oracle Exadata Deployment Assistant (OEDA) in der aktuellen Version. Die Spezifikationen unter KVM-Virtualisierung haben sich leicht verändert, wie *Abbildung 6* zeigt.

Für den KVM Host muss man 24GB zuzüglich 6% des gesamten Memory reservieren, dieser Bereich ist für Gast-VMs nicht nutzbar. In Summe macht das bei einem Datenbank-Server mit vollständigem Memory-Ausbau immerhin 1390GB, die für Gast-VMs bereitgestellt werden können. Es empfiehlt sich also, bereits während der Beschaffung die Datenbank-Server mit voll aufgerüstetem Memory zu planen, wenn man große virtuelle Umgebungen betreiben möchte. Der KVM Host benötigt 2 CPU-Kerne für den Betrieb, somit stehen maximal 46 Kerne pro Datenbank-Server für die Gast-VMs zur Verfügung. Während Prozessor-Kerne beziehungsweise vCPUs online den unterschiedlichen VMs zugeordnet werden können, ist das für die Umverteilung von Speicher online nicht möglich.

Die Exadata-X8M-2-Datenbank-Server beinhalten vier 1,2TB lokale Festplatten, die als RAID-5 konfiguriert sind. Es stehen standardmäßig 141GB pro KVM-Gast zur Verfügung, insgesamt sind das für alle Gast-VMs insgesamt 1,46TB, die

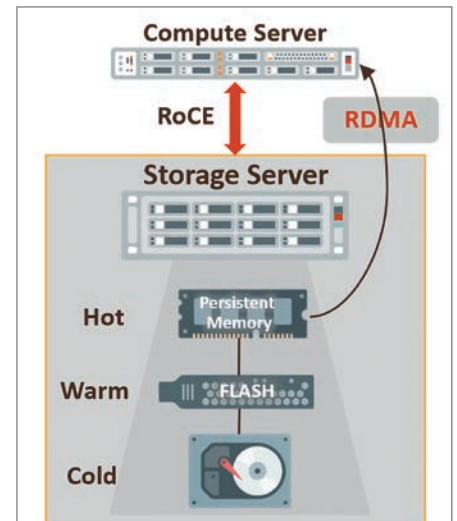


Abbildung 5: Speicherhierarchie mit PMEM (Quelle: Frank Schneede)

online auf bis zu 3,15TB erweitert werden können.

Mit der KVM-Virtualisierung ist es möglich, zwölf Gast-VMs pro Server zu betreiben, während es auf der Exadata X8-2 mit OVM nur maximal acht Gast-VMs waren. Neben der Änderung in der Terminologie (kvmhost statt dom0, guest statt domU) ergeben sich einige weitere kleine Änderungen im Detail durch das unterschiedliche Handling von KVM zu OVM. So erfolgt das Patching für KVM Host und Gast-VMs über ein einziges Yum Repository, das

Exadata X8M-2: KVM Spezifikationen

Category		X8M-2
VMs	Max KVM guests per database server	12
	Physical per database server (default/max)	384 GB / 1.5 TB
Memory	Min per KVM guest	16 GB
	Max per KVM guest	1390 GB
CPU/vCPU	Cores/vCPU* per database server	48/96
	Min cores/vCPU per KVM guest	2/4
	Max cores/vCPU per KVM guest	46/92
Disk	Total usable disk per database server for all KVM guests	3.15 TB
	Used disk per KVM guest at deployment	141 GB

Copyright © 2019 Oracle and/or its affiliates.

*1 core = 1 OCPU = 2 hyper-threads = 2 vCPUs



Abbildung 6: Spezifikation der KVM-Virtualisierung (Quelle: © Oracle)

VM-Management erfolgt über `vm_maker` und das `OEDACLI`, das Zurückfordern von Festplattenplatz durch `reclaimdisk.sh` während der Installation, ist unter KVM nicht notwendig.

Softwarevoraussetzungen

Die Exadata X8-2 erfordert mindestens eine Exadata-Storage-Server-Software Version 19.2 oder neuer, die Exadata X8M-2 die Version 19.3 oder neuer.

Auf den aktuellen Exadata-Modellen werden selbstverständlich alle aktuellen Releases der Oracle-Datenbank unterstützt, beginnend mit Oracle 11.2.0.4.0.180717. Ältere Versionen (11.2.0.3.28) sollten ohnehin, auch wenn das auf der Exadata X8-2 technisch noch möglich ist, nicht mehr verwendet werden – die genannte Version wird auf der Exadata X8M-2 nicht mehr unterstützt. Der volle Umfang der neuen Funktionen zur Performancesteigerung (PMEM Cache und PMEM Log) auf der Exadata Database Machine X8M-2 stehen jedoch erst ab Oracle-Datenbank 19.4.0.0.0.190716 zur Verfügung.

Aufrüstmöglichkeiten und Interoperabilität

An der Systematik der Aufrüstmöglichkeiten hat sich im Vergleich zum Modell der Exadata X8-2 nichts geändert. Die Exadata Database Machine X8M-2 High Capacity (HC) Eighth Rack ist nach wie vor als Einstiegsmodell mit einer eingeschränkten Aufrüstmöglichkeit im Storage einzuordnen. Als Test- und Entwicklungssystem ist es uneingeschränkt zu empfehlen, zumal mit 1,5TB Persistent Memory die volle Kapazität der neuen Speicherschicht genutzt werden kann. Wenn der Kunde jedoch bereits in der Planungsphase einer neuen Exadata ein starkes Wachstum des Datenvolumens prognostizieren kann, so ist eine Exadata Database Machine X8M-2 High Capacity (HC) Quarter Rack mit einer bedarfsgerechten Beschränkung der genutzten CPUs (Capacity on Demand, CoD) eine zukunftsichere Wahl.

Wegen des unterschiedlichen internen Netzwerks ist ein Multi-Rack-Betrieb von Exadata X8 und X8M nicht möglich. Auch eine Erweiterung eines bestehenden InfiniBand-basierten Modells mit Exadata-

X8M-2-Storage-Servern ist nicht möglich. Trotzdem ist der schrittweise Umstieg auf das neue Exadata-X8M-2-Modell durchaus sinnvoll, denn man kann die neue, leistungsfähigere Maschine als Produktionssystem nutzen, während die alte Maschine als Standby-System über Data Guard oder Active Data Guard angebunden ist.

Für die aktuellen Exadata-Modelle X8-2 ist kein Upgrade auf X8M-2 vorgesehen. Die aktuellen InfiniBand-basierten Modelle werden auch in Zukunft weiter unterstützt, die Erweiterbarkeit und Ausbaufähigkeit ist weiterhin gegeben.

Während ein Multiracking von Exadata-X8M-2-Systemen über 100GbE und RoCE-Protokoll sichergestellt ist, gibt es zum jetzigen Zeitpunkt noch Fragen von Kunden, die „Backup & Recovery“-Lösungen (zum Beispiel ZFS Storage Appliance oder die ZDLRA) bislang über InfiniBand an ihre Exadata angebunden haben. Ein Anschluss von ZFSSA oder ZDLRA über 100GbE und RoCE-Protokoll ist zurzeit nicht unterstützt. Die Empfehlung lautet daher, „Backup & Recovery“-Systeme über die 10/25GbE-Adapter der Datenbank-Server und ein separiertes Backup-Netzwerk anzubinden. Durch die Nutzung der 25GbE-Ports und gleichzeitiges Bonding lassen sich die notwendigen Geschwindigkeiten für Backup und Recovery in der Regel erreichen.

Fazit

Um die eingangs gestellte Frage zu beantworten: Ja, ich persönlich halte die neue Exadata Database Machine X8M-2 für „The Next Big Thing“! Die neue Exadata-Generation bietet den Kunden mit dem neuen RoCE-Protokoll und persistentem Memory eine noch nie dagewesene Leistung und somit viele neue Möglichkeiten für den Hochleistungsbetrieb jeder Art von Oracle-Datenbanken. Mit der KVM-Virtualisierung erhöht sich zudem die Konsolidierungsdichte auf der Exadata. Gleichzeitig bietet der Umstieg auf das interne 100GbE-Netzwerk und RoCE-Protokoll genügend Sicherheit, da umfangreiche Teile des bewährten InfiniBand-Software-Stacks weiterhin genutzt werden. Die Investitionssicherheit für Exadata-Bestandskunden ist durch die auch künftig zugesicherte Unterstützung von InfiniBand-Komponenten gegeben. Ein

schrittweiser Umstieg auf die neue Architektur ist trotzdem sinnvoll und möglich, da eine MAA-Architektur zwischen älteren Exadata-Generationen und der Exadata X8M über Data Guard oder Active Data Guard aufgebaut werden kann.

Weiterführende Informationen

- <https://www.oracle.com/engineered-systems/exadata/database-machine/>
- <https://www.oracle.com/a/ocom/docs/engineered-systems/exadata/exadata-x8m-2-ds.pdf>
- <http://www.oracle.com/technetwork/database/exadata/oeca-download-2817713.html>
- <https://www.oracle.com/database/technologies/oeda-downloads.html>



Frank Schneede
Frank.Schneede@oracle.com



Hilfreiche Tools für den täglichen Exadata-Betrieb (Teil 1)

Michael Schulze, Opitz Consulting

Wer sich im Zuge der Konsolidierung seiner Oracle-Datenbanken für eine Exadata-Datenbanklösung entscheidet und diese in den Betrieb bringt, weiß, wie sich das auf die gesamte Umgebung auswirkt. Sie wächst enorm in kurzer Zeit und wird intensiv genutzt. Nicht selten müssen Administratoren mehr als 100 Datenbankinstanzen versorgen, verteilt auf verschiedene Systembereiche, wie Test, Stage und Prod, mit den unterschiedlichsten Verfügbarkeitsanforderungen. Für die Verfügbarkeit kommt Oracle Data Guard zum Einsatz sowie RAC und zusätzliche Standby-Lösungen an einem zweiten Standort. In diesem komplexen Kontext sind passende Analyse-Tools eine Grundvoraussetzung, um den Überblick zu behalten und einen sicheren und reibungslosen Datenbankbetrieb zu gewährleisten. Diese Tools liefern eine strukturierte Dokumentation von Einzelkomponenten und Systemständen und helfen bei Patch-Unterstützung, Housekeeping, Monitoring und vielem mehr. In dieser zweiteiligen Artikelserie stelle ich Ihnen einige Tools vor, die sich bei der Unterstützung des Exadata-Betriebs in unserem Kundenumfeld bewährt haben. Darunter finden sich nicht nur Oracle-Werkzeuge für die Betriebsunterstützung, sondern auch einige eigene oder frei verfügbare Skripte. Im diesem ersten Teil der Artikelserie geht es um Tools für Dokumentation und Patching. Der zweite Teil behandelt Tools für Housekeeping, Monitoring sowie Analysewerkzeuge und erscheint in einer der nächsten Ausgaben.

Dokumentationsunterstützung mit dem Remote Diagnostic Agent (RDA)

Den Remote Diagnostic Agent (RDA) stellt Oracle für den Support verschiedener Plattformen zur Verfügung (MOS Doc ID 314422.1). Der RDA bietet eine gute Dokumentationsgrundlage zur schnellen Erfassung von System- und Datenbankinformationen. Eigentlich ist der RDA aber weitaus mehr: Er deckt mit seinen enthaltenen Profilen einen großen Teil des Oracle-Produktportfolios ab und ermöglicht so einen Einblick in den Aufbau und die Strukturen von größeren Oracle-Umgebungen:

- Welche Komponenten werden verwendet?
- Wie sind diese konfiguriert?
- Welche Abhängigkeiten gibt es unter ihnen?

Uns geht es hier vor allem um die Dokumentationsaspekte im Exadata-Datenbankumfeld. Im Folgenden dazu einige Beispiele aus der Praxis:

IST-Aufnahme in vier Schritten

Bei der IST-Aufnahme durch den RDA sind vier Schritte erforderlich:

1. Bereitstellung der relevanten RDA-Version (z. B. Linux-x64)
2. Konfigurierung des RDA für die jeweilige Umgebung oder Datenbank
3. Generierung der Daten für die Auswertung
4. Auswertung des Reports im Browser

Parameter	Beschreibung
-L	profile list
-S	rda setup
-p <profile>	define profile
-v	verbose mode
-q	quiet mode
-CR	create report
-P	pack as zip
-s <config>	define configfile

Tabelle 1: Parameter des RDA, die in den Beispielen genutzt wurden

Tabelle 1 zeigt die Parameter, die der RDA bei den Auswertungen in unseren Beispielen benutzt.

RDA-Profile

Mit dem Befehl `rda.sh -L` kann die komplette Profilliste ermittelt werden. Hier stehen, wie schon oben erwähnt, viele Oracle-Produkte und -Versionen zur Verfügung. Bei einer Oracle-Datenbank wären die relevanten Profile DB11g und DB12c.

RDA-Konfiguration

Nach der Bereitstellung der RDA-Software muss eine Konfigurationsdatei erstellt werden. Hierfür wird zunächst ein RDA-Profil gewählt, zum Beispiel DB12c. Dann wird ein Name für die Konfiguration festgelegt. Wird kein Name definiert, erzeugt der RDA einen Default-Namen, wie hier im Beispiel gezeigt wird (siehe Listing 1):

Es folgt eine interaktive Abfrage dieser und weiterer Daten wie:

- Oracle Home Directory
- ORACLE_SID
- Oracle User ID

Ergebnis ist eine RDA-Konfiguration. Auf deren Grundlage kann jetzt der RDA-Report generiert werden. Dabei wird durch Angabe von Parametern gesteuert, wie sich die Generierung zur Laufzeit verhält (siehe Beispiel in Listing 2)

Der RDA-Report besteht aus vielen statischen HTML-Dateien. Diese Dateien befinden sich in einer zur Laufzeit erstellten Verzeichnisstruktur. In ihnen finden sich unzählige Informationen zu System, Konfiguration etc. sowie Details zum spezifischen Produkt, in unserem Fall zu Oracle-Datenbank, Konfiguration, Status etc. Die gesammelten Report-Informationen bil-

```
rda.sh -S -p DB12c -s db001.cfg.
```

Listing 1

```
rda.sh -qCRP -s db001.cfg
```

Listing 2

den eine wichtige Grundlage für die Beurteilung von IST-Zuständen.

Ausfernde Exadata-Landschaften

Wie in der Einleitung schon dargestellt, kann eine Exadata-Cluster-Umgebung aus mehreren Teilsystemen bestehen. Zum Beispiel wird sie oft getrennt nach Testing, Staging und Production. Je nach Ausbaustufe, also „full“, „quarter“, „half“, „rack“, können diese Exadata-Einzelsysteme mehrere Database Nodes beinhalten. Im RAC-Cluster kann man Multi-Instance-Datenbanken betreiben; die Anzahl der Instanzen erhöht sich zusätzlich, wenn zur Ausfallsicherung eine Data-Guard-Konfiguration als Primary- oder Standby-Lösung verwendet wird. So haben Admins in unseren Kundenumfeldern schon mal mehr als 100 Oracle-Datenbankinstanzen zu verwalten. Die Abbildung 1 soll das in einer 2-Node-Konfiguration noch einmal veranschaulichen:

Eine Zentrale für die Dokumentationsverwaltung

In komplexen Exadata-Umgebungen ist eine saubere Dokumentation der kompletten Datenbankumgebung also zwingend notwendig, um im Fehlerfall schnell reagieren zu können und eine zentrale Übersicht über alle relevanten Daten

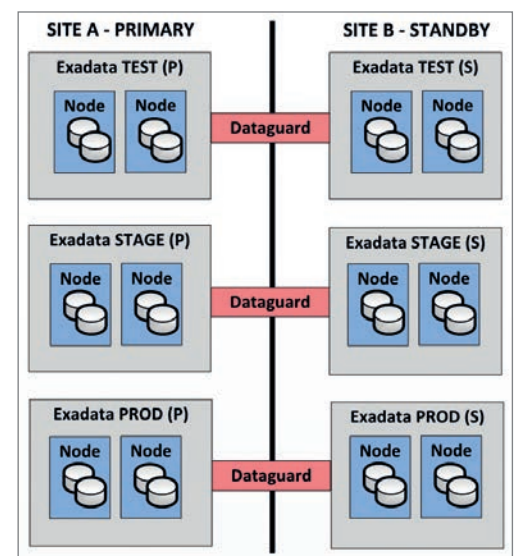


Abbildung 1: Multi-Instance-Datenbanken in der Exadata (Quelle: Michael Schulze)

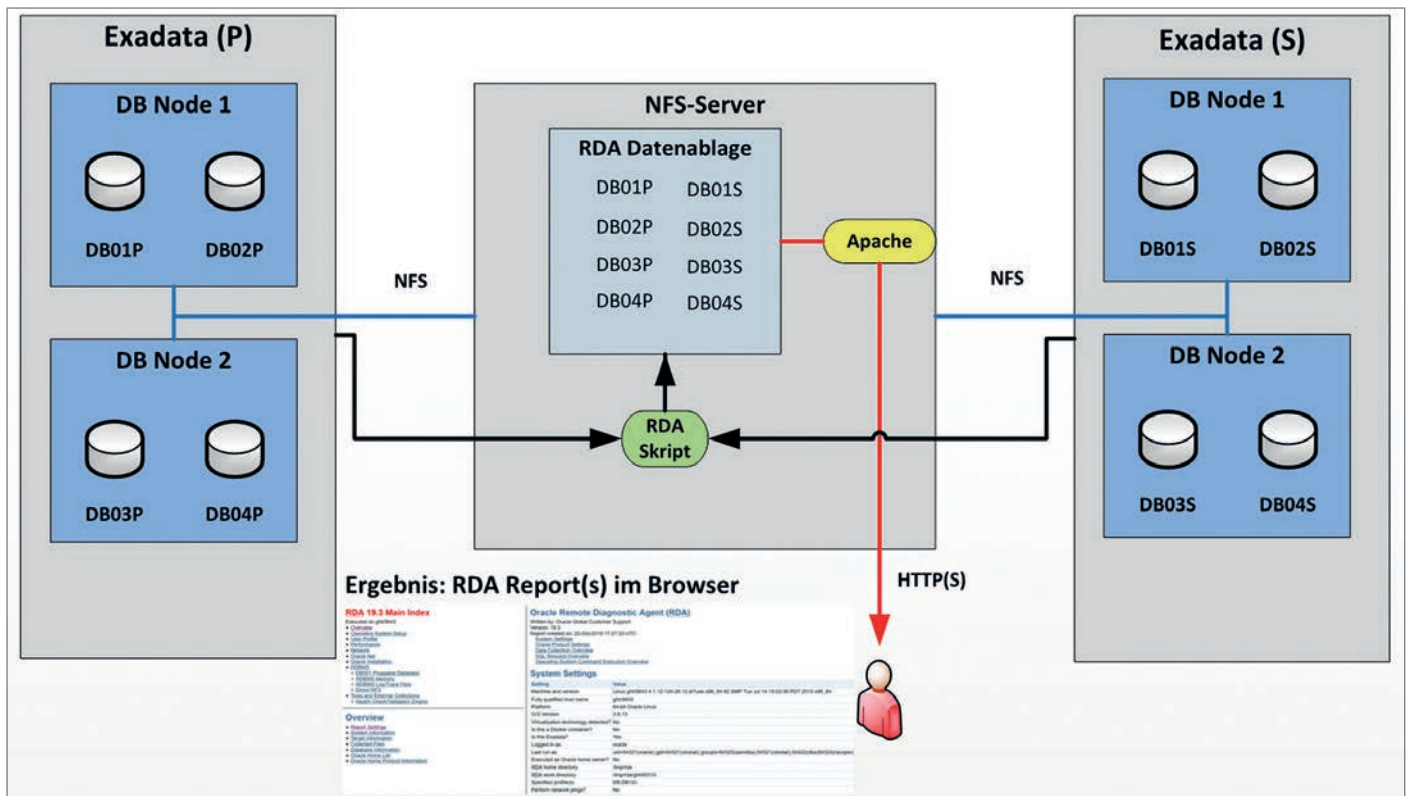


Abbildung 2: Prinzip des RDA-Repository (Quelle: Michael Schulze)

herzustellen. Die Verwendung des RDA als generischer Ansatz versetzt die Verantwortlichen in die Lage, diese Informationen schnell und ohne Impact zu er-

mitteln, und bietet sich hier förmlich an. Auch die statischen HTML-Dateien der RDA-Reports sind ideal für einen zentralen Zugriff.

Es ergibt also Sinn, RDA-Reports in regelmäßigen Zyklen für alle Exadata-Datenbankinstanzen zu sammeln und zentral auf einer an die Exadata angebotenen Network-File-System-(NFS-)Ressource zusammenzufassen. Warum externe NFS-Ressourcen einbeziehen? Da ein Exadata-System immer höheren Sicherheitsanforderungen unterworfen ist – schließlich befinden sich hier oft unternehmenskritische Daten –, wären Dateien auf diesem System nur für einen beschränkten Personenkreis zugänglich.

Praktisch ist das kein Problem: Die Daten können durch geeignete Mechanismen wie Skripts oder Cron-Steuerung auf die NFS-Systeme geschrieben werden. Diese sind geringeren Anforderungen unterworfen und können diese Daten damit optimal hosten. Die statische Datensammlung kann dann zentral beispielsweise über einen Webserver zur Verfügung gestellt werden. Es ist also kein direkter Zugriff auf Exadata und Datenbank nötig, um einen Überblick über Systeme und Datenbanken zu bekommen.

Die automatisch gesammelten RDA-Daten können so als generische Dokumentationsgrundlage genutzt werden, die die Arbeit der zuständigen Mitarbeiter und Administratoren unterstützt und vereinfacht. Die *Abbildung 2* soll das Prinzip des RDA-Repository illustrieren.

```

=====
12.11.2019 11:40:01: Batch RDA-Report Generierung startet
=====
ghtr5643: 12.11.2019 11:40:01: run rda for: ghtr5643/ght4531
ghtr5643: 12.11.2019 11:42:09: run rda for: ghtr5643/ssder41
ghtr5643: 12.11.2019 11:44:20: run rda for: ghtr5643/jzt5561
ghtr5643: 12.11.2019 11:46:20: run rda for: ghtr5643/juz4531
ghtr5643: 12.11.2019 11:48:30: run rda for: ghtr5643/lop6751
ghtr5643: 12.11.2019 11:50:58: run rda for: ghtr5643/loi7661
    
```

Listing 3: Automatische RDA-Generierung (alle Datenbankinstanzen des Node)

```

=====
RDA-Generator
=====
[1] - ghtr5643/ght4531
[2] - ghtr5643/ssder41
[3] - ghtr5643/jzt5561
[4] - ghtr5643/juz4531
[5] - ghtr5643/lop6751
[6] - ghtr5643/loi7661

Ihre Wahl: 1
30.10.2019 15:13:36: run rda for: ghtr5643/ght4531

real 5m2.565s
user 3m48.314s
sys 0m38.211s
    
```

Listing 4: Manuelle Generierung (Auswahl einer Datenbankinstanz)

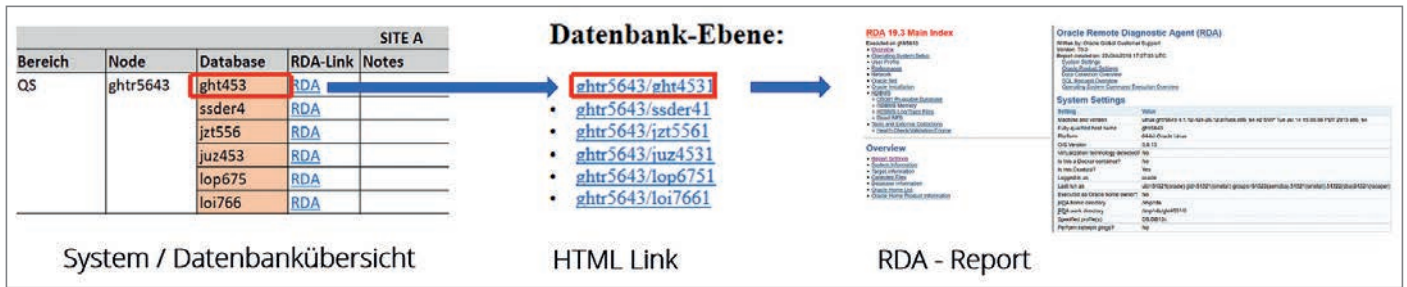


Abbildung 3: Zentraler Überblick über die RDA-Dokumentationen (Quelle: Michael Schulze)

Wir haben das zentrale RDA-Repository als Skriptlösung bei einem Kunden realisiert und im Einsatz. Die Lösung ermöglicht die automatische und manuelle Sammlung von RDA-Reports der laufenden Datenbanken einer Exadata sowie die manuelle Integration neuer Datenbanken in RDA. Zur Laufzeit werden automatisch übergeordnete HTML-Strukturen erstellt, über die auf alle RDA-Reports zugegriffen werden kann.

Die folgenden anonymisierten Skriptbeispiele (siehe Listing 3, 4 und 5) gewähren einen Blick in die Funktionsweise der RDA-Generierung. Wie sich das Ergebnis am Ende darstellt, zeigt Abbildung 3.

Tools zur Patching-Nachbereitung

Das Thema Oracle Database Patching spielt bei Exadata-Cluster-Systemen eine große Rolle. Im Normalfall wird die bestehende Oracle-Database-Software zyklisch durch Patches aktualisiert. In diesem Zusammenhang unterscheidet man Software Patches im Dateisystem und die zugehörigen Data Patches in der Datenbank.

Der Patching- und Verteilprozess auf der Exadata

Für das initiale Patchen der Oracle-Database-Software nutzt man am besten eine eigene Patchumgebung (separates Oracle_Home mit angeschlossener Test-Datenbank). Nach Aktualisieren der Patchumgebung (neuer Patchstand) und initialen Tests kann dieses Oracle_Home nun auf weitere Exadata Nodes verteilt und in die jeweilige Umgebung integriert werden.

Das Verteilen erfolgt mit OS-Mitteln (tar, scp). Die Integration des verteilten Oracle_Home ins System erfolgt mittels Cloning (Perl Skript: clone.pl) (siehe Listing 6).

Abbildung 4 gibt einen Überblick über den Patchvorgang, das Verteilen und das Cloning eines Oracle_Home in der Exadata-Umgebung.

Cloning-Prozess und Nachbereitung

Nach dem Cloning-Vorgang sind noch einige Dinge zu beachten und nachzubereiten: Beim Cloning über clone.pl wird das verteilt-gepatchte Oracle_Home in

die Cluster-Umgebung integriert. Dafür durchläuft das Skript clone.pl verschiedene Stadien, unter anderem einen Re-compile-Prozess, die Registrierung ins oralInventory und einige mehr. Bedingt durch den Recompile-Prozess können hier wichtige Eigenschaften und Optionen verloren gehen. So wie diese drei, die in der Exadata-Umgebung dringend notwendig sind:

- Real Application Cluster (RAC)-Option
- Reliable Datagram Sockets (RDS) für die RAC-Kommunikation über Infini-Band
- Direct Network File System (DNFS) beispielsweise als Grundlage der Anbindung an die Dateisysteme ZFS oder NFS für Backups, Oracle Data Pump etc.

Aus diesem Grund ist die präventive Kontrolle und gegebenenfalls Korrektur als Vorbereitung zur Datenbankmigration in das neue Oracle_Home eine wichtige Voraussetzung und im Vorfeld zwingend erforderlich. Probleme im Migrationsprozess wären sonst vorprogrammiert!

Bei der präventiven Kontrolle wird sichergestellt, dass

- alle relevanten Features und Optionen, z. B. RAC, RDS oder DNFS, genutzt werden können und in der Software enabled sind,
- die Rechte des Oracle Binary für die betreffenden User und Gruppen richtig gesetzt sind,

```

=====
RDA-Generator
=====
[1] - kliz4530/sder44
[2] - kliz4530/bght55

Ihre Wahl: 1
30.10.2019 13:24:39: rda config for: kliz4530/sder44 started

-----
RDA.BEGIN: Initializes the Data Collection
-----

Enter the Oracle home to be used for data analysis
> /u01/app/oracle/product/12.2.0.1/dbhome_1
. . .

```

Listing 5: Integration neuer Datenbank in RDA (Config)

```

Beispiel Syntax (12c):
$ORACLE_HOME/clone/bin/clone.pl \
ORACLE_BASE="<>PATH">" \
ORACLE_HOME="<>PATH">" \
'-O"CLUSTER_NODES={<NODELIST>}'" \
'-O"LOCAL_NODE=<NODE>"'

```

Listing 6

- Network Files wie tnsnames.ora, sqlnet.ora etc., beziehungsweise die Links dafür, vorhanden sind.

In größeren Exadata-Umgebungen wird ein Oracle_Home-Patchstand auf eine Vielzahl von Teilsystemen ausgerollt. Um eine zeitaufwendige manuelle Kontrolle der genannten Voraussetzungen zu vermeiden, nutzen wir mittlerweile ein eigenes Skript, das in diesem Kontext gut unterstützen

kann. Mithilfe von farblichen Hervorhebungen stellt es im Terminal sehr intuitiv dar, welche Dinge nach dem Cloning-Prozess fehlen. Mit der Möglichkeit, das Skript auf der Exadata mittels Distributed Command Line Interface (dcli utility) zu nutzen und somit alle im Cluster befindlichen DB-Nodes parallel abzufragen, bietet es eine zentrale Übersicht über alle Nodes.

Abbildung 5 zeigt den Skript-Output in einer Exadata-Umgebung.

Der Screenshot in *Abbildung 4* zeigt, dass hier in einigen Oracle_Homes das DNFS-Feature fehlt, auch verschiedene Network-Dateien sind nicht präsent und daher rot gekennzeichnet.

Ein möglicherweise fehlendes Feature kann nachträglich durch folgende Befehle enabled werden (*siehe Listing 7*). Nach Kontrolle und Korrektur des entsprechend vorbereiteten Oracle_Home erfolgt als letzter Schritt der Umzug der Datenbanken.

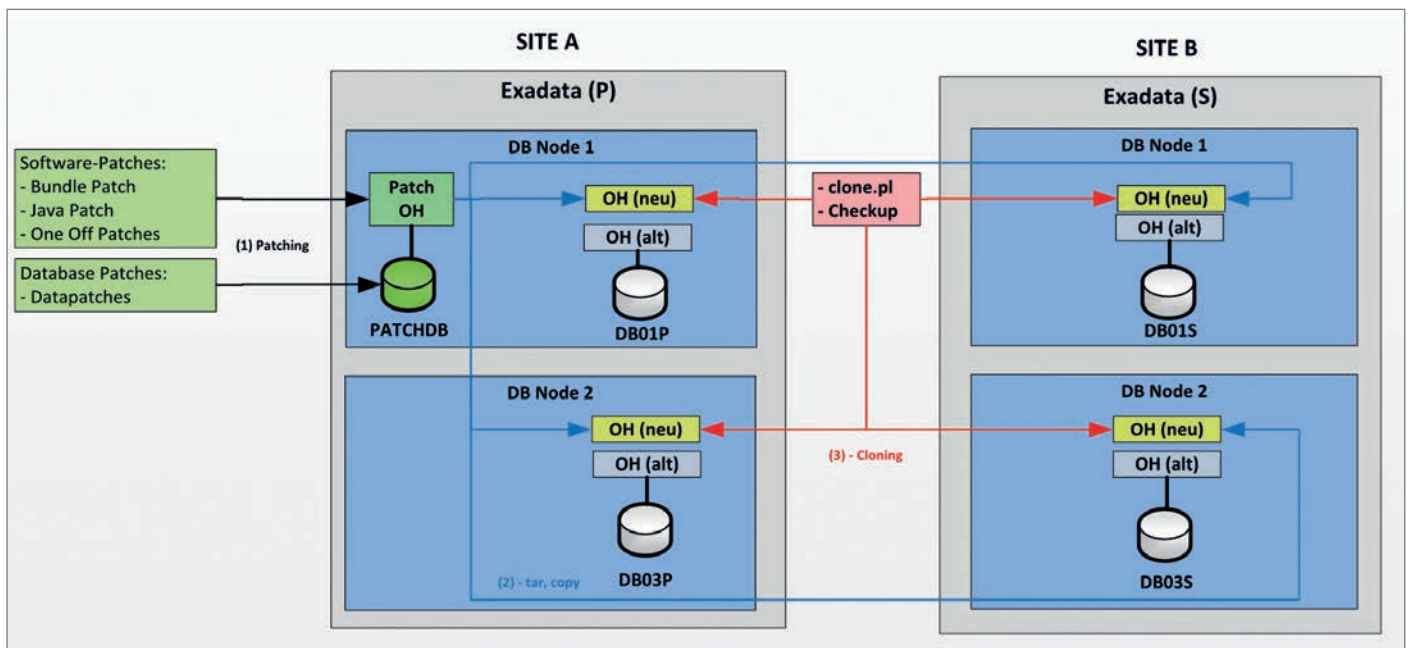


Abbildung 4: Der Patching- und Cloning-Prozess (Quelle: Michael Schulze)

```

=====
# check Overview (12.2): fjjgh765
=====
/u01/app/oracle/product/12.2.0.1_1/dbhome_1 : -rwsr-s--x oracle:asmadmin RDS RAC /  SOLNETORA TNSNAMES LDAPORA
/u01/app/oracle/product/12.2.0.1/dbhome_1 : -rwsr-s--x oracle:oinstall RDS RAC DNFS /  SQLNETORA TNSNAMES LDAPORA

=====
# check Overview (19.4): fjjgh765
=====
/u01/app/oracle/product/19.4.0.0_1/dbhome_1 : -rwsr-s--x oracle:asmadmin RDS RAC /  SOLNETORA TNSNAMES LDAPORA

=====
# check Overview (11.2): kliz4530
=====
/u01/app/oracle/product/11.2.0.4_1/dbhome_1 : -rwsr-s--x oracle:asmadmin RDS RAC DNFS /  SQLNETORA TNSNAMES LDAPORA
/u01/app/oracle/product/11.2.0.4/dbhome_1 : -rwsr-s--x oracle:oinstall RDS RAC DNFS /  SQLNETORA TNSNAMES LDAPORA

=====
# check Overview (12.1): kliz4530
=====
/u01/app/oracle/product/12.1.0.2_1/dbhome_1 : -rwsr-s--x oracle:asmadmin RDS RAC DNFS /  SQLNETORA TNSNAMES LDAPORA
/u01/app/oracle/product/12.1.0.2/dbhome_1 : -rwsr-s--x oracle:oinstall RDS RAC /  SQLNETORA TNSNAMES LDAPORA

=====
# check Overview (12.2): kliz4530
=====
/u01/app/oracle/product/12.2.0.1_1/dbhome_1 : -rwsr-s--x oracle:asmadmin RDS RAC DNFS /  SQLNETORA TNSNAMES LDAPORA
/u01/app/oracle/product/12.2.0.1/dbhome_1 : -rwsr-s--x oracle:asmadmin RDS RAC DNFS /  SQLNETORA TNSNAMES LDAPORA

=====
# check Overview (19.4): kliz4530
=====
/u01/app/oracle/product/19.4.0.0_1/dbhome_1 : -rwsr-s--x oracle:asmadmin RDS RAC /  SOLNETORA TNSNAMES LDAPORA
    
```

Abbildung 5: Anonymisierter Skript-Output einer Exadata-Umgebung (Quelle: Michael Schulze)

Diesen Prozess möchte ich hier nur grob umreißen. Für ihn wären die folgenden Teilaktionen erforderlich:

- Umzug der Datenbanken nach Oracle_Home (neuer Patchstand)
- Versorgung der umgezogenen Datenbanken mit den passenden neuen Data Patches
- Patchstand testen und verifizieren in der Clusterumgebung
- Prüfen der umgezogenen Datenbanken und Freigabe

Fazit

Auch in komplexen Exadata-Umgebungen kann man mit geeigneten Hilfsmitteln wie dem RDA den Überblick behalten. Zentral zur Verfügung gestellte RDA-Reports können dafür eine wertvolle Dokumentationsgrundlage schaffen. Das Patching im Exadata-Datenbankumfeld ist ein Prozess, der gut vorzubereiten, zu dokumentieren und zu kontrollieren ist. Die Oracle-Database-

```
# RAC Option
make -f ins_rdbms.mk rac_on
# RDS Option
make -f ins_rdbms.mk ipc_rds ioracle
# DNFS
make -f ins_rdbms.mk dnfs_on
```

Listing 7

Software hat einige spezifische Exadata-Abhängigkeiten zu Features wie RAC, RDS oder DNFS sowie Einstellungen, die zwingend notwendig sind. Eine Skriptunterstützung kann die notwendige Migrationsgrundlage dafür sicherstellen und Lücken aufdecken.

Ausblick Teil 2

Auch für Housekeeping, Monitoring und Analyse-Checks gibt es passende Werkzeuge, die die Verwaltung komplexer Exadata-Landschaften erleichtern und sicherer machen. Diese Tools stelle ich im nächsten Teil dieses Artikel-Zweiteilers näher vor. Hier ein kleiner Vorgeschmack:

- Housekeeping:
 - Zentrales Housekeeping zur zyklischen Bereinigung von nicht benötigten Daten auf Exadata
- Monitoring und Analyse-Unterstützung:
 - Clustermonitoring
 - RDA: Exadata Profile anwenden
 - AHF- und TFA-Funktionen
 - Exachk als Bestandteil von TFA



Michael Schulze
michael.schulze@opitz-consulting.com

Robotron-Schulungszentrum

ORACLE APPROVED EDUCATION CENTER

Kompetente Wissensvermittlung mit Durchführungsgarantie

Buchen Sie
jetzt Ihren
Kurs!

Praxis-Workshop Oracle Database Appliance

26. – 27.02.2020 Dresden
26. – 27.05.2020 Dresden
2 Tage geballtes Wissen für den
ODA Admin



PostgreSQL 12: Administration Workshop

10. – 12.03.2020 Dresden
16. – 18.06.2020 Dresden
3 Tage Intensivtraining zum
aktuellen PostgreSQL Release



Praxisworkshop Administration der Oracle Datenbank 19c Standard Edition 2 (SE2)

25. – 29.05.2020 Dresden
Alle wichtigen Informationen über das
neue Long Term Release Oracle 19c



Mehr Kurse finden Sie hier



Virtualized Exadata – die vier ersten produktiven Jahre...

Daniele Massimi, appliedtec

Ab der Image-Version 12.1.2.1.0 ermöglicht Oracle den Einsatz der Virtualisierung auf der Exadata. Dieses Novum gestattet neue Business Cases und kommt Kunden-Anforderungen entgegen, indem eine optimale Isolation der Server sowie eine nutzungsgerechte Lizenzoptimierung ermöglicht wird. In diesem Artikel werden die Herausforderungen sowie einige Knacknüsse der ersten vier „produktiven“ Jahre einer virtualized Exadata erläutert.

Zuallererst, wieso eigentlich die Exadata virtualisieren?

Die Exadata Database Machine ist nun seit bereits über 10 Jahren auf dem Markt. Anfänglich, zumindest ist dies meine Erfahrung, wurden die Maschinen für dedizierte Zwecke eingesetzt. Somit konnten dedizierte Probleme auf einen Schlag gelöst werden. Jedoch, für die damalige Zeit, zu einem relativ hohen Preis-Leistungs-Verhältnis.

Nur wenige Kunden nutzten zu Anfang die Exadata als Konsolidierungs-Plattform. Mit der Evolution dieser Maschine und den Erweiterungen in der Hardware sowie im Software-Bereich wurde die

Exadata nach und nach attraktiver und es entstand der Wunsch, diese auch als Konsolidierungs-Plattform einzusetzen. Mittlerweile hat sich der Trend um 180 Grad gewendet und es sind mehrheitlich Konsolidierungs-Projekte, die mit der Exadata realisiert werden.

So weit so gut, wieso brachte nun Oracle die Virtualisierung auf der Exadata? Wenn es rein um eine Partitionierung der Maschine geht, bietet Oracle bereits genügend „flexible“ Möglichkeiten wie Capacity on Demand oder dedizierte Cluster mit dedizierten Compute Nodes.

In den bisherigen Konsolidierungs-Projekten wurden die Datenbanken in einer linearen Form auf allen „Compute Nodes“,

soweit es die Ressourcen zuließen, verteilt. Der Haupt-Nachteil solcher Implementierungen besteht darin, dass keinerlei Möglichkeiten vorhanden sind, etwaige Umgebungen voneinander zu trennen.

Demzufolge ist einer der Hauptgründe, weshalb eine Exadata virtualisiert werden soll, „Isolation“.

Mögliche Use Cases, die Isolation einzusetzen, sind zum Beispiel Mandantenfähigkeit, Einhaltung von Sicherheitsrichtlinien, strikte Trennung zwischen produktiven und vor-produktiven Umgebungen, Anbindung von verschiedenen Netzwerken (VLANs) und deren Isolation.

Ein weiterer Grund für die Virtualisierung ist die Unterstützung von „Trusted

Partition“. Hiermit muss man nur noch die verwendeten Cores lizenzieren. Als kleinste Einheit gilt allerdings die Mindestmenge von 14 Cores per Compute Node (gültig für X6 und höher). Demnach kann mit dieser Gegebenheit eine Lizenzoptimierung herbeigeführt werden.

Wieso entschied man sich für eine virtualisierte Exadata?

In vielen Unternehmen steht nach einigen Jahren das Ende des Lebenszyklus von IT-Plattformen an. Das heißt in der Regel, dass man beginnt, sich auf dem IT-Markt umzuschauen, um bestmögliche Lösungen für seine Bedürfnisse zu finden. Auch in diesem Projekt war dies der Fall, hierbei ging es um die Ablösung einer Datenbank-Plattform.

Dabei werden Anforderungen daran definiert, was die zukünftige Plattform leisten sollte. Danach kann mit einer Evaluation von möglichen Lösungen gestartet werden.

Ziemlich ähnlich ging man hierbei vor und folgende Kriterien waren schlussendlich die groben Eckanforderungen:

- Realisierung einer Konsolidierungs-Plattform für Oracle-Datenbanken
- Ablösung der bestehenden Datenbank-Plattform
- Lizenz-Optimierung
- Bedienung von zwei Netz-Zonen (Intranet und DMZ)
- Isolation von Produktion zu Vor-Produktions-Umgebungen
- Dedizierte Kunden-Umgebungen
- Kompatibel zu DBaaS

Aufgrund der definierten Anforderungen und für die Erreichung des besten Resultates gemäß dem Kriterienkatalog entschied man sich für die Exadata – um genauer zu sein, für eine virtualisierte Exadata. Das Novum der virtualisierten Exadata war noch recht frisch, allerdings konnten die neuen Funktionalitäten sämtliche Anforderungen erfüllen.

Hier die Auflistung der Haupt-Eigenschaften, die die Entscheidung für die virtualisierte Exadata herbeigeführt haben:

- virtualized Exadata kann als Konsolidierungs-Plattform eingesetzt werden
- virtualized Exadata bietet Isolation mittels VMs



Abbildung 1: Aktuelle Konfiguration (Quelle: Daniele Massimi)

- virtualized Exadata bietet Lizenz-Optimierung mittels „Trusted Partition“
- virtualized Exadata hat die gleiche Funktionalität wie mit „Bare Metal“
- Exadata kann erweitert werden (z.B. mit Elastic Configuration)

Als „first mover“ geht man immer das Risiko ein, dass gewisse Funktionen noch nicht reif genug sind oder man in Bugs reinläuft. Wie jedoch erwünscht, hatte man gleich bei Projektbeginn die nötige Unterstützung seitens Oracle. Es wurden auch entsprechende „product enhancements“ eigens für dieses Projekt implementiert, wie zum Beispiel VLAN Tagging für Management Interface (MOS 2090345.1).

Definierte Ziel-Plattform

Nachdem das Produkt für die Ziel-Plattform bekannt war, ging es darum, das Setting für den Initial-Setup zu definieren.

Dabei wurden folgende Eckdaten erhoben:

- benutzte Compute Power (CPU)
- benutzte Memory für SGA und PGA

- benutzte Storage
- Anzahl der Datenbanken

Aufgrund der erhobenen Kennzahlen entschied man sich, mit zwei Quarter Racks zu starten. Zudem war schnell klar, dass, um eine höhere Konsolidierungs-Dichte zu erzielen, der Einsatz der damals neu erschienenen Multitenant-Option unerlässlich war.

Zusammen mit Oracle ACS wurden das Initial-Setup unternommen und die ersten VMs erstellt. Alle weiteren VMs wurden nachträglich gemäß den Projekt-Anforderungen erstellt.

Die Datenbanken wurden von den Legacy-Systemen auf die VMs in den jeweiligen Container-Datenbanken übernommen.

Nach und nach wurden zusätzliche Infrastruktur-Ressourcen benötigt und so kam es, dass man immer wieder die Plattform ausbauen musste (siehe Abbildung 1).

Folgende Ausbaustufen wurden mit der Zeit getätigt:

- Initial-Konfiguration → zwei X5 Quarter Rack pro Standort
- Erweiterung der Cell Disks von 4Tb auf 8Tb der X5-2 Storage Shelves

- Erweiterung von vier X6 Compute Nodes (je zwei Nodes per Quarter Rack)
- Erweiterung von X7 Compute Nodes (an Quarter Rack 1)
- Erweiterung von zwei X7-2 Storage-Servern (an Quarter Rack 1)
- Ersatz von alten Storage-Servern durch fünf X8-Storage-Server (an Quarter Rack 1)
- Alte X5 Storage als Zusatz-Storage (an Quarter Rack 2)
- Ausbau lokale Disks von vier auf acht Stück (je 600Gb) um /EXAVMIMGES-Verzeichnis zu erweitern

Herausforderungen

Wie in jedem Projekt kann es auf dem Weg zum Ziel kleine Probleme geben, die es zu meistern gilt.

In den folgenden Kapiteln werden diverse Herausforderungen beschrieben, die uns in den produktiven Jahren auf dem Weg zum heutigen Stand begegnet sind.

Es ist keine abschließende Aufzählung, aber diese sind die Nennenswertesten und blieben bei uns in guter Erinnerung.

System-Herausforderungen

Security-Anforderungen und RPMs

In jedem IT-Betrieb wird Security großgeschrieben. Demnach muss geschaut werden, dass die Einhaltung der Grundschutz-Vorschriften der jeweiligen Firma auf den IT-Systemen sichergestellt ist.

Um dies technisch zu realisieren, wird größtenteils die Hilfe zusätzlicher Utilities benötigt. Als Beispiel werden Standard-RPM-Pakete (z.B. rsyslog) verwendet, etwa um systemnahe Informationen abzuzeihen und diese dann zentral weiter zu analysieren.

Diese spezifischen RPMs werden grundsätzlich auf allen Exadatas als „non Standard“ RPMs angeschaut und müssen bei einem Patching speziell behandelt werden.

Das heißt konkret, dass sie entweder deinstalliert werden müssen oder via „exclude lists“ explizit behandelt werden. Wenn sie deinstalliert werden, müssen allfällige Konfigurations-Files gesichert werden. Nach dem Patching können diese RPMs wieder installiert werden – jedoch muss wie im Fall vom 19c-Patching, wo auf Linux 7 upgegradet wird, auf die Linux-Versionen-Kompatibilität der RPMs geachtet werden.

In diesem Projekt wurde bewusst darauf geschaut, dass die Anzahl der „spezifischen“ RPMs so gering wie möglich ausfällt. Zudem wurde ein Inventory für die benötigten RPMs erstellt, sodass diese bei Bedarf (z.B. Patching) immer griffbereit sind und wiederverwendet werden können. Bei eventuell vorkommenden Versions-Upgrades des Linux-Betriebssystems wird das Inventory zentral nachgeführt.

Filesysteme und Shares

Jeder Datenbank-Server hat als Grundlage ein Betriebssystem und infolgedessen werden Filesysteme auf verschiedene Disks oder Partitionen erstellt, um Dateien abzuspeichern. Da die Virtual Disks – also die lokalen Disks – der jeweiligen DomU eine begrenzte Größe aufweisen, wird es mit größter Wahrscheinlichkeit knapp, auf dessen Filesystemen sämtliche Log- und Trace-Files sowie etwaige applikationsnahe Dateien (z.B. .csv- oder .xml-Files) abzuspeichern.

Rein technisch wäre es kein Problem, die Virtual Disks zu erweitern und demzufolge zu vergrößern. Allerdings werden die Virtual Disks von den lokalen Diskplat-

ten der physikalischen Disks erstellt und diese sind leider endlich. Unter anderem auch, weil während der Upgrades der Grid Infrastructure oder wenn zusätzlichen Oracle Homes benötigt werden, wiederum Virtual Disks für die Images der soeben genannten Software-Produkte benötigt werden.

Daher wird schnell klar, dass die Erhöhung der virtuellen Disks mit Vorsicht zu genießen ist.

Die Vergrößerung der Filesysteme wäre schnell und schmerzlos, da sämtliche Aktionen online durchgeführt werden können, jedoch könnte es aus den bereits genannten Gründen schnell in einer Sackgasse enden.

Deshalb ist es naheliegend, gängige Verfahren wie ACFS- oder NFS-Shares zur Verfügung zu stellen, um die Log- und Trace-Files sowie applikationsnahen Files abzuspeichern. In diesem Projekt entschied man sich dafür, NFS-Shares einzusetzen, dies auch hauptsächlich, um nicht unnötig „teure“ Disk-Ressourcen der Storage-Server zu verwenden. Zudem war zum Zeitpunkt der Entscheidungs-Findung, auf NFS-Shares zu setzen das Know-how rund um das NFS-Shares-Management bei den Linux System-Administratoren verbreiteter als das bezüglich ACFS.

Erhöhung der lokalen Disk-Platten

In diesen vier Jahren haben wir bereits etliche Male Patches einspielen müssen und auch verschiedene Grid-Infrastructure-Upgrades durchgeführt. Mit den Upgrades des Grid-Infrastructure-Stacks wurde auch immer die entsprechend passende RDBMS-Version installiert.

Bei der virtualized Exadata werden ISO-Images als Virtual Disks verwendet, um neue Grid- oder Oracle-Homes zu installieren. Diese werden auf der Dom0 angebunden und als Sparse Disk der jeweiligen DomUs weiter provisioniert. Trotz des „Copy on Write“-Mechanismus der Sparse Disks wird irgendwann die Höchst-Kapazität der lokalen Disk-Platten erreicht.

Demnach ist irgendwann ein Ausbau der lokalen Disk-Ressourcen unumgänglich! Auch hierbei ist die Lösung schnell herbeigeführt... vier neue Platten, sogenannte „Disk Expansion Kit“ per Compute Node bestellen und folgende Schritte, die im Exadata Maintenance Guide beschrieben sind, ausführen:

top - 18:34:15 up 10 days,	3:36,	1 user,	load average: 1.38,	1.46,	1.46
Tasks: 484 total,	4 running,	480 sleeping,	0 stopped,	0 zombie	
Cpu0: 1.0%us,	1.4%sy,	0.0%ni,	97.6%id,	0.0%wa,	0.0%hi, 0.0%si, 0.0%st
Cpu1: 0.7%us,	0.7%sy,	0.0%ni,	98.3%id,	0.0%wa,	0.0%hi, 0.0%si, 0.3%st
Cpu2: 0.7%us,	2.7%sy,	0.0%ni,	96.3%id,	0.0%wa,	0.0%hi, 0.0%si, 0.3%st
Cpu3: 1.7%us,	1.4%sy,	0.3%ni,	96.6%id,	0.0%wa,	0.0%hi, 0.0%si, 0.0%st
Cpu4: 0.0%us,	0.0%sy,	0.0%ni,	0.0%id,	0.0%wa,	0.0%hi, 0.0%si, 100.0%st
Cpu5: 0.0%us,	0.0%sy,	0.0%ni,	0.0%id,	0.0%wa,	0.0%hi, 0.0%si, 100.0%st
Cpu6: 0.0%us,	0.0%sy,	0.0%ni,	0.0%id,	0.0%wa,	0.0%hi, 0.0%si, 100.0%st
Cpu7: 0.0%us,	0.0%sy,	0.0%ni,	0.0%id,	0.0%wa,	0.0%hi, 0.0%si, 100.0%st
Mem: 123304248k total,	20590912k used,	102713336k free,	616668k buffers		
Swap: 16777212k total,	0k used,	16777212k free,	6786116k cached		

Tabelle 1: vCPU-Setting innerhalb der DomU

- Plastik Filler mit Disk-Platten ersetzen
- Ein Alert wird ausgegeben, wo die Disks erkannt werden und automatisch in RAID aufgenommen werden → `dbmcli -e list alerthistory`
- Ein weiterer Alert wird ausgegeben, wenn der „resync“ der Platten durch ist → `dbmcli -e list alerthistory`

Anders als beim Hinzufügen oder Entfernen von Disks respektive Grid Disks in ASM-Diskgruppen kann man bei dieser Operation keine „rebalance power“ definieren.

Demzufolge beginnen die Spindeln gleich, nachdem diese ins RAID aufgenommen wurden, sich zu synchronisieren. Unweigerlich löst diese Aktion eine hohe Anzahl an I/O aus... in unserem Fall dauerte dies ca. 18 Stunden!!!

Der Ausbau wurde zu einer Randzeit ausgeführt, trotzdem litten die VMs durch das hohe Aufkommen an I/O während dieser Zeit unter massiven Performance-Problemen → I/O Waits!

Nachdem die Synchronisation zu Ende war, konnte das Filesystem des Mount-Points /EXAVMIMAGES erweitert werden.

Unser Fazit aus dem Ausbau der lokalen Disk-Ressourcen: Sparsam damit umgehen und eine gute Planung im Vorfeld ist unabdingbar.

Erhöhung von vCPUs

Eine weitere Eigenheit der virtualized Exadata ist die Möglichkeit der dynamischen Erhöhung der vCPUs für die DomUs. Diese Eigenschaft wird durch die Xen-Virtualisierungstechnologie, die in der virtualized Exadata (non X8M) im Einsatz ist, zur Verfügung gestellt. Dabei kann die vCPU-Einstellung bis auf die maximale Anzahl an vCPUs (maxvcpus) erhöht werden, die auf der Dom0 im `vm.cfg` Konfigurations-File definiert wurde.

Zu Beginn des Projektes, als die Plattform und deren Ressourcen noch nicht voll unter Last standen, funktionierte dieses Feature wie erwartet und problemlos.

Aber im Laufe der Zeit, mit zunehmender Last, wird das Allokieren und Deallo-

zieren von vCPUs zu einem waghalsigen Glücksspiel.

Soll heißen, wenn das Kommando `xm vcpu-set <domain_name> <Anzahl an vCPU>` absetzt, wird das Kommando auch ausgeführt, jedoch sieht dann das vCPU-Setting innerhalb der DomU wie folgt aus (siehe Tabelle 1):

Die neu allozierten vCPUs bleiben auf 100% „steal time“. Die Bedeutung von „steal time“ ist die Wartezeit einer vCPU auf eine physikalische CPU, während der Hypervisor eine andere vCPU bedient. Demnach wird die DomU warten müssen, bis die Ressourcen – an dieser Stelle die Cores einer physikalischen CPU – freie Kapazitäten aufweisen.

Es versteht sich von selbst, dass, wenn die Plattform unter ständiger Last sein sollte, sich diese Situation nicht von selbst lösen kann. Zwar funktioniert die DomU tadellos weiter; um aber auf das erwartete Ziel zu kommen, ist ein Reboot notwendig! Dabei werden die vCPUs auf einer anderen physikalischen CPU des entsprechenden Compute Node alloziert.



MUNIQSOFT
— CONSULTING —



Performance-Tuning mit IQ

Mehr Power für Ihre Oracle Lösungen!

Nutzen Sie unseren proaktiven Datenbank-Healthcheck als Startschuss für die Optimierung Ihrer Oracle Datenbanken.

Ungebremst ans Ziel mit der Muniqsoft Consulting GmbH
www.muniqsoft-consulting.de

ORACLE® Gold Partner

Specialized
Oracle Database



Jetzt Beratungstermin vereinbaren:
+49 89 62286789-39

Noch schlimmer wird es, wenn man ein Downgrade von vCPU tätigen will und das System unter Last ist. Hierbei werden sämtliche vCPUs dealloziert und einem anderen CPU Socket zugewiesen, demzufolge rebootet die DomU ungewollt!

Fazit... das Feature funktioniert wie erwartet, solange genügend Ressourcen vorhanden sind. Danach eher vorsichtig damit umgehen, was gleichbedeutend einer „bedingten“ dynamischen Konfigurationsoption ist.

Erweiterung von Compute Nodes ins Enterprise Manager

Hierbei geht es nicht um die physikalische Erweiterung eines oder mehrerer physikalischer Compute Nodes. Dieser Prozess ist ziemlich „straightforward“ und eher unkompliziert. Sondern es geht mehr darum, wie die Compute Nodes in den Enterprise Manager aufgenommen werden können, vor der „Elastic Configuration“-Zeit.

Als das Projekt mit der virtualized Exadata anfang, existierte die „Elastic Configuration“ noch nicht, diese kam kurz darauf. Daher bestand die Problematik, wie man eine bestehende virtualized Exadata um die neuen Compute Nodes erweitert. Mittels OEDA (Oracle Exadata Configuration Assistant) musste nochmals eine neue und komplette Exadata-Konfiguration erstellt werden. So weit, so gut – und auch das Deployment funktioniert problemlos... Die Herausforderung war dann eher, die neuen Compute Nodes in den Enterprise Manager zu „discover“. Mit der OEDA-Konfiguration wurde nochmals ein komplettes „databasemachine.xml“-File erstellt, in dem wiederum sämtliche Komponenten nochmals enthalten waren. Dies wollte man vermeiden, damit keine zweite Exadata im OEM erscheint, sondern man wollte nur die physikalischen Hosts und deren VMs hinzu-

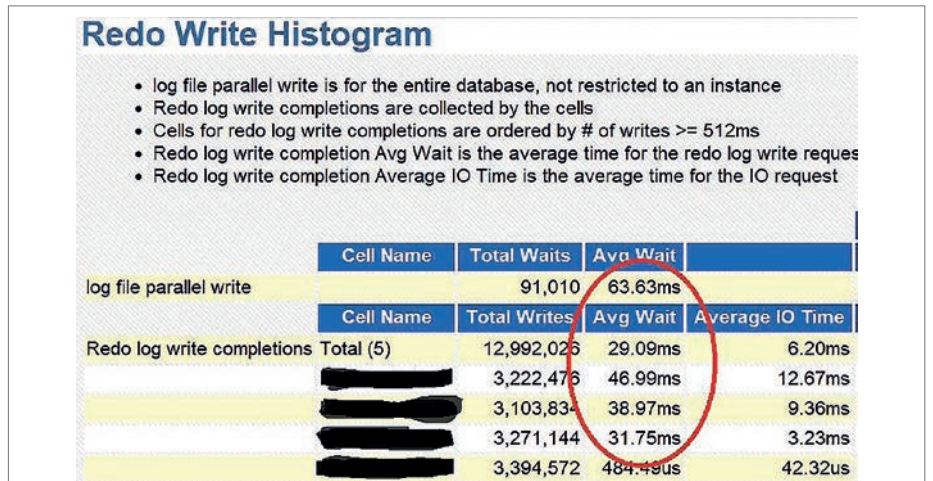


Abbildung 2 (Quelle: Daniele Massimi)

fügen. Um dieses Problem zu lösen, musste damals das „databasemachine.xml“ manuell angepasst werden, um die neuen Compute Nodes hinzuzufügen.

Storage-Server-Herausforderungen

Erhöhung Storage-Server-Diskkapazität

Wie bereits erwähnt wurde, startete man das Projekt mit einer X5 Quarter Rack Exadata. Die erste Generation von X5-Storage-Servern hatte zu Beginn nur 4TB-Disks. Bedingt durch den schnellen Fortschritt der Datenbank-Migrationen auf der neuen Plattform wurde auch schnell die Diskkapazität der Storage-Server knapp. Da die zweite Generation der X5-Storage-Server mit 8TB ausgeliefert wurde, entschied man sich, die 4TB-Disks gegen die größeren 8TB-Disks auszutauschen.

Diese Aktion wurde in Zusammenarbeit mit Oracle ACS unternommen und es wurde wie folgt vorgegangen:

- Erhöhung des Parameters „disk_repair_time“ auf 12h
- Storage Cells offline setzen
- Disk- und RAID-Controller-Austausch durch ACS
- Re-Imaging der Storage Cells
- Neuerstellen der Cell Disks und Grid Disks
- Rebalancing der ASM-Diskgruppen starten

Natürlich wurde dabei eine Storage-Zelle um die andere herumgebaut, somit konnte der Umbau „online“ durchgeführt werden.

Einige Zeit später wurde nochmals ausgebaut... es kamen zwei zusätzliche X7-Storage-Server hinzu. Auch diese Erweiterung ist ziemlich unkompliziert. Sobald die Storage-Server physisch eingebaut sind, müssen folgende Tasks ausgeführt werden:

- Erstellen der Cell Disks
- Erstellen der Grid Disks für alle Cluster mit korrekter Größe
- Writeback aktivieren

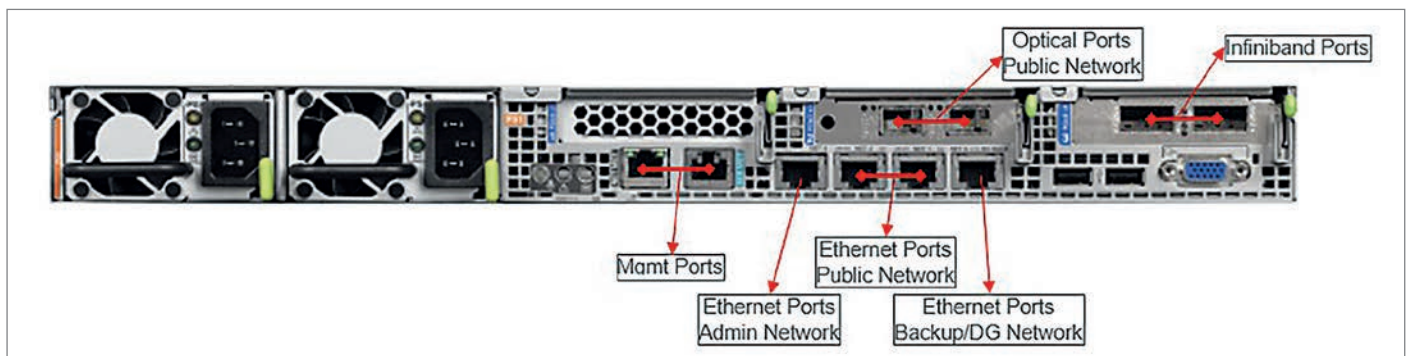


Abbildung 3: X5 Compute Node (Quelle: Daniele Massimi)

- Erstellen des Flashlog und Flashcache
- cellip.ora ergänzen auf allen DomUs
- Rebalancing der ASM-Diskgruppen starten in allen ASM-Instanzen

In den X7 Storage sind 10TB-Disks eingebaut, soweit kein Problem... bis auf das Detail, dass bei ASM ab der Version 12.1.0.2 nur noch Disks mit der gleichen Größe unterstützt werden. Demnach konnten 2TB der neuen X7-Storage-Server-Disks, also insgesamt 24 Disks x 2TB = 48TB, nicht benutzt werden.

Beim Einbau der neuen X7-Storage-Server fiel uns eine weitere Besonderheit bei der Aktivierung der Writeback-Funktionalität auf. Dabei, entgegen dem früheren Verfahren mit der Storage-Server-Software < 11.2.3.3.1, das einen Restart des CELLSRV-Service bedingte, kann das Writeback ohne Restart des CELLSRV eingeschaltet werden. Das Ganze ist auch in der My Oracle Support Node 1500257.1 nachzulesen.

Entsprechend sind wir nach dieser MOS Note vorgegangen und hatten auch für einen gewissen Zeitraum kein negatives Feedback seitens der Applikations-Benutzer.

Irgendwann kamen dann die ersten Querelen über mangelnde Performance... die AWRReports zeigten durchwegs immer das Gleiche „log file sync wait“-Symptom.

Das Flash-Log-Write-Histogramm in der Exadata-Statistics-Sektion innerhalb des Reports sah wie in *Abbildung 2* aus:

Die zwei neuen Storage-Server hatten extrem schlechte Antwortzeiten (ms anstelle von us) für ihre Flashlogs, sodass hierbei sämtliche Transaktionen darunter litten. In unseren Fall half das Restarten des CELLSRV-Service dabei, die Antwortzeiten wieder auf Mikrosekunden herunter zu bringen. Es stellte sich jedoch in der Folge heraus, dass dieser Zustand aufgrund eines Bugs im Bereich Memory-Management der Storage-Servers (Memory Leak) und IORM, ausgelöst wurde. Der Patch war bis Dezember 2019 noch nicht vorhanden, was gelegentliche CELLSRV Services-restarts als Workaround notwendig machte.

Netzwerk-Herausforderungen

Wie bereits erwähnt, ist einer der Gründe, das Projekt mit der virtualized Exadata zu realisieren, die Isolation der Netzwerke zu bewerkstelligen. Zu Projektbeginn sahen die X5 Compute Nodes wie in *Abbildung 3*

aus. Also ein Quad Ethernet Ports Interface und ein Two Optical Ports Interface. Somit konnten die Security-Anforderungen – jede Netzwerkzone mit physikalisch getrenntem Kabel zu bedienen – entsprechend erfüllt werden. Nämlich Netz-Zone 1 via Ethernet Ports und Netz-Zone 2 via Optical Ports.

Die verschiedenen VLANs in jeder Netz-Zone wurden dagegen mit VLAN Tagging, also nach dem IEEE-Standard 802.1Q implementiert.

Um dies auf einer virtualized Exadata zu realisieren, müssen einige Konfigurationen getätigt werden.

- Konfigurieren der firmeneigenen Switch Ports als „Port Trunking“ inklusive Hinzufügen sämtlicher VLANs, die benötigt werden
- Erstellen der Bridges (Virtual Switches) für jedes VLAN auf den Compute Nodes (Dom0)
- Konfigurieren der DomUs mit den entsprechenden VLANs im vm.cfg-File
- Udev Rules für die Interfaces innerhalb der DomU definieren

Grundsätzlich nicht so eine große Sache, vielmehr liegt die Herausforderung im Management solcher Konfigurationen. Vor allem, wenn aus Security-Gründen nicht überall die gleichen VLANs angebunden werden dürfen. Was auch vorkam, ist, dass aufgrund neuer Anforderungen neue VLANs abgeschlossen werden mussten. Außer dem Aktivieren der VLANs auf DomU-Ebene kann alles online konfiguriert werden... für die DomU-Aktivierung ist ein Reboot notwendig, also muss ein Wartungsfenster organisiert oder abgewartet werden!

Datenbank-Herausforderungen

Multitenancy und RAC One Node
Gleich zu Beginn des Projektes beschloss man, die Multitenant-Option einzusetzen. Es war das Jahr 2015 und Multitenant meines Erachtens nach damals noch nicht weitverbreitet. Nichtsdestotrotz wollte man dieses Feature einsetzen, um die größtmögliche Konsolidierungsdichte herbeizuführen.

Neben der Multitenant-Option hatte man sich auch für den Einsatz von Real Application Cluster (RAC) One Node entschieden. RAC One Node deswegen, weil man le-

diglich die Hochverfügbarkeits-Features von RAC und keine Lastverteilung haben wollte. Zudem bietet RAC One Node auch hohen Komfort bezüglich Wartung, wie zum Beispiel bei Patchings oder Rolling-Upgrades.

Man definierte folgende Standards für die DB-Architekturen:

- Single-Instance-Datenbanken
- RAC-One-Node-Datenbanken
- RAC-One-Node-Datenbanken mit Data Guard

Und so startete man mit dem Erstellen von Container-Datenbanken. Schnell wurde klar, dass die Konsolidierungsdichte durch folgende Faktoren beeinflusst wurde:

- Character-Set-Problematik (<=12.1.0.2)
- Verfügbarkeitsklassen-Problematik
- DB-Architektur-Problematik (z.B. Data Guard)
- Interne-Daten-Schutzklassen-Problematik
- RTO und RPO gemäß definierten SLA mit Stakeholder

Somit wurden folgende CDB-Konsolidierungsdichten definiert:

- Prod CDBs → max. 15 PDBs
- Prod CDBs mit Data Guard → 1 PDB → keine Abhängigkeit bei Switch- und Failover durch andere Applikationen!
- Nicht produktive CDBs: max. 50 PDBs

Im Großen und Ganzen ist im Projekt die Einführung von Multitenant sehr positiv zu betrachten. Trotz der erwähnten Konsolidierungseinschränkungen konnte eine beachtliche Konsolidierungsdichte herbeigeführt werden. Es sind auch keine nennenswerten Probleme zu melden... nicht, dass es keinerlei Probleme gab, diese konnten jedoch immer zeitnah gelöst werden.

Wohin geht die Reise? Im Jahr 2020 werden sämtliche 12.1.0.2-Datenbanken auf 19c migriert. Mit dieser Version möchte man vor allem von den Ressourcen-Management-, Security- und Konsolidierungs-Features (zum Beispiel Unterstützung von unterschiedlichen Character-Sets) profitieren.

Patching-Herausforderungen

Über die letzten vier Jahre wurde die Plattform immer wieder mit den aktuellsten Pat-

ches versehen und verschiedene Upgrades durchgeführt (z.B. Grid Infrastructure).

Hier eine grobe Auflistung der Patching- und Upgrade-Pfade:

- Initial-Exadata-Image-Konfiguration → 12.1.2.1.3
 - Patching 12.1.2.1.3 → 12.1.3.x → 12.2.1.1.x → 19.2.5.0.0 (aktuelle Version)
- Initial-Infiniband-Version → 2.1.x
 - Verschiedene Patches bis auf 2.2.13 (aktuell eingesetzte Version)
- Initial-Grid-Infrastructure-Version → 12.1.0.2
 - Upgrades 12.1.0.2 → 12.2.0.1 → 19.4.0.0.0 (aktuell eingesetzte Version)
- Datenbanken-Versionen
 - 11.2.0.4, 12.1.0.2, 12.2.0.1, 19.4.0.0.0

Wie wohl in jedem Unternehmen, ist das Haupt-Problem, genügend viele und vor allen genügend große Wartungsfenster zu kriegen, um Patches zu installieren. Das Initial-Konzept sah kein „Online-Patching“ vor, obwohl immer auch die Möglichkeit besteht, dass die Komponenten online gepatcht werden. Dies soll heißen, dass außer den Storage-Servern die restlichen Komponenten (IB, OS, GI, RDBMS) online gepatcht werden. Die zukünftige Ausrichtung geht ganz klar in Richtung 100% Online-Patching, dafür wurden bereits Optimierungs-Aktionen an der Plattform initiiert.

Upgrade Grid Infrastructure auf 19c und OEDACLI

Das Upgrade der Version 18c oder höher des Grid Infrastructure auf einer virtualized Exadata wird nur noch durch das OEDACLI Utility unterstützt. OEDACLI ist das Command-Line Utility des OEDA-Tools und übernimmt während des Upgrades folgende Tasks:

- Discovery der aktuellen Virtualized-Exadata-Umgebung
- Hinzufügen von Virtual Disks → Dom0 und DomU für GI und RDBMS
- Provisionieren (Copy, Installation, Relink etc.) von GI und RDBMS-Software
- Upgrade und Downgrade von Grid Infrastructure Stack (inkl. cluvfy Checks)

Leider hatten wir mit diesem Tool keinen großen Erfolg und stießen immer wieder auf Fehler, die dem Utility zuzuordnen waren. Nach einigen erfolglosen OEDACLI-Patches wurde in Absprache mit Oracle

ACS das Upgrade des Grid Infrastructure mittels Workarounds durchgeführt.

Capacity-Management-Herausforderung

Auch betreffend Capacity Management macht man sich zu Beginn nicht genügend Gedanken. Je höher der Workload wird, desto mehr werden die Seiten-Effekte ersichtlich. Grundsätzlich ist zu beachten, dass genügend Reserven gebildet werden. Klar, Reserven bedeutet auch gleichzeitig Ineffizienz respektive Ressourcenverschwendung, aber unsere Erfahrung hat gezeigt, dass bei einem defensiveren Ressourcen-Einsatz mit stabileren Systemen zu rechnen ist.

Folgende Ressourcen werden via Enterprise Manager überwacht:

- vCPU
- Storage (Storage-Server und ASM)
- Memory

Da die virtualized Exadata als Konsolidierungs-Plattform genutzt wird, wird die Huge-Pages-Konfiguration auf maximal 50-60% des DomU-Gesamt-Memory definiert. Da auf der Exadata im Allgemeinen nur mit ASMM (Automatic Shared Memory Management) gearbeitet wird, muss auf die Parameter PGA_AGGREGATE_TARGET und PGA_AGGREGATE_LIMIT geachtet werden. Dieser Memory-Bereich wird mit „normalen“ Pages alloziert und so kann es vorkommen, dass bei unpassender Parametrisierung das System mit „swapping“-Problemen zu kämpfen hat.

Fazit

Grundsätzlich ist die Implementation und Integration der virtualized Exadata gelungen. Man konnte die angepeilten Ziele wie Isolation, Lizenz-Optimierung und Konsolidierung erreichen.

Erfreulich ist auch, dass die Haupt-Vorteile der Exadata-Storage-Software, also Smart Scan, auch auf der virtualized Exadata erhalten geblieben sind.

Es ist jedoch zu sagen, dass die Komplexität – ob mit oder ohne Virtualisierung – nicht unterschätzt werden darf. Die Komplexität natürlich wird aus der Kombination zwischen Exadata und Kunden-Anforderung nochmals erhöht. Technische Heraus-

forderungen wird es immer wieder auf allen Systemen geben, daher sollte man eher das Prinzip „keep it simple, stupid“ verfolgen. Auch nicht zu unterschätzen ist das Know-how der „Database-Machine-Administratoren“, das eine enorme Bandbreite aufweisen muss. Klar, dass das benötigte Know-how nicht gleich zu Beginn bei allen Administratoren vorhanden sein kann; hier muss geschaut werden, dass diese sich die benötigten Skills so rasch wie möglich aneignen können. Eine Kombination aus Kursen und Training-on-the-Job ist erfahrungsgemäß der schnellste Weg dabei.

Grundsätzlich könnte Oracle betreffend Komplexitäts-Reduktion und Know-how mit Utilities Abhilfe schaffen, ähnlich wie bei der ODA (oakcli, odadmcli und odacli). Bis dahin können selbst definierte Management Utilities in Kombination mit Automation-Tools (z.B. Ansible) eine enorme Unterstützung bieten.

Über den Autor

Daniele Massimi begann seine Informatik-Laufbahn im Jahr 1997. Seit 2000 steht Oracle im Mittelpunkt seiner Tätigkeiten. Der Informatik-Background von Daniele Massimi basiert auf einem Studium der Wirtschaftsinformatik und auf einem Master of Advanced Studies in Information Technologies an der Fachhochschule Bern. Zudem hat Daniele Massimi über die Jahre hinweg verschiedene Oracle-Zertifizierungen bis zum Oracle Certified Master (11g und 12c) erlangt. Sein Tätigkeitsspektrum als Principal Consultant bei appliedtec gmbh deckt Infrastruktur-Architekturen und -Implementationen (Cloud und On-Premises), High Availability, Upgrades, Migrationen und Virtualisierung ab. Die letzten sieben Jahre war er hauptsächlich in Engineered-Systems-Projekten (Exadata, ODA und ZDLRA) beschäftigt.



Daniele Massimi
daniele.massimi@appliedtec.ch

ODA ausfällt, sei es geplant durch Maintenance- Tätigkeiten oder ungeplant durch Hardware-Ausfälle o. Ä., steht ein baugleiches Ersatzsystem zur Verfügung, das die Betriebsaufgaben übernehmen kann.

Um nun eine Alternative für Data Guard zu entwickeln, muss man zuerst verstehen, was Data Guard eigentlich tut. Im Prinzip wird die primäre Datenbank dupliziert und durch ein permanentes Managed Recovery aktuell gehalten. Dazu werden die Änderungen nicht nur in die Online-Redologs der Primärdatenbank geschrieben, sondern parallel auch in die Standby-Redologs der Standby-Datenbank. All das wird durch Data Guard gesteuert und automatisiert. Das Prozedere ist in *Abbildung 1* schematisch dargestellt.

Im Grunde entspricht das einem Disaster Recovery einer Datenbank. Initial wird die Datenbank aus einem Backup wiederhergestellt und im Anschluss durch Einspielen aller verfügbaren Archive logs so weit wie möglich auf den neuesten Stand gebracht. Solange die wiederhergestellte Datenbank nicht per „open resetlogs“ geöffnet wird, können immer weiter Archive logs nachgespielt werden. Diese grundlegende Funktionalität hat man in allen Editionen der Datenbank. Dieses Prinzip kann man sich zunutze machen, um eine Standby-Datenbank auch mit der Standard Edition 2 zu realisieren. Man muss dazu also lediglich in regelmäßigen Abständen auf der Primärdatenbank einen Logswitch erzwingen und anschließend alle Archive logs zur Standby-Datenbank transportieren und dort einspielen. Dieser Prozess lässt sich recht einfach mit Skripten automatisieren. In *Abbildung 2* ist die Grundidee noch einmal grafisch aufbereitet.

Einzig eine synchrone Übertragung der Transaktionen zur Standby-Datenbank ist mit einer solchen Lösung leider nicht möglich. Der potenzielle Datenverlust im Disaster-Fall entspricht also dem

Aktualisierungsintervall der Standby-Datenbank. In vielen Fällen ist ein Intervall von 5 Minuten jedoch völlig ausreichend. Oftmals genügt auch ein größeres Intervall den Anforderungen. Dies ist im Einzelnen natürlich auf die betroffenen Anwendungen abzustimmen.

Im Falle eines Desasters kann die Standby-Datenbank einfach durch ein „open resetlogs“ aktiviert werden und steht danach direkt zur Verfügung. Die Ausfallzeit beträgt also im Idealfall nur wenige Minuten.

Interessant wird es bei einem geplanten Rollenwechsel zwischen Primär- und Standby-Datenbank. In einer Data Guard-Umgebung ist dies ein Einzeiler und ermöglicht zum Beispiel das Einspielen von Hardware-, Betriebssystem- oder Datenbankpatches mit einer nur minimalen Ausfallzeit. Aber auch ohne Data Guard kann man einen geplanten Rollenwechsel vornehmen. Dazu muss man der Standby-Datenbank lediglich vorgaukeln, sie wäre schon immer die Primärdatenbank gewesen. Wie realisiert man das? Beim Starten der Instanz werden die Informationen zur Datenbank aus den Controlfiles und anschließend aus den Headern der Datafiles gelesen. Danach erfolgt ein Recovery mithilfe der Online-Redologs. Für einen geplanten Rollenwechsel mit einer manuellen Standby-Datenbank muss man also lediglich beide Datenbanken stoppen und der Standby-Datenbank die Control- und Redologfiles der Primärdatenbank zur Verfügung stellen. Gegebenenfalls müssen vor dem Öffnen noch die Dateinamen für die Standby-Seite angepasst werden. Nach dem Öffnen der Standby-Datenbank wird diese sich wie die vorige Primärdatenbank verhalten. Ab diesem Punkt erfolgt der Transport der Archive logs nun in umgekehrter Richtung. Hier noch einmal die Schritte für einen geplanten Rollenwechsel:

1. Stoppen der Primär- und Standby-Datenbank
2. Kopieren der primären Redologs und Controlfiles zur Standby-Seite
3. Anpassen der Dateinamen auf der Standby-Datenbank
4. Starten der Standby-Datenbank, diese wird zur neuen Primärdatenbank
5. Starten der alten Primärdatenbank im MOUNT-Modus, diese wird zur neuen Standby-Datenbank
6. Umkehrung der zyklischen Aktualisierung

Damit sind die wichtigsten Funktionalitäten von Data Guard mit den Bordmitteln der Standard Edition 2 nachgebildet. Man kann also zwei kleine ODAs recht effektiv und vor allem preiswert zu einem robusten Gesamtsystem verbinden. Als einzige Nachteile bleiben der mögliche Datenverlust durch die Einspielverzögerung sowie der höhere Aufwand zur Implementierung einer solchen Umgebung.

Ersatz für Real Application Cluster

Bis zur Datenbank Version 18c ist Real Application Cluster (RAC) Bestandteil der Standard Edition 2, ab Version 19c jedoch nicht mehr. Falls man die Enterprise Edition einsetzt, ist RAC eine kostenpflichtige Option. Es ist also durchaus wünschenswert, einen Ersatz für RAC zu finden. Dafür muss man wissen, was RAC überhaupt bietet. Zum einen ist das eine Skalierung der Last über die Server im Cluster, bei den ODA-HA-Varianten also zwei Server. Zum anderen ist das die Absicherung von Ausfällen eines einzelnen Servers, im Idealfall kann mit der verbliebenen Instanz nahtlos weitergearbeitet werden. Die Skalierung lässt sich leider nicht ersetzen, die Ausfallsicherheit dagegen schon.

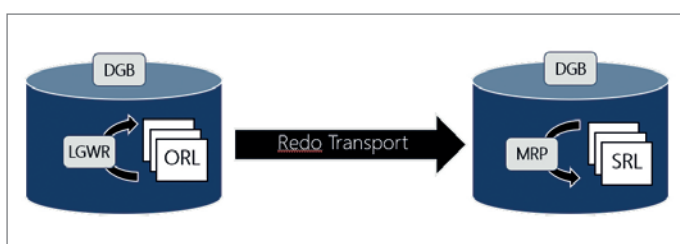


Abbildung 1: Grundprinzip Data Guard (Quelle: Marco Mischke)

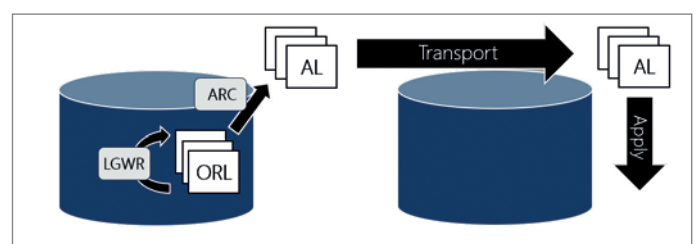


Abbildung 2: Manuelle Standby-Datenbank (Quelle: Marco Mischke)


```
Type Definition (File):
ATTRIBUTE=USR_DB_HOME
DEFAULT_VALUE=
TYPE=STRING
ATTRIBUTE=USR_DB_SID
DEFAULT_VALUE=
TYPE=STRING

Type registrieren:
crsctl add type my.app.type -basetype cluster_resource -file <File>
```

Listing 1: Type Definition für eine Cluster-Ressource

Die Alternative zu RAC auf einer ODA ist ganz einfach ein Failover- beziehungsweise Active/Passive-Cluster. Beide Bezeichnungen beschreiben dieselbe Technologie. Dabei werden die Datenbanken als Single Instance betrieben. Fällt der Server aus, auf dem eine Instanz läuft, startet die Clusterware die Instanz automatisch auf dem verbleibenden Server. Dies kann auf verschiedenen Wegen erreicht werden.

Variante 1

Die Technologie für Variante 1 ist im Prinzip schon seit Version 10 in der Oracle Clusterware verfügbar und im „Clusterware Administration and Deployment Guide“ [1] beschrieben. Man kann beliebige Ressourcen clusterfähig machen, indem ein eigenes Skript entwickelt wird, das vier Aktionen für die Ressource durchführen kann: Starten, Stoppen, Prüfen und Berei-

```
#!/bin/sh
ORACLE_HOME=${USR_DB_HOME}
ORACLE_SID=${USR_DB_SID}

case $1 in
  start)
    do_start;;
  stop)
    do_stop;;
  status)
    do_status;;
  clean)
    do_clean;;
esac
```

Listing 2: Vorlage für ein ActionScript

nigen. Dieses Skript wird als ActionScript bezeichnet. Um eine Datenbank auf diese Art ausfallsicher zu machen, wird diese also nicht per „srvctl“ registriert, sondern als eigene Ressource. Da zur Durchführung der Aktionen noch spezielle Informationen wie beispielsweise das ORACLE_HOME erforderlich sind, wird zusätzlich ein eigener Ressourcentyp mit diesen er-

```
Ressourcen-Eigenschaften (File):
ACTION_SCRIPT=/u01/app/oracle/scripts/action_db.sh
PLACEMENT=favored
HOSTING_MEMBERS=server1,server2
CHECK_INTERVAL=20
RESTART_ATTEMPTS=5
SCRIPT_TIMEOUT=300
START_DEPENDENCIES=hard(global:type:ora.scan_listener.type,ora.LISTENER.lsnr)
pullup(type:ora.scan_listener.type,ora.LISTENER.lsnr)
STOP_DEPENDENCIES=hard(ora.LISTENER.lsnr)
USR_DB_HOME=/u01/app/oracle/product/18/dbhome_1
USR_DB_SID=<MYSID>

Ressource registrieren:
crsctl add resource my.<SID>.inst -type my.app.type -file <File>
```

Listing 3: Anlegen einer eigenen Cluster-Ressource

```
$ srvctl status database -db MMISI
Instance MMISI is running on node odax3-base-01
$ srvctl modify database -db MMISI -node odax3-base-02 -force
PRCS-1011 : Failed to modify server pool MMISI
CRS-2736: The operation requires stopping resource 'ora.mmisi.db' on server 'odax3-base-01'
CRS-2738: Unable to modify server pool 'ora.MMISI' as this will affect running resources and resource groups, but the force option was not specified
$ srvctl stop database -db MMISI
$ srvctl status database -db MMISI
Instance MMISI is not running on node odax3-base-01
$ srvctl modify database -db MMISI -node odax3-base-02
$ srvctl start database -db MMISI
$ srvctl status database -db MMISI
Instance MMISI is running on node odax3-base-02
```

Listing 4: Ändern der Host-Eigenschaft einer Datenbank

weiterten Eigenschaften erstellt. Die Erstellung des Typs zeigt *Listing 1*.

Diese im Type definierten Eigenschaften werden von der Clusterware an das ActionScript als Umgebungsvariablen übergeben und können so ganz einfach verwendet werden. *Listing 2* zeigt das Grundgerüst eines solchen ActionScript.

Mit dem ActionScript und dem selbst definierten Type kann nun eine Single-Instance-Datenbank als Cluster-Ressource erstellt werden. Wie das geht, zeigt *Listing 3*.

Nachteil dieser Lösung ist, dass man von nun an immer „crsctl“ bemühen muss, um die Datenbank zu starten und zu stoppen. Wird die Datenbank per SQL*Plus gestoppt, so erkennt die Clusterware dies als Ausfall und startet die Datenbank sofort wieder neu. Auch die Patchprozesse der ODA können mit diesen Ressourcen nicht umgehen, diese sollten also vor dem Patchen entfernt werden und müssen dementsprechend im Anschluss wieder neu angelegt werden.

Variante 2

Um die Nachteile der ersten Variante zu umgehen, hat der Autor nach einer alternativen Möglichkeit gesucht. Ziel war es, die von Oracle bereitgestellten Ressourcen der Clusterware weiterzuverwenden. Wenn man sich die Eigenschaften einer Single-Instance-Ressource anschaut, so findet man dort die Eigenschaft „Configured Nodes“. Eingestellt wird diese Eigenschaft per „-node“-Schalter des „srv-

```
#!/bin/bash
# put this into $GRID_HOME/racg/usrco
#
LOGFILE=/tmp/relocate.log
GRID_HOME=/u01/app/18.0.0.0/grid
MYHOST=$(hostname)

# parse event string
for i in "$@"; do
  case $i in
    host=*)
      HOST="${i#*=}"
      shift # next argument=value
      ;;
    reason=*)
      REASON="${i#*=}"
      shift # next argument=value
      ;;
    status=*)
      STATUS="${i#*=}"
      shift # next argument=value
      ;;
    *)
      shift # unwanted info
      ;;
  esac
done

if [ "$HOST" != "$MYHOST" ] && [ "$STATUS" == "nodedown" ] && [ "$REASON" == "member_leave" ]; then
  echo $(date +%Y-%m-%d\ %H:%M:%S) " Found node $HOST down event." >> $LOGFILE
fi

for dbres in $(($GRID_HOME/bin/crsctl stat res -w "(TYPE co database) AND (DATABASE_TYPE == SINGLE)" |grep -i NAME|cut -d= -f2); do
  echo $(date +%Y-%m-%d\ %H:%M:%S) " Checking DB ressource $dbres" >> $LOGFILE
  db=$(echo $dbres|cut -d. -f2)
  oh=$(($GRID_HOME/bin/crsctl stat res $dbres -p |grep ORACLE_HOME=|cut -d= -f2)
  node=$(($oh/bin/srvctl config database -db $db|grep "Configured nodes:"|cut -d: -f2|sed -e 's/^[[:space:]]*//')

  echo $(date +%Y-%m-%d\ %H:%M:%S) " DB ressource $dbres configured on node $node" >> $LOGFILE

  if [ "$node" != "$MYHOST" ]; then
    echo $(date +%Y-%m-%d\ %H:%M:%S) " DB ressource $dbres must be relocated" >> $LOGFILE

    $oh/bin/srvctl modify database -d $db -x $MYHOST >> $LOGFILE 2>&1
    $oh/bin/srvctl start database -d $db >> $LOGFILE 2>&1
    $oh/bin/srvctl status database -d $db >> $LOGFILE 2>&1

  fi
done
```

Listing 5: Callout-Script für automatisches Relocate

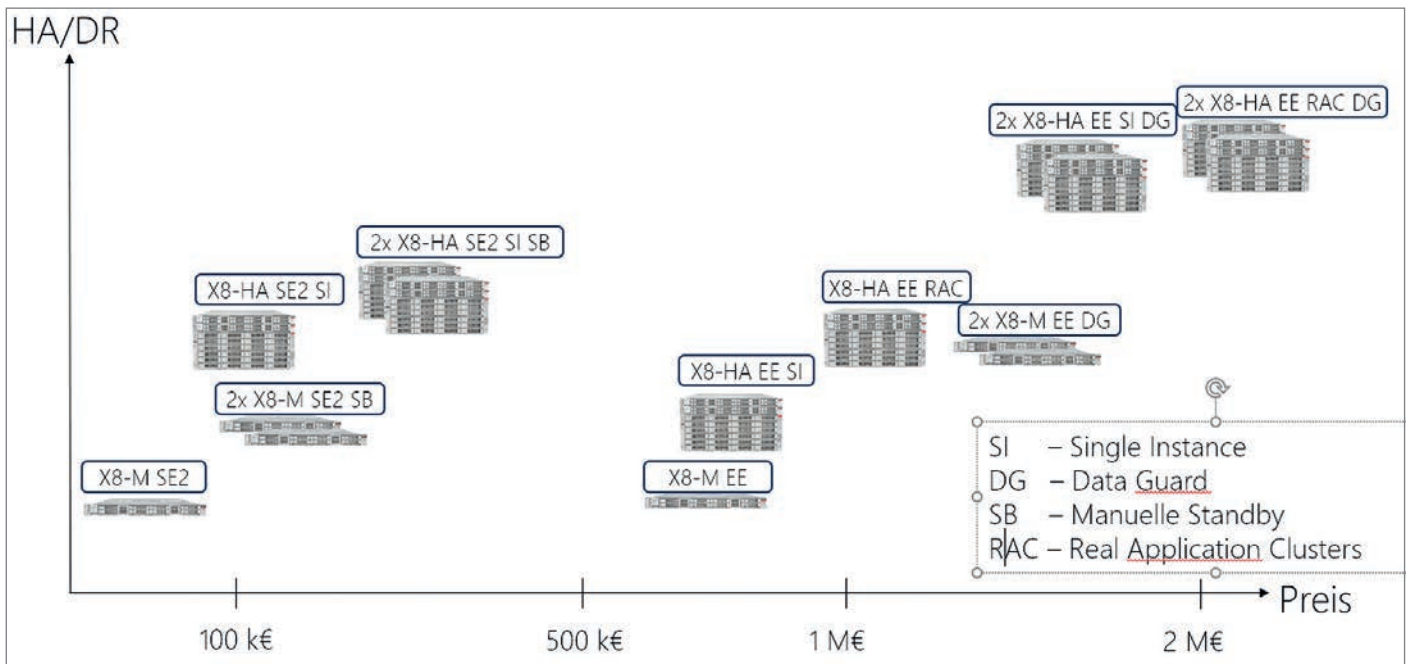


Abbildung 3: Kosten für ODAs und Lizenzierung im Vollausbau (Quelle: Marco Mischke)

ctl“. Während die Datenbankinstanz läuft, lässt sich der Wert der Eigenschaft nicht ändern; ist die Instanz gestoppt, dagegen schon. Listing 4 verdeutlicht diese Möglichkeit.

Voraussetzung für das Starten der Datenbankinstanz auf dem anderen Server ist natürlich, dass das Datenbank-Home sowie Passwort- und SPfile dort vorhanden beziehungsweise über das Shared Storage zugreifbar sind. Nun braucht es nur noch eine Möglichkeit, die Host-Eigenschaft automatisch zu ändern, wenn ein Server ausfällt. Diese Möglichkeit wurde in den Fast Application Notification Callouts gefunden, die im „Real Application Clusters Administration and Deployment Guide“ [2] erklärt sind. Jede Statusänderung im Cluster verursacht ein Ereignis, auf das mit einem Callout Script reagiert werden kann. Die Callouts werden immer auf dem Server aktiviert, auf dem die Ereignisse ausgelöst wurden. Einzige Ausnahme ist das Node-Down-Ereignis, dieses wird auf allen anderen Servern im Cluster ausgelöst. Die Konstellation ist natürlich ideal, um auf der ODA eine Instanz auf dem jeweils anderen Server zu starten. Man muss lediglich auf dieses Event reagieren und alle Datenbanken, die auf dem ausgefallenen Server liefen, auf den lokalen Server umkonfigurieren und wieder starten. Listing 5 kann

als Basis zur Implementierung eines solchen Mechanismus dienen.

Der Vorteil dieser Variante ist, dass sie sich nahtlos in die standardisierte Oracle-Infrastruktur der ODAs einfügt. Die Oracle Tools können wie gewohnt verwendet werden und auch die Patching-Mechanismen der ODA werden nicht beeinträchtigt. Außerdem kann der Automatismus einfach deaktiviert werden, indem das Callout Script aus dem Verzeichnis gelöscht wird.

Fazit

Auch ohne teure Lizenzen für die Enterprise Edition oder die RAC-Option ist ein ausfallsicherer Betrieb von Oracle Database Appliances möglich. Somit können die Kosten für eine auf ODAs basierende Infrastruktur niedrig gehalten werden. Die Appliances selbst verursachen Kosten im niedrigen 5- bis 6-stelligen Euro-Bereich; mit dem Einsatz der Standard Edition 2 bleibt man ebenfalls in diesem Bereich und kann sofort die komplette CPU-Power in Anspruch nehmen. Durch den Einsatz der Enterprise Edition steigen die Kosten schnell in 7-stellige Bereiche, wenn man die volle CPU-Power nutzen möchte. In Abbildung 3 ist dieses Kostenverhältnis auf Basis der Listenpreise grob dargestellt.

Quellen

- [1] Oracle: Clusterware Administration and Deployment Guide <https://docs.oracle.com/en/database/oracle/oracle-database/19/cwadd/making-applications-highly-available-using-oracle-clusterware.html>
- [2] Oracle: Real Application Clusters Administration and Deployment Guide <https://docs.oracle.com/en/database/oracle/oracle-database/19/racad/ensuring-application-continuity.html>

Über den Autor

Marco Mischke arbeitet seit mehr als 20 Jahren hauptberuflich mit Oracle- Datenbanken und seit mehr als 10 Jahren als Berater bei der robotron Datenbank-Software GmbH in Dresden.



Marco Mischke
marco.mischke@robotron.de



Die neue Oracle-Forms-Version 12.2.1.4 und aktuelle Trends

Frank Hoffmann, Cologne Data

Ende September 2019 erschien die nun letzte Version der 12c-Familie als „Maintenance Release“. Darin befinden sich einige interessante Neuerungen und Verbesserungen, auf die in diesem Artikel eingegangen wird. Noch etwas spannender wird es, wenn wir in die Zukunft schauen. In etwa 2 Jahren wird das nächste Major-Release erwartet, von dem Michael Ferrante auf der DOAG Konferenz + Ausstellung im November 2019 schon erste Bilder einer frühen Entwicklungsstufe gezeigt hat. Die Petition, die von der DOAG unterstützt wurde, war nicht der alleinige Auslöser, hat jedoch dazu beigetragen, dass nach 14 Jahren endlich auch die Oberfläche des GUI modernisiert wird und weitere wichtige Neuerungen umgesetzt werden.

Solider Support seit 40 Jahren

Die beiden wohl stabilsten und sichersten Produkte aus dem Hause Oracle sind und bleiben die Datenbank und Oracle Forms. Beide Produkte genießen seit mehr als 40 Jahren vollen Support und sind über die Jahre immer stabiler, sicherer und besser geworden. Auch wenn Oracle in das Kernprodukt, die Datenbank, den Löwenanteil der Entwicklung investiert hat, kommen auch zunehmend die Investitionen in den älteren, oft vernachlässigten Bruder, das erste

C-Werkzeug der Firma, in Fahrt. Fast selbstverständlich ist Forms in den aktuellen Versionen für den Betrieb der aktuellen Oracle-Datenbanken 11.2.0.4, 18.3+ und 19.3+, Windowserver (2012, 2016, 2019) sowie für alle aktuellen Unix-Systeme zertifiziert.

Das Major-Release der 12c-Familie ist komplett

Michael Ferrantes erstes Forms-Major Release, das er als Produktmanager am 23.

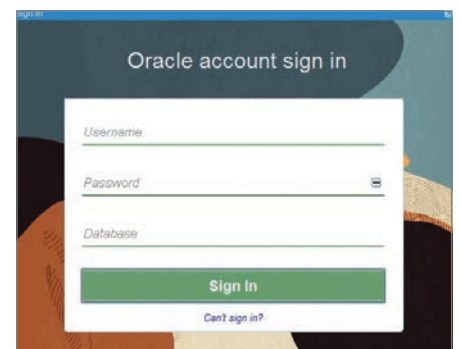


Abbildung 1: Preview von Forms (eventuell 14) in einer frühen Entwicklungsstufe (Quelle: © Oracle)

Oktober 2015 als 12c publizierte, ist nun nach seinen Angaben abgeschlossen. Viele Kunden haben bereits die neue Version zügig und unproblematisch in Betrieb genommen und profitieren von neuen Features, Bugfixes für Forms und Reports und Performance-Optimierungen. Mit diesem Release bietet Oracle bis August 2022 einen Premium-Support sowie bis August 2025 einen Extended Support an. Das Vorgänger-Release 12.2.1.3 kann noch zwölf Monate, also bis Ende September 2020 mit Support betrieben werden, es empfiehlt sich allerdings, einen Wechsel auf das neue Release durchzuführen, um Bugfixes schneller einspielen zu können. Das neue Release kann auch bereits auf dem DOAG-Community-Formserver (forms-demo.com) getestet werden.

Die wichtigsten Neuerungen kurz beschrieben

Als „Maintenance Release“ stehen Bugfixes und kleine nützliche Verbesserungen oder Erweiterungen der bisherigen Funktionen im Vordergrund, wie zum Beispiel:

1. Der WebUtil File Transfer (Client/Server) wurde um 30% beschleunigt
2. FSAL, die browserfreie Startoption, wurde in den Forms Builder integriert und bietet nun einen One-Button RUN/DEBUG in 1-2 Sekunden Aufrufzeit

und Single Sign-on sowie verbesserte Cache- und Debug-Funktionen

3. Das Custom-Color-Schema erlaubt nun, mit „desktop“ die Hintergrundfarbe der Statuszeile und den Hintergrund der Menüzeile zu konfigurieren – eine kleine, aber wichtige Ergänzung
4. Mit JDK11 steht eine neue supportete Client-Java-Version zur Verfügung
5. Im Zuge der Extended String-Anpassungen wurde die maximale Größe von Forms- und Display-Items von 32k auf 2M-1 erhöht

Diese Features können mit wenig Aufwand genutzt werden.

Alle Änderungen wurden auch bereits in das „New Feature“-Dokument von Michael Ferrante, Produktmanager von Oracle Forms, eingebunden.

<https://www.oracle.com/a/tech/docs/oracleforms-1221-newfeatures.pdf> (Version 3, Stand Ende September 2019)

Forms12c hat rund 20.000 Forms-Kunden dazugewonnen

Das Oracle-E-Business-Team führt gerade den Wechsel von Forms 10g auf Forms 12c durch. Alle Wege führen zu Forms 12c! Eine sehr gute Nachricht für die Nachhaltigkeit der Software ist der damit verbundene angekündigte Support bis 2030. Da-

mit wird nach vielen Jahren endlich der gesamte Forms-Anwenderkreis auf die aktuelle Version gezogen. Die neue Version wird also nun für einen deutlich größeren Anwenderkreis entwickelt und damit auch intensiver getestet und weiterentwickelt. Viele amerikanische Kunden können aus Gründen der Sicherheit oder der JAVA-Client-Funktionalität nicht in die Oracle Cloud wechseln und werden auf absehbare Zeit Oracle-Forms-„On-Premises“-Kunden bleiben. Das muss Oracle akzeptieren. Das sichert jedoch für Forms auch die Verfügbarkeit als On-Premises-Lösung.

Aktuelles Stimmungsbild bei den Forms-Kunden

Nach Aussage von Michael Ferrante ist der Kundenkreis sehr konstant und trotz vieler Werbeversuche in den letzten Jahren sind es nur wenige Kunden, die Forms verlassen. Das ist auch mein Eindruck. Ein Großteil der Oracle-Forms-Kunden mit komplexen Lösungen bleibt bei dem Produkt und setzt nicht auf eine aufwendige, teure Neuentwicklung mit anderen Tools und unbekanntem Ausgang. Wie die Petition gezeigt hat, haben viele Bestands-Kunden in den letzten zwanzig Jahren bis zu 3.000 Module mit viel Aufwand entwickelt, betreiben ihre Applikation für bis zu 30.000 Anwenderinnen und Anwender und verwenden Technologien, die mit einfachen HTML- oder JavaScript-Web-Anwendungen nicht abzubilden sind (OLE Automation, WIN_API-Funktionen, C-Schnittstelle, Java-Schnittstelle und Extensions über die Middle-Tier-Komponente). Neue Forms-Kunden kommen jedoch nur wenige dazu. Das liegt an hohen Lizenzkosten, an einer etwas veralteten GUI-Oberfläche, an einem in die Jahre gekommenen Forms Builder und fehlenden REST-Funktionen. Einige dieser Mängel sollen nun mit dem nächsten Release abgestellt werden.

Damit wird Forms wieder ein Stück interessanter.

Oracle arbeitet bereits an einem neuen Major-Release

Seit etwa zwei Monaten und für weitere mindestens achtzehn Monate plant und entwickelt das Oracle-Forms-Team nun das neue Major-Release. Dieses wird dann wieder wahrscheinlich fünf Jahre (2026)

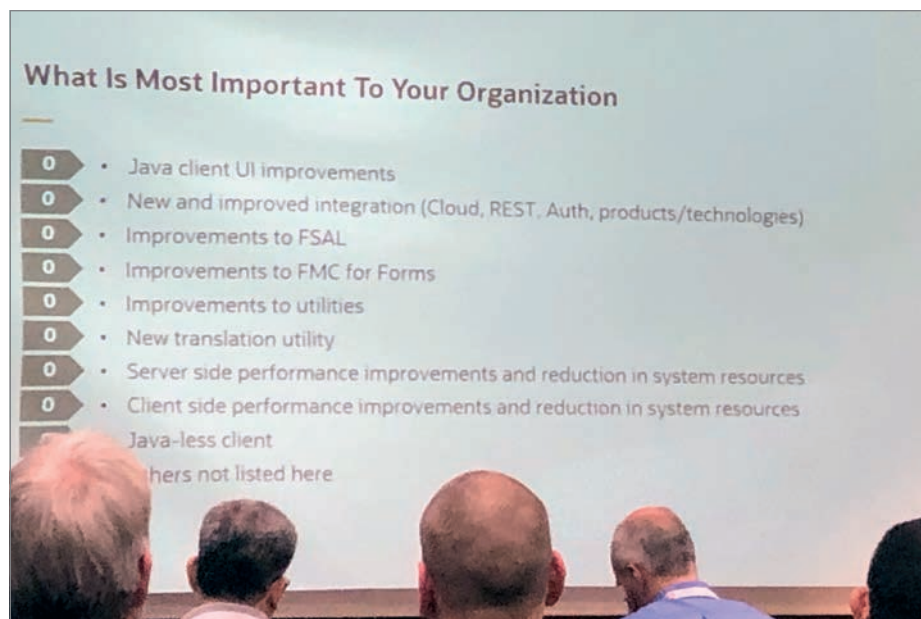


Abbildung 2: Präsentation der Anwenderwünsche auf der DOAG 2019 Konferenz + Ausstellung (Quelle: Frank Hoffmann)

Premium Support und drei Jahre Extended Support (2029) bekommen. Schon mit dem nächsten Release wird Forms auf jeden Fall schon einmal fünfzig Jahre alt. Für die neue Version wäre sicherlich ein Beta-Programm interessant, um frühzeitig das neue GUI „Look & Feel“ mit der eigenen entwickelten Forms-Applikation testen zu können. Oracle hat angefangen, wie auch in der DOAG-Petition gefordert, eine Modernisierung des Client GUI durchzuführen. Einige Beispiele dazu wurden live auf der DOAG 2019 Konferenz + Ausstellung von Michael Ferrante vorgestellt und mit ca. fünfzig Personen, die die DOAG Forms Community in Nürnberg besuchten, diskutiert. Für Oracle Forms war die DOAG-Messe diesmal wie eine Zukunftswerkstatt, die man noch etwas mehr hätte nutzen können. Mit zehn Vorträgen und einem weiteren wichtigen FORMSDAY am 20.2.2020 in Frankfurt, zu dem auch Michael Ferrante live zugeschaltet werden soll, ist die deutsche Community weiterhin sehr aktiv. Diese Aktivitäten in der deutschen Oracle-Anwendergruppe werden auch vom amerikanischen Oracle-Forms-Team registriert und geschätzt.

Oracle-Forms-Petition der DOAG

Die Petition von Dr. Menge und dem Autor dieses Artikels wurde im September 2019 offiziell von Stefan Kinnen an Oracle übergeben. Die Petition fordert im Kern eine Client-GUI-Modernisierung mit einem neuen „Look & Feel“ und eine HTML-Lösung basierend auf JavaScript. Aus einer Liste von über hundert Petitions-Unterzeichnern wurden die fünfzig größten Firmen ausgewählt und in einer Liste an Oracle übergeben, wenn sie ihr Okay dazu gegeben hatten. Oracle hat diese Petition genau geprüft und mit dem Entwicklerteam bereits eine Reihe von Ideen und Vorschlägen dazu entwickelt. Beide Forderungen stehen auf der Feature-Optionen-Liste des neuen Release. Diese Optionen werden in den nächsten Monaten noch evaluiert und fließen dann in das neue Release ein. Eine Übersicht zu den vorgeschlagenen Punkten befindet sich in der *Abbildung 2*. Es sind zu den Punkten aus der Petition noch weitere Kundenwünsche dazugekommen. Ein Beispiel zu einem neuen Client GUI (*siehe Abbildung 3*), der dem neuen Oracle „Look & Feel“ des aktuellen Oracle-Webauftritts nachempfunden

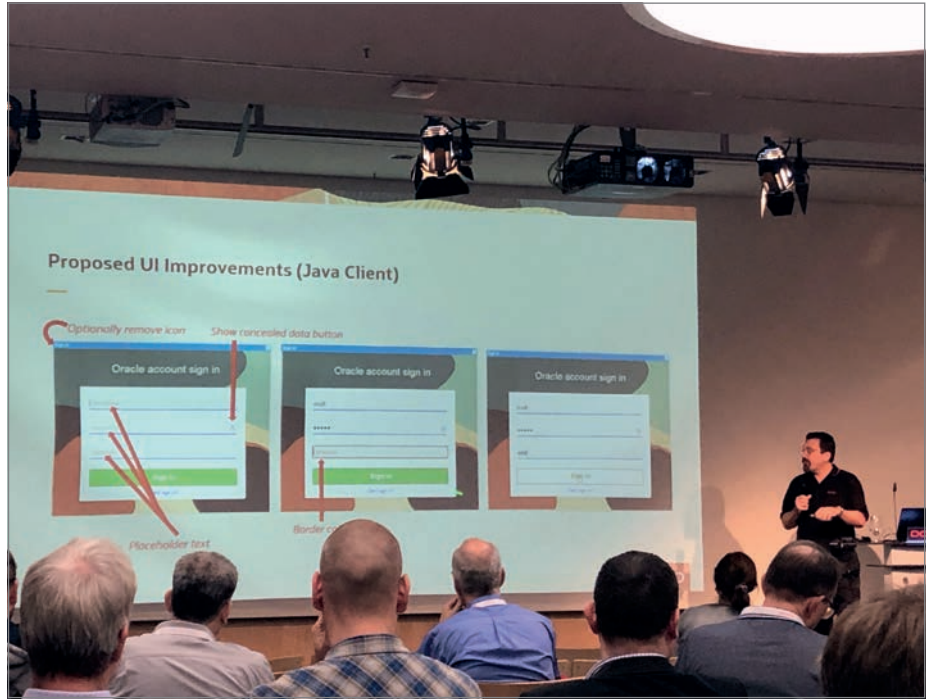


Abbildung 3: Präsentation einer neuen Client GUI auf der DOAG 2019 Konferenz + Ausstellung (Quelle: Frank Hoffmann)

wurde, konnte bereits auf der DOAG 2019 detailliert und live von Michael Ferrante präsentiert werden. Alle neuen Optionen und Änderungsvorschläge unterliegen jedoch hier dem „SAFE Harbour Statement“ – Oracle verpflichtet sich, diese Ideen nicht wirklich umzusetzen. Nach aktuellem Stand ist das also noch eine Vision. Das Facelift wird aber sicherlich kommen. Jetzt ist ein guter Zeitpunkt, um persönliche Änderungswünsche rund um Oracle Forms an den Oracle-Forms-Produktmanager Michael Ferrante via E-Mail oder Twitter zu schicken. In der Vergangenheit wurden viele Änderungswünsche, die mit vertretbarem Aufwand machbar waren, auch umgesetzt. Für die kommenden zwei bis drei Jahre empfiehlt Oracle, auf die aktuelle Forms-Version zu migrieren und bestehende neue Funktionen zu nutzen. Mit der aktuellen Version kann es dann auch nahtlos in das zukünftige Release gehen.

Über den Autor

Frank Hoffmann beschäftigt sich seit mehr als fünfundzwanzig Jahren mit Oracle Forms. Er war selbst über drei Jahre bei Oracle als Senior-Berater tätig und hat in dieser Zeit und auch danach eine Vielzahl von Oracle-Forms-Lösungen entwickelt. Viele dieser Softwarelösungen sind

bis heute absolut zuverlässig und sicher in Betrieb. Seine Firma hat sich seit 2015 auf die Entwicklung mit Oracle Forms spezialisiert und arbeitet seitdem täglich mit dem Produkt. Er betreibt den Forms-Demo-Server für die DOAG Forms Community in der Oracle Cloud, hat einen Oracle-Forms-Modernisierungsblog, ist zusammen mit Dr. Menge Autor der Forms-Petition sowie regelmäßiger Sprecher auf der DOAG-Konferenz und dem DOAG Forms Day. Weiterhin gehört er zu den zehn führenden Leitern der internationalen Oracle Forms Community, die Forms-Entwicklern bei Problemen helfen, und veröffentlicht Artikel über die Geschichte von Forms. Außerdem beschreibt er Ideen aus der Praxis, wie Forms-Kunden den Nutzen ihrer Forms-Anwendungen verbessern können.



Frank Hoffmann
fh@cologne-data.de



Brainwash your Developers – Richtiges Java für PL/SQLer

Wolf G. Beckmann, TEAM

Bei neuen Entwicklungen ist JavaEE im Backend als Standard gesetzt. Als PL/SQL-Entwickler kennt man sich gut mit der Verarbeitung und Speicherung von Daten aus, hat aber typischerweise eine datenbankorientierte Sichtweise. Java geht da einen anderen Weg, den es lohnt, mitzugehen.

In Java hat sich die Art, wie auf persistente Daten, also Daten, die in der Datenbank gespeichert werden, zugegriffen wird, über die Jahre stark verändert. Am Anfang sind die „Datenbank-Entwickler“ genauso vorgegangen, wie es in PL/SQL auch gemacht wurde. Über JDBC (Java Database Connectivity) wurden native SQL-Abfragen, Updates und auch DDL-Statements an die Datenbank gesendet. Dabei wurde das SQL-Statement als String an JDBC übergeben, gegebenenfalls noch Parameter gesetzt und ausgeführt. Das Ergebnis wurde in einem speziellen ResultSet abgearbeitet (siehe Listing 1).

```
public void jdbcExample() throws SQLException {
    Connection con = getConnection();

    PreparedStatement stmt = con.prepareStatement(
        "select * from Location where code = ?");
    stmt.setString(1, "PB");
    stmt.execute();
    ResultSet res = stmt.getResultSet();

    while (res.next()) {
        System.out.println(res.getString("CITY"));
        System.out.println(res.getString("COUNTRY"));
    }
    stmt.close();
}
```

Listing 1: Select via JDBC

```

public class Department {
    private String name;
    private String city;
    private List<Employee> employees;

    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    ...
}

public class Employee {
    private String name;
    private String job;
    private Float salary;

    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    ...
}

```

Listing 2: Beispiel POJOs

```

@Entity
public class Department {
    @Id
    private String name;
    private String city;
    @OneToMany(mappedBy = "department", cascade = CascadeType.ALL)
    private Set<Employee> employees;

    public void addEmployee(Employee employee) {
        if (employees == null) {
            employees = new HashSet<Employee>();
        }
        employees.add(employee);
        employee.setDepartment(this);
    }

    public void remEmployee(Employee employee) {
        getEmployees().remove(employee);
        employee.setDepartment(null);
    }
    ...
}

public class Employee {
    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    private Long id;
    private String name;
    private String job;
    private Float salary;
    @ManyToOne
    private Department department;

    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    ...
}

```

Listing 3: Beispiel Entities

Einige Nachteile dieser Vorgehensweise waren, dass relativ viel Code notwendig war, das Statement erst zur Laufzeit auf seine Korrektheit geprüft wurde und die Daten umkopiert werden mussten.

Als nächster Schritt wurde ein Pattern, also ein Muster für die Entwicklung, eingeführt: das DAO (Data Access Object) Pattern. Das DAO kapselt die Persistierung in eine eigene Klasse und stellt Methoden zum Laden, Abfragen, Verändern und Speichern eines Objektes bereit, wobei das Objekt eine Repräsentation einer Datenbank-Tabelle ist. Durch diese Kapselung wurde schon viel erreicht: Der eigentliche Business-Code arbeitet mit Objekten und ist aufgeräumter, da die umständlichen JDBC-Aufrufe in der DAO-Klasse liegen. Dennoch gibt es unter anderem die Nachteile, dass die Verwaltung der Objekte explizit im Code vorgenommen werden muss, das Durchlaufen von Hierarchien aufwendig ist und sich der Entwickler weiter um die Datenbank kümmern muss.

Das Java Persistence API (JPA)

Dann wurde das Thema Persistierung noch einmal völlig neu überdacht. Java und die Verwendung der Objekte sollten im Vordergrund stehen und nicht mehr die Speicherung der Daten.

Das Ergebnis war JPA. JPA ist ein definiertes API, das von verschiedenen ORM-Mappern (objektrelationalen Mappern) wie Hibernate oder EclipseLink implementiert wird. Diese ORM-Mapper sind dann beispielsweise Bestandteil des jeweiligen Application-Servers.

JPA versucht, den Entwickler so wenig wie möglich von der Persistierung spüren zu lassen. In Java werden die Daten in Objekten gehalten und diese sollten lediglich persistiert werden. Es sollte die Rückkehr zu den „Plain Old Java Objects“ (POJO) sein. Ein POJO besteht nur aus den Attributen sowie den dazugehörigen Gettern und Settern (*siehe Listing 2*).

Persistieren von Daten

Um diese POJOs persistieren zu können, werden sie mit Annotationen zu Entities erweitert (*siehe Listing 3*). Dabei müssen sie auch einigen Anforderungen genü-

gen. Sie brauchen beispielsweise eine eindeutige Id. Des Weiteren müssen die Beziehungen näher beschrieben werden. Um (vernünftig) eine Liste von Unterob-

jekten (in *Listing 3* die Employees) zu persistieren, wird typischerweise im Unterobjekt eine Rückreferenz benötigt. Um diese Rückreferenz einfach zu bearbei-

ten, werden zwei Hilfsmethoden im Entity hinzugefügt (hier `addEmployee()` und `remEmployee()`).

Über die Annotation `@Id` wird das Attribut mit dem Primärschlüssel angegeben, wobei die Verwaltung auch von JPA über die Angabe `@GeneratedValue` übernommen werden kann. Bei der `@OneToMany`-Annotation wird in dem Beispiel zum einen angegeben, in welchem Attribut die Rückreferenz gespeichert ist (`mappedBy`), und zum anderen, wie mit den Unterentitäten umgegangen werden soll. Dazu später im Beispiel mehr.

Die Entities können nun vom EntityManager (in den Beispielen in `em` instanziiert) verwaltet werden. Der EntityManager überwacht dazu alle ihm anvertrauten Entities und wenn diese sich im Rahmen einer Transaktion (JPA-Transaktion, nicht Datenbank-Transaktion) verändern (anlegen, ändern, löschen), werden diese Veränderungen in der Datenbank persistiert. Oder anders ausgedrückt: Es werden die entsprechenden Inserts, Updates und Selects gegen die Datenbank ausgeführt.

In *Listing 4* wird gezeigt, wie ein Entity gespeichert wird.

Wie man sieht, muss man sich auch nicht selbst um die Transaktion kümmern, sondern braucht, dank JTA (Java Transaction API), nur den Einstiegspunkt der Transaktion zu annotieren. Die Methode und auch alle von der Methode aufgerufenen Methoden verwenden eine gemeinsame Transaktion, die mit dem Verlassen beendet wird.

```
@PersistenceContext
EntityManager em;

@Transactional
public void departmentAnlegenBeispiel() {

    // Department erstellen
    Department dep = new Department();
    dep.setName("Sales");
    dep.setCity("Paderborn");

    // Mitarbeiter erstellen
    Employee emp = new Employee();
    emp.setName("Peter Müller");
    emp.setJob("Salesman");
    emp.setSalary(7000f);
    dep.addEmployee(emp); // und hinzufügen

    // Mitarbeiterin erstellen
    emp = new Employee();
    emp.setName("Andrea Meier");
    emp.setJob("Saleswoman");
    emp.setSalary(8000f);
    dep.addEmployee(emp); // und hinzufügen

    // persistieren
    em.persist(dep);
}
```

Listing 4: Persistieren von Objekten

```
// Department über Id laden
Department dep = em.find(Department.class, "Sales");

// Department suchen
Query q = em.createQuery(
    "select d from Department d where name = :name",
    Department.class);
q.setParameter("name", "Sales");
List<Department> depList = q.getResultList();
```

Listing 5: Laden oder Suchen von Entities

```
@Transactional
public void increaseDepartmentSalary(String depName,
                                     Float percent,
                                     Float maxSal) {
    Department dep = em.find(Department.class, depName);
    for (Employee emp: dep.getEmployees()) {
        Float newSal = emp.getSalary() * (1f+(percent/100));
        if (newSal <= maxSal) {
            emp.setSalary(newSal);
        }
    }
}
```

Listing 6: Verändern von Entities

Mit persistierten Daten arbeiten

Um ein Entity aus der Persistierung wieder zu restaurieren, kann man ein Entity anhand seiner Id finden oder über einen Query suchen (*siehe Listing 5*).

Auf den ersten Blick könnte man den Query mit SQL verwechseln. Das ist auch gewollt. In einem solchen Query wird die Sprache JPQL (Java Persistence Query Language), die stark an SQL angelehnt ist, verwendet. In *Listing 5* wird von Department selektiert. Dabei ist nicht die Tabelle, sondern die Klasse (oder konkret das Entity) gemeint, die mit `d` „gealiast“ wird und durch die `d` (also das Entity Department) komplett als Liste zurückgegeben wird. Queries werden allerdings normalerweise nicht wie in

Listing 5 als String übergeben, sondern als NamedQuery per Annotation in der Klasse des Entitäts definiert. Das hat zum einen den Vorteil, dass der Query schon zur Compile-Zeit geparkt und auf syntaktische Korrektheit geprüft wird, und zum anderen, dass er an einer zentralen Stelle gehalten wird. Statt `em.createQuery()` wird dann `em.createNamedQuery()` aufgerufen.

Dem geschulten Auge ist sicherlich auch aufgefallen, dass weder in *Listing 4* noch in *Listing 5* der EntityManager über die Employees informiert wurde, die als Liste im Department vorhanden sind. Dass sie mit persistiert wurden, liegt an CascadeType.ALL bei der @OneToMany-Annotation (siehe *Listing 3*). Sie bewirkt, dass unter anderem die Unterentitäten mit persistiert, aber auch gelöscht etc. werden.

Spannend ist auch, dass beim Restaurieren des Departments aus der Persistenz wirklich nur das Department geladen wird. Für die Liste der Employees wird von JPA ein Proxy-Objekt eingesetzt, dass beim Ansprechen schnell die notwendigen Unterentitäten nachlädt. Der Entwickler braucht sich also auch hier um nichts zu kümmern.

Wenn die Entitäten über den EntityManager geladen werden, egal ob über `find()` oder einen Query, werden sie vom EntityManager überwacht. Das bedeutet, dass die Entitäten bei Änderungen am Ende der Transaktion gespeichert werden.

Listing 6 zeigt sowohl das Lazy-Loading als auch das Speichern von Änderungen.

Auffällig ist, dass der Code, der sich in *Listing 6* mit der Persistierung beschäftigt, gerade mal zwei Zeilen darstellt: `@Transactional` und die Zeile mit dem `em.find()`. Als Entwickler kann ich mich also auf die Fachlichkeit konzentrieren.

Es gibt noch viele weitere Features von JPA, die dem Entwickler die Arbeit mit persistenten Daten erleichtern. An dieser Stelle soll noch auf die Möglichkeit eingegangen werden, Entitäten gemeinsam zu persistieren. In Java sind Objekte mit vielen Eigenschaften verpönt, da sie dadurch unübersichtlich werden. Es werden lieber Unterobjekte angelegt. Beispielsweise besteht der Name einer Person zumindest aus Titel, Vorname und Nachname. Dennoch ist er so stark an die Person gebunden, dass wohl nur wenige auf die Idee kommen würden, den Namen einer Person in einer eigenen Tabelle halten zu wollen. Es wird auch im Normalfall kein

Geschwindigkeitsvorteil zu erwarten sein, wenn sich die Daten verringern und nur bei Bedarf der Name nachgeladen wird. Ein weiterer Punkt für das Auslagern ist, dass eine Namens-Klasse wiederverwendet werden kann. Ein Kunde, Verkäufer, Verantwortlicher etc. hat einen Namen. Wenn ein Namensobjekt verwendet wird, werden alle die gleichen Attribute haben und kein Entwickler braucht darüber nachzudenken, wie die Attribute im konkreten Fall heißen.

Um ein Unterentität mit einer so starken Bindung abzudecken, wird der Dependent beziehungsweise das Embeddable Entity angeboten.

In *Listing 7* wird der Employee dahingehend erweitert, dass der Zeitraum seiner Einstellung (`employedSince`, `hiredUntil`) durch das @Embedded Entity `employmentPeriod` abgebildet wird.

Hinweis: Durch die Angabe `@Temporal(TemporalType.DATE)` weiß JPA, dass kein Uhrzeitanteil persistiert werden soll. Bei einer MySQL-Datenbank wird durch die Angabe beispielsweise ein anderer Spaltentyp verwendet.

In der Datenbank befinden sich die Spalten `employedSince` und `hiredUntil` in der Tabelle `Employee` genauso wie `name`, `job` und `salary`.

Man muss wissen, was man tut

Obwohl JPA dem Entwickler sehr viel abnimmt, muss man dennoch wissen, wie JPA funktioniert.

Beispielsweise entstehen schnell verwaiste Tabelleneinträge, wenn Unterentitäten gelöscht werden. Um in unserem Beispiel einen Employee zu löschen, reicht es nicht, den Listeneintrag zu entfernen. Auf den ersten Blick sieht die Methode `removeEmployee()` sehr vielversprechend aus. Wenn man mit ihr ein Entity löscht und danach das Department neu lädt, ist auch der Employee verschwunden. In der Datenbank bleibt der Tabelleneintrag jedoch stehen, lediglich der Verweis auf das Department ist null. Das hat damit zu tun, dass eigentlich nur die Verbindung zwischen den Objekten gekappt wurde. Aus Sicht des EntityManager wird der Employee mit `removeEmployee()` nur verändert, er bleibt aber dem EntityManager weiter bekannt. Er bleibt im sogenannten EntityCache. Um ein Entity zu löschen, muss es wie in *Listing 8* explizit mit `em.remove()` aus dem Cache entfernt werden.

Hinweis: Das Konstrukt `new ArrayList<Employee>(dep.getEmployees())` ist notwendig, da wir aus einem Set Elemente löschen wollen, während darüber iteriert wird.

```
@Embeddable
public class EmploymentPeriod {
    @Temporal(TemporalType.DATE)
    Date employedSince;
    @Temporal(TemporalType.DATE)
    Date hiredUntil;

    public Date getEmployedSince() {
        return employedSince;
    }
    ...
}

@Entity
public class Employee {
    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    private Long id;
    private String name;
    private String job;
    private Float salary;
    @Embedded
    private EmploymentPeriod employmentPeriod;
    @ManyToOne
    private Department department;
    ...
}
```

Listing 7: Verändern von Entitäten

Natürlich ist es auch notwendig, JPA zu verstehen, wenn man auf die Performance achten muss. Natürlich ist es möglich, über NativeQueries wie bei JDBC direkte Datenbank-Abfragen in SQL zu erstellen. Aber das sollte wirklich nur im absoluten Notfall passieren. Am häufigsten stolpert man bei der Performance über das Lazy-Loading-Verhalten. Wenn wir uns *Listing 6* anschauen, werden zum Holen der Entities von JPA zwei Selects ausgeführt. *Listing 9* zeigt die beiden generierten Selects.

Zwei Selects kann man noch verschmerzen, die Anzahl kann sich allerdings schnell potenzieren, wenn der Objektbaum noch eine 3. Ebene bekommt (beispielsweise Locations → Departments → Employees).

Deshalb ist es möglich, JPA anzuweisen, mehrere Ebenen direkt zu selektieren. Das kann permanent bei der Definition der Entities in der @OneToMany-Annotation über fetch = FetchType.EAGER geschehen. Es ist auch möglich, in einem Query, bei dem die Entities explizit über join fetch miteinander verknüpft werden, oder über einen EntityGraph, der dem Query oder find() als Property übergeben wird, beim Laden explizit anzugeben, wann Lazy-Loading verwendet werden soll. Die Verwendung eines EntityGraph ist sehr elegant, da beim gleichen Query oder find() je nach Bedarf unterschiedliche EntityGraphs verwendet werden können. *Listing 10* zeigt die Methode aus *Listing 6* (increaseDepartmentSalary()), um einen EntityGraph erweitert. Durch diese Erweiterung generiert JPA nur noch einen Select, um die Daten zu holen.

Hinweis: Häufig werden in Beispielen von Entities Listen statt Sets für Unterobjekte verwendet. Das macht an vielen Stellen den Code einfacher. Aber gerade, wenn das Lazy-Loading ausgehebelt werden soll, entstehen unerwartete Ergebnisse ab einer Objekthierarchie mit 3 Ebenen.

Datenstrukturen

Zum Abschluss noch ein Hinweis zu den Folgen, wenn mit JPA gearbeitet wird:

Durch JPA wird in Java mit der Datenbank völlig anders umgegangen. Der Fokus liegt auf der Verwendung der Daten, nicht auf der Speicherung. Dadurch erge-

```
@Transactional
public void deleteHighSalaryEmployees(String deptName, Float maxSalary) {
    Department dep = em.find(Department.class, deptName);
    for (Employee emp:new ArrayList<Employee>(dep.getEmployees()))
    {
        if (emp.getSalary() > maxSalary) {
            dep.removeEmployee(emp);
            em.remove(emp);
        }
    }
}
```

Listing 8: Löschen von Entities

```
Hibernate:
select
    department0_.name as name1_0_0_,
    department0_.city as city2_0_0_
from
    Department department0_
where
    department0_.name=?

Hibernate:
select
    employees0_.department_name as departme7_1_0_,
    employees0_.id as id1_1_0_,
    employees0_.id as id1_1_1_,
    employees0_.department_name as departme7_1_1_,
    employees0_.employedSince as employed2_1_1_,
    employees0_.hiredUntil as hiredUnt3_1_1_,
    employees0_.job as job4_1_1_,
    employees0_.name as name5_1_1_,
    employees0_.salary as salary6_1_1_
from
    Employee employees0_
where
    employees0_.department_name=?
```

Listing 9: Zwei Selects durch Lazy-Loading

```
private Map<String, Object> getFetchAllEntityGraphProperty() {
    EntityGraph<Department> fetchAll =
        em.createEntityGraph(Department.class);
    fetchAll.addAttributeNodes("employees");
    Map<String, Object> properties = new HashMap<>();
    properties.put("javax.persistence.fetchgraph", fetchAll);
    return properties;
}

@Transactional
public void increaseFastDepartmentSalary(String depName,
                                         Float percent,
                                         Float maxSal) {
    Department dep = em.find(Department.class,
                             depName,
                             getFetchAllEntityGraphProperty());
    for (Employee emp: dep.getEmployees()) {
        Float newSal = emp.getSalary() * (1f+(percent/100));
        if (newSal <= maxSal) {
            emp.setSalary(newSal);
        }
    }
}
```

Listing 10: find() mit EntityGraph

ben sich auch Änderungen an den Datenstrukturen. Bei der Modellierung der Entities sollte auf Folgendes geachtet werden:

- Kleine, an der Nutzung ausgerichtete, fachliche Strukturen/Hierarchien aufbauen
- Der Kopf der Struktur kann einen fachlichen Schlüssel bekommen, die darunterliegenden Strukturen nur einen technischen.
- Objektstrukturen werden aufgeteilt, Datenstrukturen zusammengefasst (Stichwort Embeddable)
- Bei großen Datenmengen, z.B. BLOBs, Unterstrukturen anlegen

Noch ein allerletzter Hinweis: Um mit JPA etwas zu spielen, stelle ich ein

Demo-Projekt unter <https://gitea.apps.soco.team-pb.de/wb/DoagDemoJPA> zur Verfügung.

Quellen

- [1] Ansonsten benötigt JPA eine Zwischentabelle, aber wer will das schon bei einer 1:n-Beziehung?

Über den Autor

Wolf G. Beckmann hat selbst über 10 Jahre PL/SQL entwickelt, bis er als Teamleiter in den Consulting-Bereich gewechselt ist, der primär mit Java arbeitet. Fast alle Projekte im Consulting-Bereich beschäftigen sich mit dem Wechsel aus einer von PL/SQL hin zu einer von Java getriebenen Welt. Auch

aus eigener Erfahrung legt er in den Projekten neben der Technik großen Wert darauf, wie die Mitarbeiter bei so einem Wechsel mitgenommen werden.



Wolf G. Beckmann
wb@team-pb.de

Bessere APIs dank GraphQL?

Markus Lohn, esentri

GraphQL wurde ursprünglich von Facebook entwickelt und im Jahre 2015 als Open Source zur Verfügung gestellt. In der Zwischenzeit ist ein regelrechter Hype um die Spezifikation entstanden, ja sogar das Ende von REST wurde ausgerufen. GraphQL ist eine interessante Möglichkeit, APIs zu definieren. Ferner bietet es Lösungen für eine Reihe von Herausforderungen, die typischerweise im Bereich REST auftreten können. Allerdings gibt es nicht nur GraphQL: HAL, JSON:API etc. sind weitere Möglichkeiten, um robuste und flexible APIs zu definieren. In diesem Artikel stelle ich GraphQL vor und versuche, die Unterschiede darzustellen.

APIs haben häufig das Problem mit dem Absetzen von vielen Anfragen an einen Server, um alle benötigten Daten für ein Geschäftsobjekt zu laden. Als Beispiel dient die Anzeige einer Rechnung auf einem mobilen Endgerät oder einer Webseite (siehe *Abbildung 1*). Zunächst werden die Basisdaten einer Rechnung wie Rechnungsnummer, Rechnungsdatum, Gesamtbetrag sowie die Referenzen auf den Kunden etc. angefragt. Im nächsten Schritt müssen die Details zu einem Kunden erfragt werden. Danach erfolgt die

Abfrage der Rechnungspositionen. Für alle verfügbaren Rechnungspositionen werden unter Umständen weitere Abhängigkeiten, zum Beispiel zu den Produktdaten vom Server, abgeholt. Somit ergibt sich für dieses einfache Beispiel eine Vielzahl von Serveranfragen, die allein für sich nicht komplex sind, aber dennoch eine unnötige Anfragemenge bei einem Server erzeugen.

Generell musste die Anbindung von mobilen Applikationen verbessert werden. Die Entwicklung zu GraphQL durch

Facebook startete 2012. Im Jahr 2015 erfolgte die Bereitstellung der Spezifikation als Open Source. Für die Weiterentwicklung der GraphQL-Spezifikation zeichnet die GraphQL Foundation verantwortlich. In der Foundation sind neben Facebook weitere namhafte Unternehmen wie Airbnb, AWS, Twitter oder IBM aktiv.

GraphQL definiert eine Abfragesprache für APIs. Auch wenn es der Name vermuten lässt, hat GraphQL nichts mit Graphendatenbanken gemein! Der Name resultiert aus der Tatsache, dass

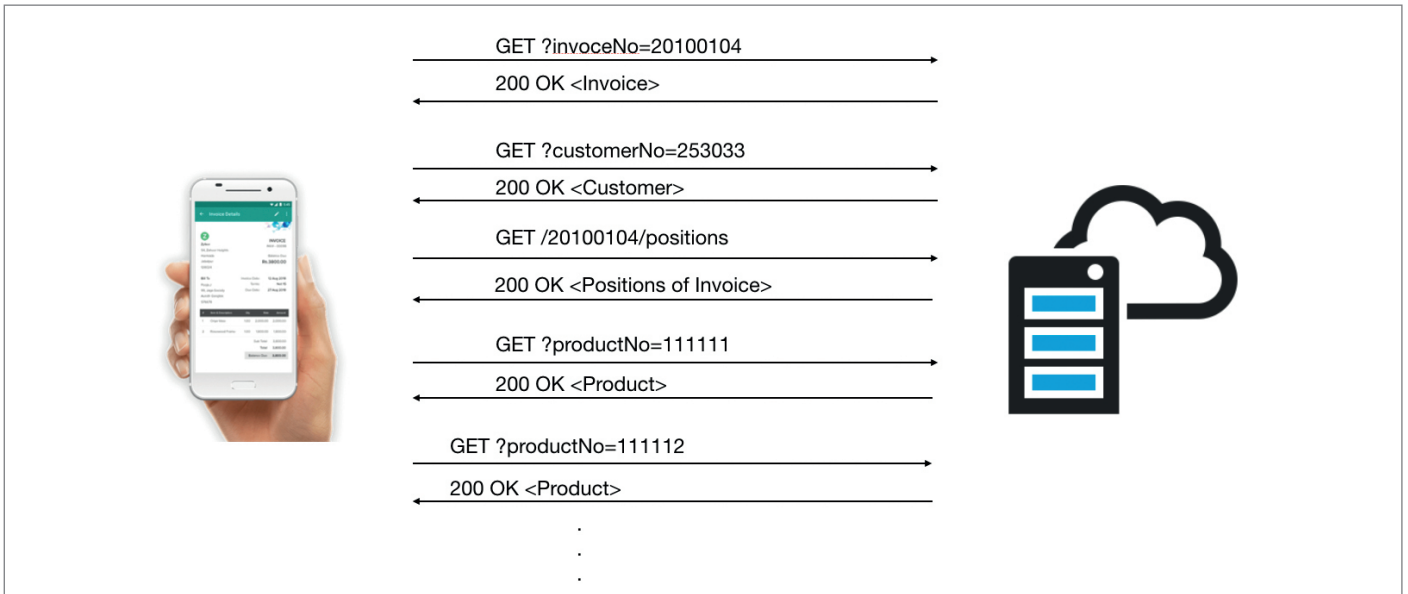


Abbildung 1: Typisches Kommunikationsmuster zwischen Client und Server durch APIs (Quelle: Markus Lohn)

komplexe Objektstrukturen durch eine GraphQL-Abfrage erzeugt werden können. Vergleicht man die Konzepte von GraphQL und relationalen Datenbanken mit der Abfragesprache SQL, gibt es viele Analogien. Ein wesentlicher Gesichtspunkt ist die Unabhängigkeit von einer Programmiersprache oder konkreten Technologie. GraphQL beschreibt Konzepte zur Definition, Abfrage und Manipulation von Daten. In welcher Technologie oder Programmiersprache die Spezifikation letztendlich umgesetzt wird, ist nicht relevant. Es existieren eine Vielzahl von Implementierungen, zum Beispiel für Java, C#, JavaScript etc.

Herausforderungen beim Design von APIs

Beim Design von APIs müssen viele Fragestellungen beachtet werden. Diese Fragestellungen lassen sich in verschiedene Teilbereiche gruppieren, wie etwa Sicherheit, Performance, Robustheit. Insbesondere Suchen/Filtering, Pagination, Umgang mit Beziehungen, hierarchische Objektstrukturen sowie Ressourcen-Identifikation sind in diesem Zusammenhang zu nennen (siehe Abbildung 2). GraphQL fokussiert sich in der Spezifikation genau für die letztgenannten Herausforderungen.

GraphQL-Spezifikation im Detail

In der Spezifikation werden drei Bestandteile beschrieben: Queries, Mutations, Subscriptions (siehe Abbildung 3). Der Bereich Queries beschreibt mögliche Formen zur Abfrage von Daten, Mutations definiert das Manipulieren von Daten und Subscriptions beschäftigt sich mit der Aktualisierung von Daten zwischen Client und Server in Echtzeit.

Damit jedoch Daten über GraphQL verändert und abgefragt werden können, muss zunächst einmal ein sogenanntes Schema definiert werden. Ein Schema be-

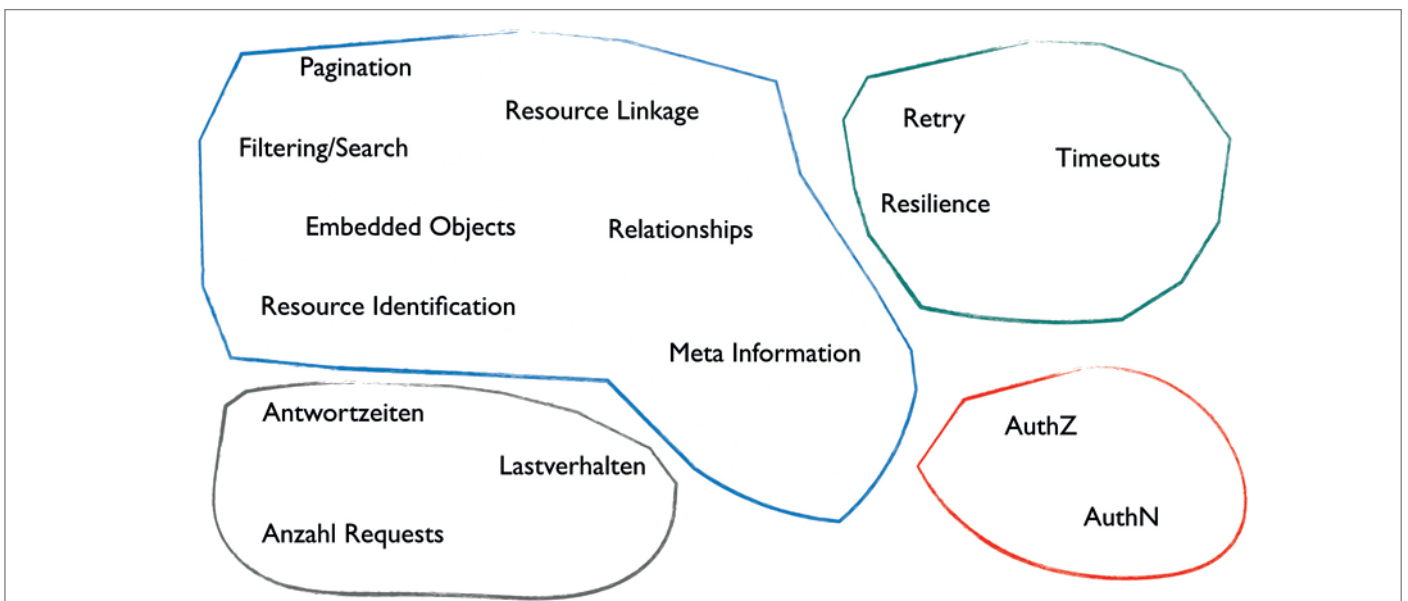


Abbildung 2: Herausforderungen beim Design von APIs (Quelle: Markus Lohn)

inhalte die Definition von Datentypen. Neben vordefinierten Datentypen können auch eigene Datentypen hinzugefügt werden. Ferner werden in einem Schema Queries und Mutations festgelegt. Ein GraphQL-Schema wird in textueller Form beschrieben und entspricht in der Syntax einem JSON-Dokument. Was nicht in einem Schema definiert wurde, kann auch durch den Client nicht abgefragt werden. Abfragen, die der Server nicht im Schema findet, werden mit einem Fehler an den Client beantwortet. Die Vorgehensweise lässt sich sehr gut mit dem Design einer relationalen Datenbank mit Tabellen und Views vergleichen. SQL-Abfragen können nur gegen existierende Tabellen und Views durchgeführt werden. Das verfügbare Schema kann gezielt abgefragt werden (siehe Abbildung 4).

Die Erstellung eines GraphQL-Schemas kann mit erheblichem Aufwand verbunden sein. Wenn das Schema auf Basis existierender Strukturen aufgebaut werden muss, zum Beispiel JPA-Objekte in Java, bietet sich die Nutzung eines Gene-



Abbildung 3: GraphQL-Bausteine (Quelle: Markus Lohn)

rators an. Je nach eingesetzter Technologie existieren bereits Generatoren für ein GraphQL-Schema. Eine Evaluierung lohnt sich auf jeden Fall.

Queries werden in einer JSON-ähnlichen Notation an den Server übertragen. In einer Query werden die Objekte und Felder, die in der Antwort erwartet werden, angegeben. Der Client kann dabei die benötigten Felder, ähnlich wie bei einem SELECT-Statement für die Datenbank, angeben. Die Queries an einen Gra-

phQL-Server bleiben immer stabil, solange keine inkompatiblen Änderungen am Schema durchgeführt werden.

Im Beispiel werden Daten für alle verfügbaren Orders abgefragt (siehe Abbildung 5). Die Attribute einer Order, zum Beispiel status, total, orderDate etc., sind frei wählbar. Der große Vorteil von GraphQL besteht darin, Server-Roundtrips zu vermeiden. In diesem Beispiel wird das sehr gut am Objekt Customer deutlich. Ferner können auch für den Customer die benötigten Attribu-

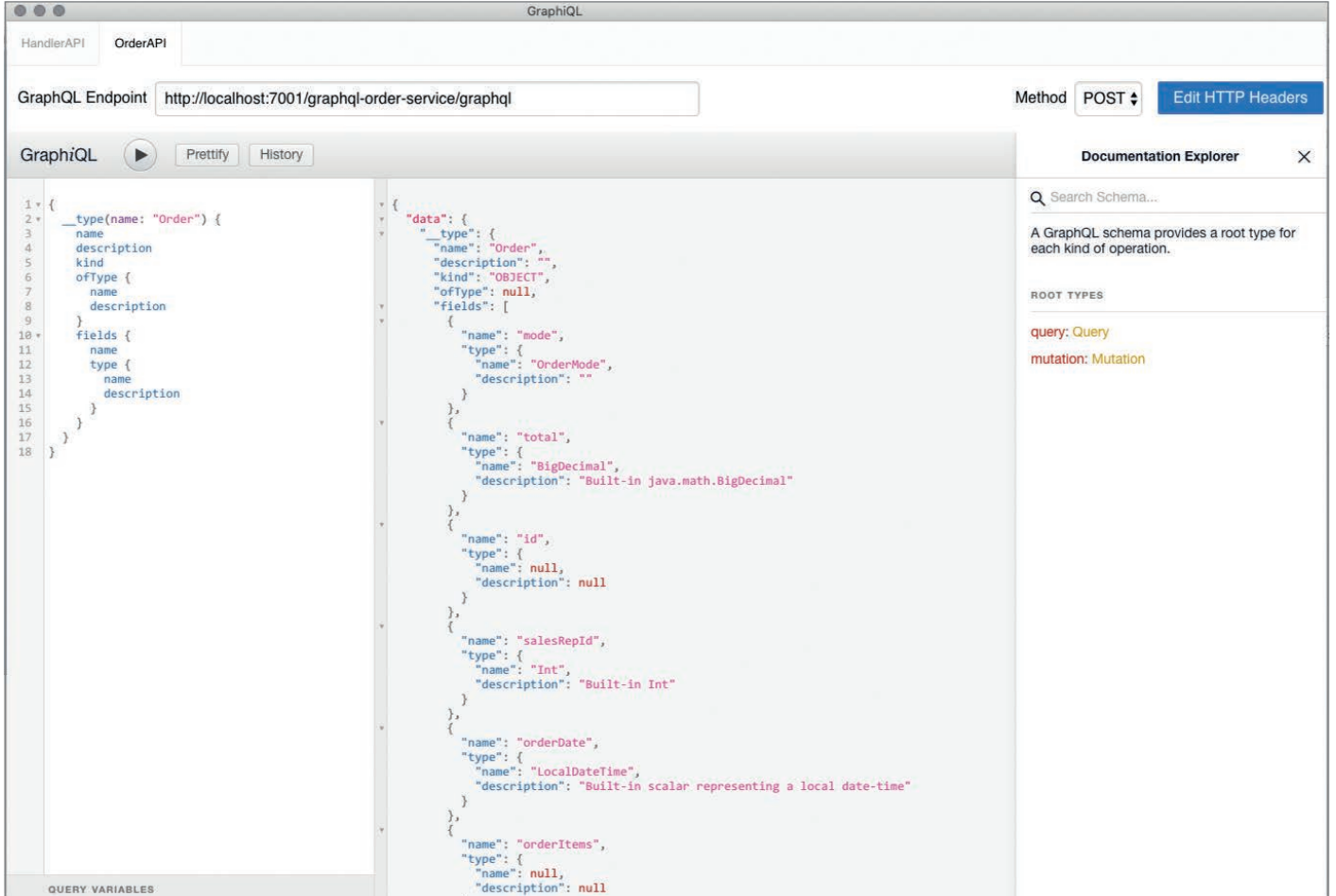


Abbildung 4: Schema-Abfrage für eine Order (Quelle: Markus Lohn)

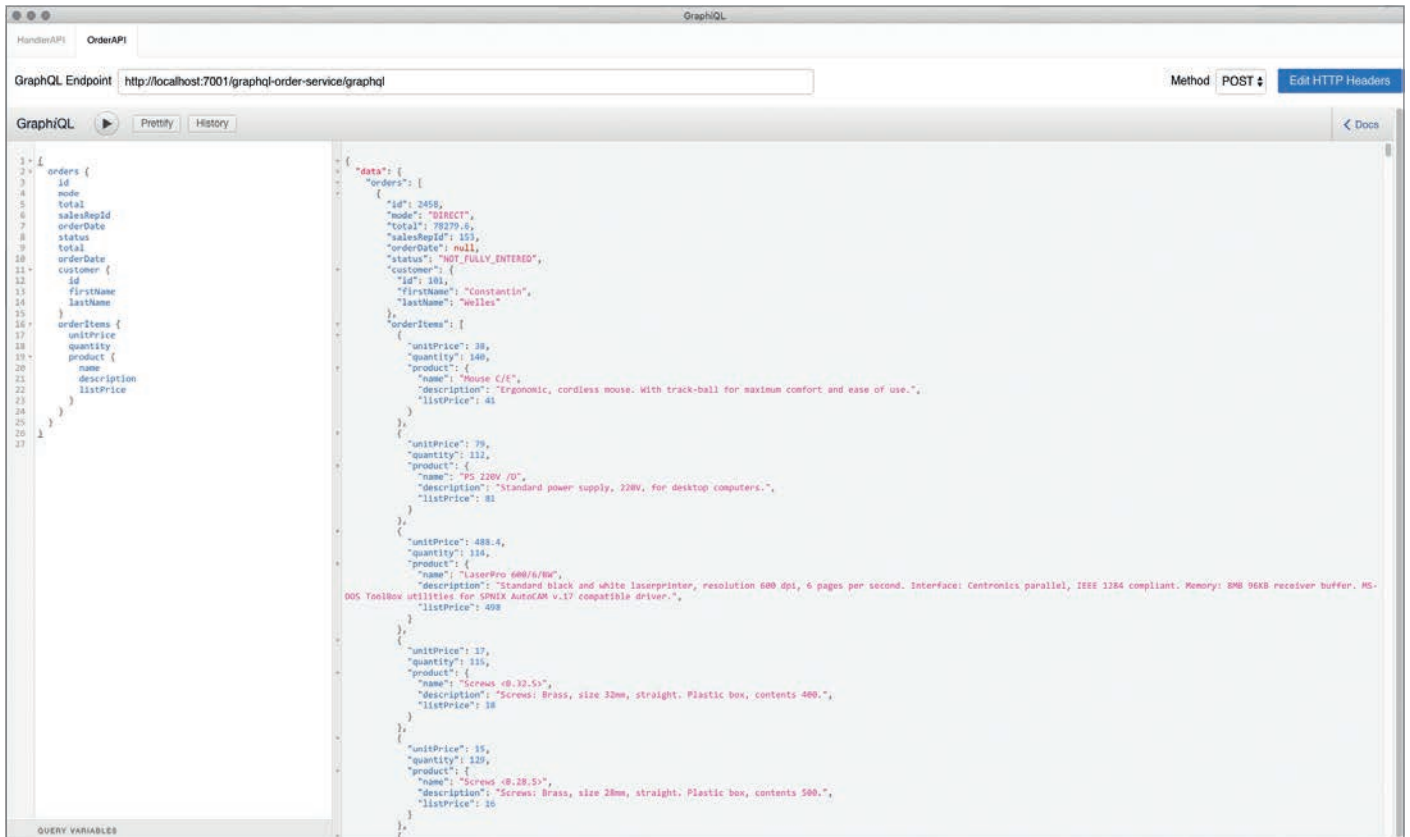


Abbildung 5: Query und Response zur Abfrage von Orders (Quelle: Markus Lohn)

te frei gewählt werden. Zusätzlich werden die Bestellpositionen (orderItems) und die gewünschten Attribute in der Antwort vom Server geliefert. Durch diesen Mechanismus können komplexe Objektstrukturen mit nur einem Request abgefragt werden.

Queries können durch verschiedene Möglichkeiten auf Client-Seite dynamisch aufgebaut werden. Leider können nicht alle Varianten im Detail in diesem Artikel erläutert werden. Nachfolgend eine Auflistung mit kurzer Beschreibung:

- Aliases
Ein Alias ist ein eindeutiger Name, mit dem eine Query referenziert werden kann.
- Fragments
Fragments ermöglichen die Wiederverwendbarkeit von Query-Bestandteilen. Beispielsweise könnten die Basisattribute zu einer Order-Query in verschiedenen anderen Queries wiederverwendet werden.
- Directives
Das dynamische Bauen von Queries kann mit sogenannten Directives auf Client-Seite gesteuert werden. Dadurch werden aufwendige String-Operationen zum Bauen von Queries auf Client-Seite

vermieden. Die Spezifikation definiert zwei mögliche Directives: include und skip. Mit diesen Keywords kann gezielt gesteuert werden, welche Bestandteile einer Query gegebenenfalls ausgeführt oder nicht berücksichtigt werden.

In GraphQL können Daten über Mutations manipuliert werden. Im Beispiel in *Abbildung 6* wird eine neue Order angelegt. In einer Mutation sollten die minimal erforderlichen Daten bereitgestellt werden, um das betreffende Geschäftsobjekt anzulegen. Typischerweise wird das mit einem sogenannten Input Type implementiert. Der Input Type definiert alle minimal notwendigen Daten zum Verändern eines Geschäftsobjektes. Somit entfällt die Notwendigkeit, das Geschäftsobjekt in allen möglichen Ausprägungen mitzugeben. Eine Mutation kann sowohl zum Anlegen eines neuen Objektes als auch zum Verändern eines bestehenden Objektes verwendet werden. Letztendlich entscheidet die konkrete Implementierung, wie diese Unterscheidung umgesetzt wird. Zusätzlich kann in einer Mutation eine Query definiert werden, die die gewünschte Repräsentation des geänderten Objektes an den Client liefert.

Alternativen für Restful APIs

Auch im REST-Umfeld hat man versucht, typische Problemstellungen in APIs durch Entwicklung von Spezifikationen zu lösen. Dabei sind einige interessante Varianten entstanden, die sicherlich auch als Alternativen zu GraphQL in Betracht gezogen werden können:

- JSON-LD
Wird durch das W3C-Konsortium gepflegt und liegt aktuell in der Version 1.0 (2014) vor. Es geht in diesem Standard um die Implementierung des Linked-Data-Konzeptes mit JSON-Dokumenten. Ferner definiert der Standard einen sogenannten Context, dem Eigenschaften eines JSON-Dokumentes zugeordnet werden. Vergleichbar ist dieses Konzept mit XML-Namensräumen. Der Standard ist insofern interessant, als sich bestehende Implementierungen sehr leicht um JSON-LD-Konzepte erweitern lassen.
- Hypertext Application Language (HAL)
HAL definiert einen eigenen Content-Type und versucht, Hypermedia-APIs auf Basis von JSON- oder XML-Dokumenten zu beschreiben. Zurzeit liegt der Standard als Internet Draft (Work

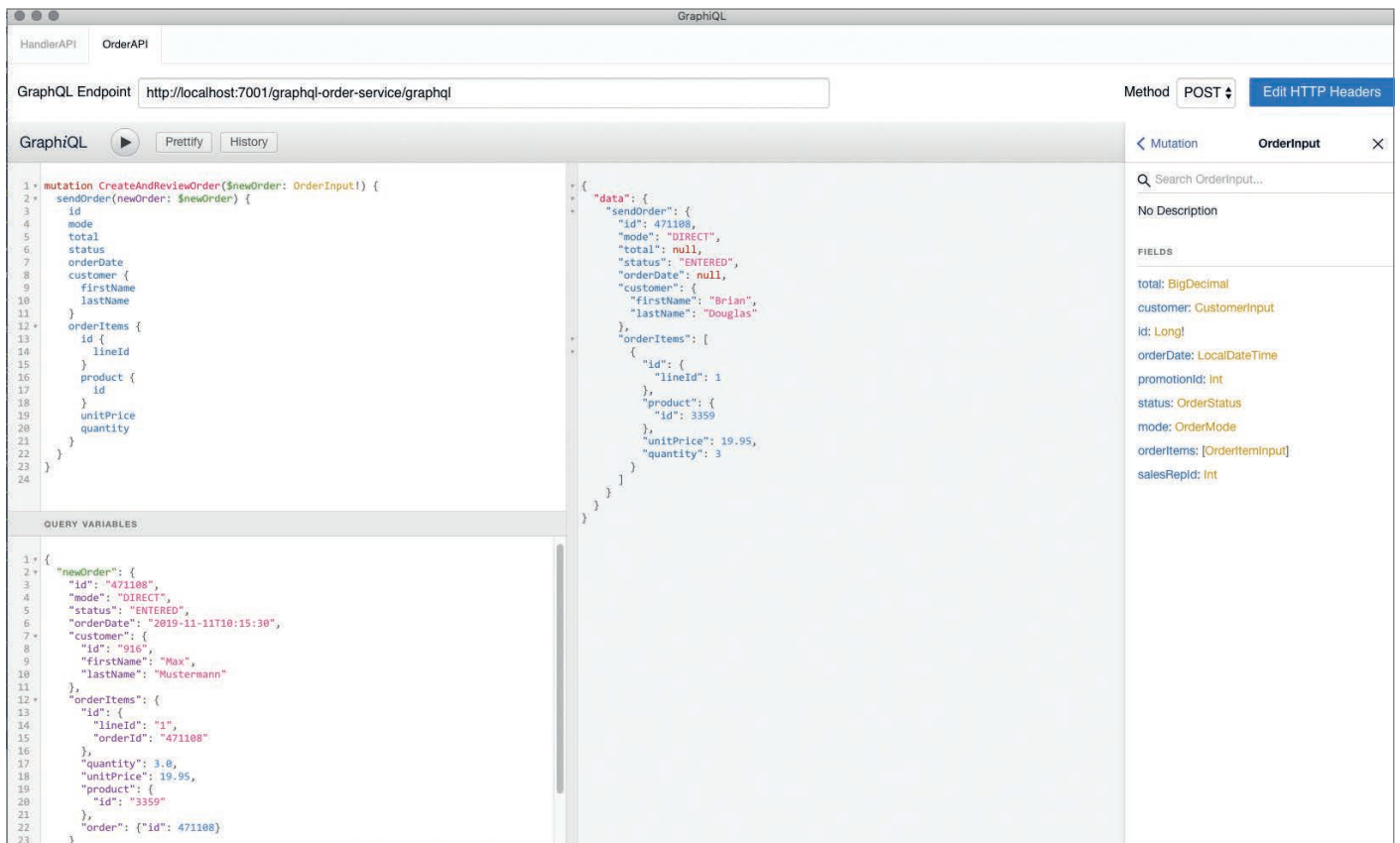


Abbildung 6: Neue Order über GraphQL anlegen (Quelle: Markus Lohn)

in Progress) bei der Internet Engineering Task Force (IETF) vor.

- JSON-API
Die Spezifikation liegt in der Version 1.0 aus dem Jahre 2015 vor. Sie definiert ebenfalls einen eigenen Content-Type und vereint viele Konzepte von HAL und GraphQL. Somit können hier die Vorteile beider Konzepte für REST-APIs genutzt werden. Für die Spezifikationen existieren verschiedene Implementierungen in unterschiedlichen Technologien.

Fazit

Häufig wird GraphQL als der Nachfolger von Restful APIs gehandelt. Sogar das Ende von REST wurde schon vorhergesagt. In der Tat bietet GraphQL interessante neue Möglichkeiten, APIs zu gestalten. Insbesondere die Gestaltung von Ad-hoc-Abfragen oder Reports gestaltet sich viel einfacher als bei einem REST-API. Hypermedia-APIs im eigentlichen Sinne lassen sich mit GraphQL jedoch so nicht umsetzen. Die Spezifikation beschäftigt und behebt einige Nachteile von Restful APIs, beispielsweise die Abfrage komplexer Datenstrukturen unter Vermeidung von Server-Roundtrips. Allerdings wer-

den nicht alle technisch relevanten Aspekte in der Spezifikation berücksichtigt, wie etwa Caching, Content-Negotiations etc. Solche Mechanismen finden in Restful-APIs Anwendung und müssten in einer GraphQL-Implementierung aufwendig integriert werden. Zwischenzeitlich gibt es aber eine Reihe von Alternativen im Umfeld von REST, um ähnliche Konzepte wie bei GraphQL in einem RESTful API zu nutzen. Somit werden aus meiner Sicht RESTful APIs nicht durch GraphQL abgelöst. Vielmehr würde ich GraphQL als weitere Option beim Design und der Implementierung eines API ansehen. Der konkrete Anwendungsfall und das Abwägen der Vor- und Nachteile sollte dann zu einer Entscheidung für oder gegen GraphQL führen.

Quellen

- [1] GraphQL-Spezifikation: <https://graphql.org/>
- [2] GraphQL Foundation: <https://foundation.graphql.org/>
- [3] GraphQL-Java-Beispiel: <https://github.com/markuslohn/graphql-order-service>
- [4] GraphQL-Java-Implementierung: <https://github.com/graphql-java/graphql-java>
- [5] JSON-LD-Spezifikation: <https://json-ld.org/>
- [6] HAL-Spezifikation: http://stateless.co/hal_specification.html
- [7] JSON-API-Spezifikation: <https://jsonapi.org/>

Über den Autor

Markus Lohn arbeitet als Principal Consultant bei der esentri AG und leitet den Geschäftskreis Integration. Seit mehr als 20 Jahren unterstützt er Kunden in anspruchsvollen IT-Projekten, um die fachlichen und technischen Ziele zu erreichen.



Markus Lohn
markus.lohn@esentri.com



Konsolidierung und Resource Management

Stefan Seck, Logicalis

Konsolidierung kommt aus dem Lateinischen. Consolidare bedeutet so viel wie „festmachen“ oder „stark machen“. Der Antrieb zur Konsolidierung kann aus verschiedenen Faktoren hervorgehen. Vor allem die Vereinheitlichung, also die Standardisierung des Betriebes ist hier zentral. Die Optimierung der Systemlandschaft, die Wartung und Pflege der Systeme und auch die Compliance nach DSGVO werden vereinfacht. Konsolidierung und Standardisierung funktioniert aber nur bei einem vernünftigen Management der zur Verfügung stehenden Ressourcen.

Um einen Zoo an Systemen, Servern oder Datenbanken, die in unterschiedlichen Versionen und Konfigurationen vorgehalten werden, in den Griff zu bekommen, ist die Standardisierung und Konsolidierung ein wichtiger Weg.

Warum Standardisierung?

Es gibt gute Gründe für eine Aufteilung von Systemen, wie beispielsweise:

- Service-Level-Agreements (SLAs) und vertragliche Einschränkungen
- Server-Beschränkungen, wie CPU- und Speichergrenzen
- Datenbank-Limitierungen
- Datenbank-Kompatibilität, zum Beispiel Nutzung gemeinsamer Optionen
- Datenbankgrößen
- Priorisierungen

Gleichwohl ist es ebenso wichtig, die Standardisierung der Systemplattformen und

die Vereinheitlichung der Sicherheitsstandards voranzutreiben.

Eine standardisierte Umgebung vereinfacht zuallererst die Administration der Systeme. Es müssen nicht unterschiedliche Versionen oder Editionen gewartet werden. Zudem wird die Bereitstellung neuer Systeme beschleunigt.

Etablierte SLAs helfen bei der Einteilung verschiedener Systeme in Gruppen. Denkbar sind hier folgende SLAs oder Gruppen:

- Platinium
- Gold
- Silber
- Bronze

Jede dieser Gruppen wird mit einer anderen Architektur aufgebaut, die zuverlässig einen bestimmten Service-Level bei niedrigen Kosten und Komplexität erreicht. Diese Architekturen adressieren alle Arten von ungeplanten Ausfällen, einschließlich Komponentenausfall, System- und Standortausfällen sowie geplanten Ausfällen aufgrund von Wartung, Migrationen oder anderen Zwecken.

Bronze eignet sich für Datenbanken, bei denen ein einfacher Neustart oder eine Wiederherstellung aus einem RMAN-Backup ausreichend sind.

Silber bietet mit Clustering, wie etwa Oracle Real Application Clusters (RAC), eine zusätzliche Verfügbarkeit. Das hilft Datenbanken, die bei einem Ausfall der Datenbankinstanz oder eines Servers sowie bei vielen Arten geplanter Wartung minimale oder keine Ausfallzeiten erfordern.

Gold erhöht weiter das Serviceniveau für geschäftskritische Anwendungen, indem Ausfallzeiten für alle ungeplanten Ausfälle und wichtigen Upgrades der Datenbanksoftware reduziert werden. Zu den vorgenannten Architekturen werden Replikationstechnologien hinzugefügt, die beispielsweise Oracle Data Guard verwenden, um eine oder mehrere Duplikate einer Produktionsdatenbank zu synchronisieren und so die Verfügbarkeit zu gewährleisten. Far Sync erweitert die Verfügbarkeit, um die Unterstützung für lange Strecken zwischen Produktions- und Disaster-Recovery-Standorten zu gewährleisten.

Platinium fügt zum Beispiel Oracle GoldenGate hinzu, um bidirektionale Funktionen bereitzustellen, die Upgrades und Migrationen ohne Ausfallzeiten ermöglichen.

Auch wenn dies nur ein kleiner Teil der Vorteile ist, zeigen schon diese Beispiele, dass mit Standardisierung viel erreicht werden kann.

Ein weiterer wichtiger Punkt ergibt sich aus diesen Bemühungen, denn der Weg in die „Cloud“ ist mit standardisierten Systemen und Datenbanken viel einfacher. Genauer gesagt, standardisierte Systeme sind eine Voraussetzung auf dem Weg in die Cloud.

Warum Konsolidierung?

Neben der Standardisierung ist ein weiterer wichtiger Baustein die Konsolidierung der IT-Landschaft. Unter Konsolidierung werden die Zusammenfassung und Vereinheitlichung von zum Beispiel Servern und Datenbanken verstanden. Dabei stehen wichtige Punkte im Fokus:

- Verringerung der Komplexität
- Reduzierung der Kosten
- Optimierung der Lizenznutzung
- Optimierung der Ressourcennutzung

Unter anderem mit diesen Punkten wird ein einfacheres Management der Systeme erreicht. Die mögliche Reduzierung auf wenige standardisierte Systeme ermöglicht zum Beispiel eine einheitliche und somit auch einfachere Bereitstellung von Datenbanken. Ebenso wird auch das Patchen der Systeme einfacher und somit auch schneller durchgeführt werden können.

Ein wichtiger Faktor bei der Konsolidierung von Datenbanken ist die Kostensenkung. Die Kosteneinsparungsvorteile der Konsolidierung gelten sowohl in traditionellen Rechenzentren vor Ort als auch bei der Nutzung von Cloud-Services. Die Konsolidierung von Datenbanken ermöglicht eine höhere Auslastung von On-Premises-Systemen durch eine dynamische, automatische Zuteilung von Ressourcen unter Verwendung der Ressourcenverwaltungsfunktionen der Datenbank. Die vereinfachte Verwaltung von Datenbanken in einer konsolidierten Umgebung beginnt mit Standardisierung und vor allem mit der Reduzierung der Vielfalt der zu wartenden Konfigurationen. Datenbankkonsolidierung ermöglicht es, viele Datenbanken als eine zu verwalten, wodurch die Verwaltung physischer Server und virtueller Maschinen reduziert wird. Die Verwendung von Oracle Multitenant kann bei der Konsolidierung weitere Unterstützung bieten, da die vorhandenen Ressourcen besser genutzt werden können. Die Vereinfachung der Administration ermöglicht es zusätzlich, sich auf die Sicherstellung der Verfügbarkeit von Datenbanken und damit der Applikationen zu konzentrieren. Die Einhaltung der Verfügbarkeitsziele von Datenbanken und Anwendungen ist in einer Konsolidierungsumgebung entscheidend, da sich mehrere Daten-

banken auf derselben Infrastruktur befinden. Die richtige Konfiguration, um Systeme gegen Ausfälle zu härten, ist wichtig. Ebenso wie mehrere Isolations-schichten einzusetzen, um die Auswirkungen von Ausfällen zu reduzieren. So kann eine kontinuierliche Verfügbarkeit bei Wartungsarbeiten, Upgrades oder bei Ausfällen von Infrastrukturkomponenten erreicht werden.

Verbesserte Sicherheit und die Isolierung von Systemen werden bei der Konsolidierung durch eine Reduzierung von Schwachstellen, eine Erhöhung der Standardisierung der Datenbanken und eine Verringerung des Administrationsaufwands für die Sicherung dieser Datenbanken erreicht.

Aber auch die Bereitstellung der erforderlichen Performance von Datenbanken und Applikationen ist ein zentrales Ziel jedes Konsolidierungsprojekts.

Methoden zur Datenbankkonsolidierung

Es gibt verschiedene Möglichkeiten zur Konsolidierung von Datenbanken. So lassen sich Datenbanken auf physischen oder virtuellen Maschinen konsolidieren. Denkbar ist natürlich auch, dass zunächst passende Datenbanken in einzelne Schemata einer Datenbank konfiguriert werden. Virtuelle Maschinen werden beispielsweise verwendet, um eine Isolierung auf der O/S-Schicht zu gewährleisten. Die Möglichkeiten können einzeln oder in Kombination verwendet werden, um Datenbanken zu konsolidieren.

Planung und Konzept

Standardisierung und Konsolidierung funktioniert nicht „out of box“. Um ein entsprechendes Projekt zu starten und vor allem erfolgreich zu beenden, ist es sehr wichtig, ein gutes Konzept und eine gute Planung zu erstellen. Wichtige Fragen müssen geklärt werden

- Wie ist der Workload der einzelnen Applikationen beziehungsweise Services?
- Welche Applikationen beziehungsweise Services können zusammen betrieben werden?
- Existieren Compliance-Vorgaben, die einzuhalten sind?
- Gibt es weitere Regeln?



Das E-3 Magazin

Information und Bildungsarbeit von und für die SAP-Community

Überfordert?

Wir bieten Information und Bildungsarbeit
von und für die SAP-Community


```

dbms_resource_manager.clear_pending_area;
dbms_resource_manager.create_pending_area;
dbms_resource_manager.create_plan (
plan=>'TAGWERK',comment=>'Tagesplan');
dbms_resource_manager.create_plan_directive(
plan=>'TAGWERK',group_or_subplan=>'SYS_GROUP', cpu_p1=>100,
comment=>'SYS User erhalten hoechste Prio');
dbms_resource_manager.create_plan_directive(plan=>'TAGWERK',
group_or_subplan=>'OLTP', cpu_p2=>100,-
comment=>'OLTP User');
dbms_resource_manager.validate_pending_area;
dbms_resource_manager.submit_pending_area;
    
```

Listing 1: Einfaches Beispiel

Abhängig von der Beantwortung der vorgenannten Fragen können nun Konzepte dafür erstellt werden, auf welcher Basis konsolidiert wird. Es ist zu klären, wie Systeme gruppiert werden. Soll physisch oder logisch zusammengefasst werden? Abhängig vom Sicherheitslevel der jeweiligen Datenbank muss überlegt werden, ob sie mit anderen Datenbanken zusammengelegt werden kann. Letztlich muss jede Datenbank daraufhin untersucht werden, in welchem Kontext sie betrieben wird und welche Datenbanken in dieselbe Kategorie fallen.

Zu Beginn einer Konsolidierung sind die folgenden Punkte relevant, um entsprechende Isolationslevel einrichten zu können.

- Fehlertoleranz
- Sicherheit
- Operational
 - Backup / Restore
 - Flashback
 - Patching
- Ressourcennutzung
 - CPU
 - Memory
 - Storage

Neben der Analyse der zu konsolidierenden Datenbanken ist es entscheidend, auf welcher Plattform konsolidiert wird. Bei

großen Umgebungen lohnt sich ein genauerer Blick auf die Oracle Exadata Database Machine. Durch den redundanten Aufbau und den intelligenten Storage bietet die Exadata die Möglichkeit, mehrere Datenbanken zu konsolidieren. Die Datenbanken lassen sich auch so betreiben, dass sie sich nicht gegenseitig stören, denn neben dem Resource Management können hier auch weitere Aspekte angegangen werden, wie z.B. eine strikte Trennung auf ASM-Ebene.

Oracle Resource Manager

Die Leistung einer Datenbank sollte gerade nach einer Konsolidierung nicht durch Probleme beeinträchtigt werden, die in einer anderen Datenbank auftreten. Datenbanken und die darauf betriebenen Applikationen müssen die notwendigen Ressourcen erhalten. Der Oracle Database Resource Manager (DBRM) und der Exadata I/O Resource Manager (IORM) stellen die Werkzeuge zur Verfügung, die erforderlich sind, um Datenbank- und Anwendungsleistung sicherzustellen. Beginnend mit der Oracle 8i hat Oracle die Möglichkeit geschaffen, Ressourcen in der Datenbank mit dem Database Resource Manager zu verwalten. Der Resource Manager kann helfen, in einer Da-

```

execute dbms_resource_manager.create_consumer_group(
consumer_group => 'TAGWERK_GROUP',
comment => 'Sessions for TAGWERK_GROUP');
    
```

Listing 2: Anlegen einer Consumer Group

```

EXECUTE DEMS_RESOURCE_MANAGER.SWITCH_CONSUMER_GROUP_FOR_SESS ('19', '1909',
'HIGH_PRIORITY');
    
```

Listing 3: Switch Group

tenbank Ressourcen-Engpässe zu lösen, die beispielsweise mit hoher CPU-Last in Zusammenhang stehen. Der DBRM läuft innerhalb der Datenbank, das Betriebssystem ist dabei nicht involviert, denn in der Regel gehören die Prozesse demselben Betriebssystem-User. Die Last wird gleichberechtigt für alle DB-Prozesse auf die zur Verfügung stehende CPU verteilt. Der Einsatz des DBRM greift allerdings nur bei einem Mangel an Ressourcen.

Für den Einsatz des DBRM ist es nötig, die Auslastung zu verstehen. Folgende Faktoren sind zu beachten:

- Analyse der Auslastung
- Dokumentation der Entscheidungen
- Aufbau eines Resource Management
 - Erstellen von Ressourcen-Plänen
 - Erstellen von Benutzergruppen
 - Festlegen von Prioritäten
 - Festlegen von Schwellwerten
 - Festlegen von Grenzen
- Monitoring

Ein einfaches Beispiel veranschaulicht *Listing 1*.

1. Gruppen

Zunächst werden Gruppen, Resource Consumer Group genannt, angelegt, die die jeweiligen Nutzungsprofile widerspiegeln (*siehe Listing 2*). Resource Consumer Groups bündeln Usersessions. Oracle liefert schon vordefinierte Resource Consumer Groups, wie etwa die SYS_GROUP, eine Gruppe, die aus den Usern SYS und SYSTEM besteht, und die OTHER_GROUPS, die aus allen anderen Usern besteht, die nicht zu einer Gruppe gehören.

Ändern sich Anforderungen, kann eine Session auch in eine andere Gruppe gewechselt werden (*siehe Listing 3*). Das geht automatisch, wenn sich die Voraussetzungen geändert haben. Aber auch ein manuelles Wechseln ist möglich.

2. Mapping Rules

Consumer Group Mapping Rules ermöglichen die Zuordnung von Usern in die für sie vorgesehenen Gruppen.

In Mapping Rules werden Regeln beziehungsweise Eigenschaften definiert, nach denen die einzelnen Sessions den Gruppen initial zugeordnet werden (*siehe Listing 4*). Ändern sich die Eigenschaften einer Session oder tritt eine Überschreitung einzelner Ressourcengrenzen auf,

kann dies zu einem automatischen Wechsel in eine andere Gruppe führen. Ein Wechsel zwischen den Gruppen ist natürlich auch manuell möglich.

Folgende Mapping Rules können zum Beispiel verwendet werden:

- DB User
- OS User
- TNS Alias
- MODULE und ACTION
- Client Program
- Client Machine

3. Plan

Der Ressourcen-Plan letztlich beinhaltet Regeln beziehungsweise Direktiven, mit denen die Ressourcen zwischen den definierten Gruppen aufgeteilt werden. Oracle hat auch hier schon einen Standardplan. Der `DEFAULT_MAINTENANCE_PLAN` wird vor allem bei den Oracle-internen Wartungsaufgaben, wie etwa Optimizer-Statistiken-Sammeln, eingesetzt.

Ressourcen-Pläne werden einfach aktiviert beziehungsweise deaktiviert (siehe Listing 5). Hier gilt es zu beachten, dass jeweils nur ein Plan aktiv sein darf. Ressourcen-Pläne können beliebig komplex sein. Einfache Pläne haben eine Ebene definiert, es ist aber genauso möglich, mehrere Ebenen mit entsprechenden Subplänen zu definieren.

Natürlich ist es möglich, Pläne jederzeit zu wechseln.

Instance Caging

Ein weiteres wichtiges Feature, das nur im Zusammenhang mit Resource Manager aktiviert werden kann, ist das sogenannte Instance Caging. Damit lässt sich seit Oracle 11g Release 2 auf allen Plattformen die Anzahl der CPUs pro Instance angeben, die diese maximal nutzen darf. Das ändert allerdings nichts an der nötigen Lizenzierung der Datenbank. Um das Feature zu aktivieren, wird einfach der Parameter `cpu_count` in der Datenbank-Instanz gesetzt. Es gibt verschiedene Herangehensweisen. So lässt sich `cpu_count` konservativ als Summe aller zur Verfügung stehenden CPUs setzen. Aber auch eine Überprovisionierung ist möglich. In allen Fällen greift der Parameter `cpu_count` nur bei einem aktivierten Ressourcen-Plan.

Für das Monitoring ist sicher dieses Wait Event interessant: „`resmgr:cpu`

```
execute dbms_resource_manager.set_consumer_group_mapping(
attribute => dbms_resource_manager.client_os_user,
value => 'STEFSECK',
consumer_group => 'TAGWERK_GROUP');
```

Listing 4: Anlegen einer Mapping Rule

```
ALTER SYSTEM SET RESOURCE_MANAGER_PLAN = 'Tagesplan';
```

Listing 5: Aktivieren eines Ressourcen-Plans

```
ALTER IORMPLAN dbplan=((name=sales, share=8, flashCacheSize=10G), -
(name=finance, share=8, flashCacheLimit=10G, flashCacheMin=2G), -
(name=dev, share=2, flashCacheLimit=4G, flashCacheMin=1G), -
(name=test, share=1, flashCacheSize=1G))
```

Listing 6: Aktivieren eines IORM-Plans

quantum“. Denn das deutet darauf hin, dass entweder der Parameter zu niedrig gesetzt ist oder in der Instanz zu viel Last auftritt.

Multitenant

Mit der Einführung von Oracle Multitenant ist es nun auch möglich, Ressourcen-Pläne für die CDB und einzelne PDBs zu erstellen. Der CDB-Plan kontrolliert, wie zum Beispiel CPUs von PDBs genutzt werden, und der PDB-Plan prüft, wie dann etwa CPUs von Consumer Groups genutzt werden.

IO Resource Management

Ein weiteres Feature, das bei der Konsolidierung auf einer Exadata wichtig wird, ist der IO Resource Manager (IORM). Mit dieser zusätzlichen Funktion ist es möglich, nicht nur die Ressourcen innerhalb einer Datenbank zu kontrollieren, sondern auch die Nutzung zwischen den Datenbanken zu priorisieren. Es ist möglich, IO-Anfragen auf Storage-Ebene, also in den Exadata Cell Nodes, zu verwalten (siehe Listing 6).

Der IORM muss auf jedem Cell Node aktiviert werden.

Fazit

Mit Standardisierung und Konsolidierung ist es möglich, den vorhandenen Zoo an

Datenbank-Versionen und Konfigurationen in den Griff zu bekommen. Dazu ist sicher eine genaue Planung und ein entsprechendes Konzept hilfreich, das aber auch sehr gewissenhaft getestet sein muss. Konsolidierte Umgebungen lassen sich mit dem Resource Manager gut betreiben, weil die vorhandenen Ressourcen auf alle intelligent verteilt werden können. Die vorgestellten Features und Funktionen bieten Unterstützung bei der Konsolidierung.

Dokumentation

- MOS 1207483.1 „CPU Resource Manager“
- MOS 1484302.1 „Overview of Oracle Resource Manager“
- MOS 1339769.1 „Master Note for Oracle Database Resource Manager“
- <https://docs.oracle.com/en/database/oracle/oracle-database/19/admin/managing-resources-with-oracle-database-resource-manager.html>



Stefan Seck
stefan.seck@logicalis.de



Ein APEX Workflow-Tool für Citizen Developer

Michael Weinberger, Verbund

Oracle APEX ist eine Plattform, mit der auch Citizen Developer ohne tiefere Programmierkenntnisse in sehr kurzer Zeit sichere und stabile Webapplikationen ohne Codierungsaufwand erstellen können. Während sich sehr gute und sichere Benutzeroberflächen mit Leichtigkeit anlegen lassen, ist es weniger leicht, eine Geschäftslogik einzubinden. Für die Umsetzung solcher Anforderungen bieten sich Workflowtools an. Deren typische Installation (Tomcat/Java auf Server), Konfiguration, Ansteuerung über Webservice und grafische Darstellung (JavaScript) stellen jedoch ihrerseits für Citizen Developer zumindest eine Herausforderung dar. Besonders in einer geschlossenen Umgebung wie beispielsweise apex.oracle.com

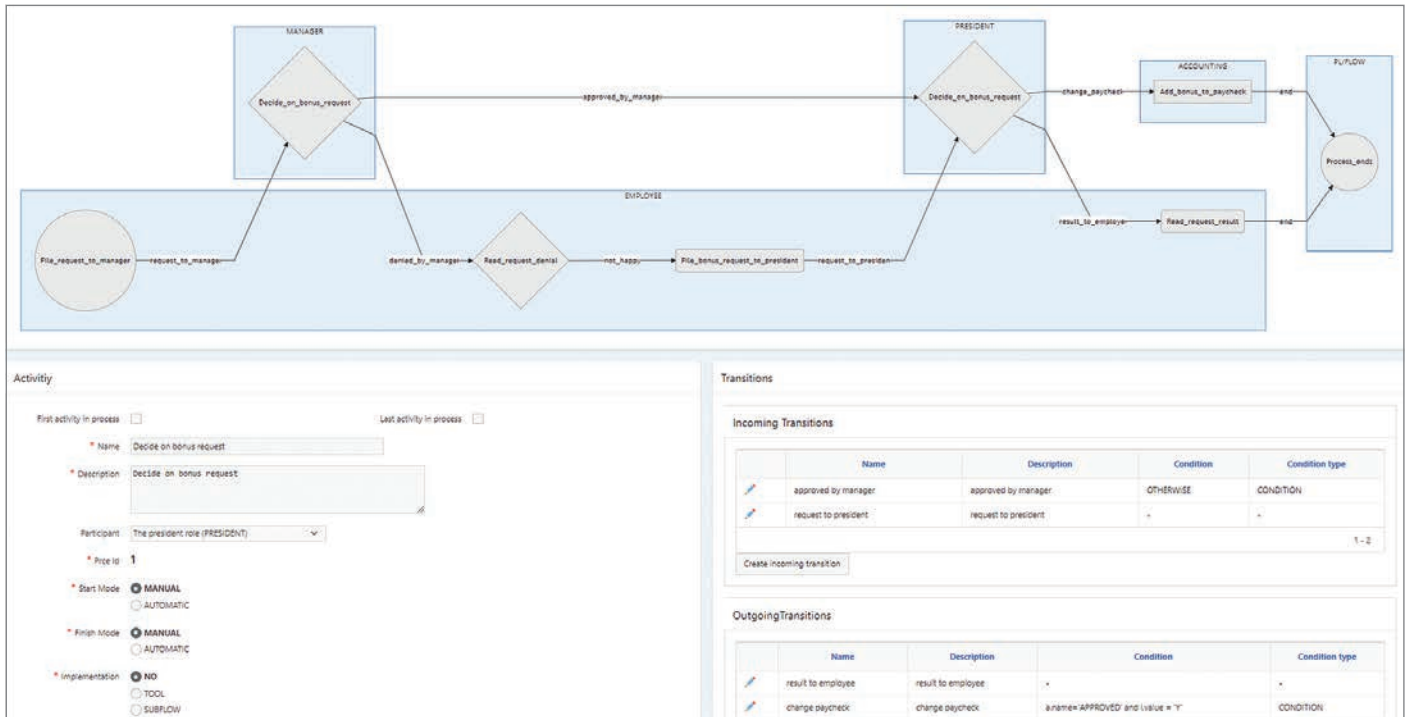


Abbildung 1: Prozess in einer APEX-Applikation mit dem Mermaid Plug-in (Quelle: Michael Weinberger)

In dieser Mini-Serie möchte ich einen einfacheren Weg zeigen. In Teil 1 habe ich die bereits 2005 von Yeb Havinga veröffentlichte Workflowengine vorgestellt [2] – von der Installation über notwendige Adaptierungen bis zur detaillierten Ansteuerung – und wie diese mit APEX umgesetzt werden kann.

Teil 2 konzentriert sich auf die grafische Prozesssteuerung mittels APEX-Plug-in und erklärt detailliert, wie Benutzerinteraktionen mit APEX ermöglicht werden.

und ist damit sehr leicht zu benutzen (siehe Listing 1).

Um die Erstellung eines Markdown-Textes für Mermaid basierend auf den

Workflow-Definitionen aus PLFLOW zu vereinfachen, habe ich für Mermaid ein APEX-Plug-in erstellt – es ist auf apex.world [1] zu finden.

```
graph TD
A[Christmas] -->|Get money| B(Go shopping)
B --> C{Let me think}
C -->|One| D[Laptop]
C -->|Two| E[iPhone]
C -->|Three| F[fa:fa-car Car]
```

Listing 1: Beispiel Markdown für Mermaid

Visualisierung des Workflows mit einem APEX-Plug-in

Master-Detail-Formulare stellen eine brauchbare Variante zur Dateneingabe dar. Schöner und übersichtlicher ist allerdings eine Visualisierung über ein Flowchart:

Diese Visualisierung wurde mit Mermaid erstellt.[3]

Mermaid ist eine offene Bibliothek, mit der sich Flowcharts (unter anderem) leicht visualisieren lassen. Der Vorteil von Mermaid ist die automatische grafische Anordnung von Prozessschritten entsprechend einer Gruppe (Swimlanes, im Beispiel die blauen Blöcke). Mermaid benutzt eine Markdown-ähnliche Sprache

```
CREATE OR REPLACE FORCE VIEW PLFLOW.WF_ACTIVITIES_V
AS
SELECT DISTINCT
  a.ID,
  prce_id,
  a.NAME,
  a.DESCRPTION,
  a.SPLIT,
  a.START_MODE,
  a.pati_query,
  a.pati_exclude_query,
  NVL2 (T_in.ACTI_ID_TO, NULL, 1) AS START_ACTIVITY_PRCE_ID,
  NVL2 (T_out.ACTI_ID_FROM, NULL, 1) AS END_ACTIVITY_PRCE_ID
FROM PLFLOW.WF_ACTIVITIES A,
  PLFLOW.WF_TRANSITIONS T_out,
  PLFLOW.WF_TRANSITIONS T_in
WHERE T_out.ACTI_PRCE_ID_FROM(+) = A.PRCE_ID
  AND T_out.ACTI_ID_FROM(+) = a.id
  AND T_in.ACTI_PRCE_ID_TO(+) = A.PRCE_ID
  AND T_in.ACTI_ID_TO(+) = a.id;
```

Listing 2: View für Abfrage der Activities

Um dieses Plug-in verwenden zu können, muss PLFLOW um folgende View erweitert werden (siehe Listing 2):

Das Mermaid Plug-in

... arbeitet wahlweise mit einem einfachen Markdown-Text wie in Listing 8 oder alternativ mit drei SQL-Statements. Letztere Option ist für die Einbindung von PLFLOW entwickelt worden. Diese drei SQL-Statements laden der Reihe nach alle Activities (Nodes), alle Transitions (Connector Links) und CSS für besondere Hervorhebungen aus der Datenbank (Schema PLFLOW), setzen daraus einen Markdown-Code zusammen und starten Mermaid.

Für die Visualisierung von PLFLOW sehen diese drei SQL-Statements folgendermaßen aus (siehe Listing 3).

SQL for Connector Links (siehe Listing 4).

SQL for CSS (siehe Listing 5).

Aus diesen drei Statements wird der Markdown-Code erzeugt, der von Mermaid als Flowchart dargestellt wird. Dabei sind Referenzen zu Apex-Items (d. h. &P201_PRCE_ID.) entsprechend anzupassen.

Es wäre zweckmäßig, eine eigene Tabelle mit den Spalten Style und Definition für das CSS anzulegen. Das „Style“-Element wird bei allen Nodes (Activities) und Connector Links (Transitions) ausgegeben. Besondere Bedeutung findet das Style-Element bei laufenden Prozessen, bei denen jede PLFLOW Activity einen von mehreren Status hat:

```
NOTSTARTED
NOTRUNNING
RUNNING
COMPLETED
SUSPENDED
TERMINATED
ABORTED
```

Über eine Tabelle kann damit jedem Status eine eigene Erscheinung gegeben und so zum Beispiel der aktuelle Prozessschritt damit hervorgehoben werden.

Über die Links in dem SQL for Nodes lässt sich leicht eine direkte Ansteuerung der Bearbeitungsoberfläche umsetzen: Bei einem Klick auf einen Node öffnen sich dann die Bearbeitungsmasken für diese Aktivität und die zugehörigen Transaktionen.

SQL for Nodes:

```
select a.id,
       replace(a.name, ' ', '_') as text,
       case when a.split is not null THEN 'rhombus'
            when a.START_ACTIVITY_PRCE_ID is not null then 'circle'
            when a.END_ACTIVITY_PRCE_ID is not null then 'circle'
            when a.start_mode = 'MANUAL' THEN 'round'
            when a.start_mode = 'AUTOMATIC' THEN 'box'
       end as Apperance,
       'f?p=&APP_ID.:&APP_PAGE_ID.:&APP_SESSION.::::P201_ID:'||a.id as link,
       null as style,
       replace(p.name, ' ', '_') as subgraph,
       replace(a.description, ' ', '_') as tooltip
from wf_activities_v a, wf_participants p
where a.prce_id = &P201_PRCE_ID.
      and a.PATI_QUERY = p.id
```

Listing 3: SQL für die einzelnen Schritte des Prozesses

```
SELECT ACTI_ID_FROM as from_node,
       ACTI_ID_TO as to_node,
       'arrow' as apperance,
       replace(NAME, ' ', '_') as text,
       null as style,
       rownum as DISPL_SEQ
FROM PLFLOW.WF_TRANSITIONS
where (ACTI_PRCE_ID_FROM = &P201_PRCE_ID. OR ACTI_PRCE_ID_TO = &P201_PRCE_ID.)
order by ACTI_ID_FROM,ACTI_ID_TO
```

Listing 4: SQL für die Übergänge zwischen den Prozessschritten

```
SELECT 'RUNNING' as style, 'fill:#f9f,stroke:#333,stroke-
width:4px;' as definition from dual
```

Listing 5: Beispiel SQL für CSS-Erstellung

```
PL_FLOW.ChangeActivityInstanceState(
  acin_id_in => :P2_ACIN_ID,
  state_in   => 'RUNNING',
  pati_id_in => :P1_PARTICIPANT
);

PL_FLOW.AssignProcessInstanceAttribute(
  prin_id_in => :P2_PRIN_ID,
  name_in    => :P2_ATTRIBUTE,
  value_in   => :P2_VALUE
);

PL_FLOW.ChangeActivityInstanceState(
  acin_id_in => :P2_ACIN_ID,
  state_in   => 'COMPLETED',
  pati_id_in => :P1_PARTICIPANT
);
```

Listing 6: PL/SQL-Statements zur Prozessschrittbearbeitung

Mit dieser Eingabemaske und mit dem Flowchart ist die administrative Oberfläche zur Erstellung von (Geschäfts-) Prozessen soweit einmal vollständig. Wer will,

kann das Erstellen von neuen Aktivitäten durch Schaltflächen wie „manuelle Bearbeitung“, „neue automatische Aktivität“ oder „Entscheidung benötigt“ beschleunigen.

gen. Hier gibt es noch ein weites Feld an Möglichkeiten, besonders, wenn der Entwickler mit JavaScript arbeiten will.

Aufgabenliste in APEX

Bis jetzt haben wir eine Administrationsoberfläche, in der Prozesse entwickelt werden können und ein vergleichsweise trivialer Start aus Applikationen heraus möglich ist.

Das wirkliche Kernstück, mit dem die Akzeptanz dieses Workflowsystems steht und fällt, ist jedoch die Aufgabenliste. Diese ist eine eigene APEX-Applikation, die ebenfalls einmalig auf dem PLFLOW Workspace entwickelt wird und jedem möglichen Participant (Mitarbeiter) zur Verfügung gestellt wird.

Dabei handelt es sich um eine einfache Oberfläche, in der alle aus Workflows resultierenden Aktivitäten für jeden Bearbeiter sicht- und bearbeitbar sind. In

APEX sieht das am besten als „Report and Form“-Kombination aus. Dabei werden die angezeigten Aktivitäten automatisch nach dem angemeldeten Benutzer gefiltert.

Auf der Formularseite zur Bearbeitung einer Aktivität sind wieder mehrere Schritte notwendig. Es handelt sich um ein Arbeitsschema, das sich durch die ganze PLFLOW Engine zieht:

1. Aktivität starten
2. Attribute ändern
3. Aktivität abschließen

oder in PLSQL (*siehe Listing 6*):

Diese drei Statements müssen wieder als PL/SQL-Prozess beim Submit ausgeführt werden. Ein direkter Zugriff auf Tabellen findet in diesem Formular nicht statt. Für diese Oberfläche kommt uns jetzt die Ergänzung der WF_ATTRIBUTES-Tabelle zugute: In der Spalte LOV stehen ja

alle für dieses Attribut gültigen Antworten (z.B: genehmigt:abgelehnt) und diese können dem Bearbeiter leicht als Auswahlliste zur Verfügung gestellt werden. Gibt es für diesen Workflow auch ein PAYLOAD-Attribut, ist auf der Formularseite jetzt der beste Platz, um das BLOB anzuzeigen und das CLOB mit dem Begleittext zu editieren.

Bleibt noch der Report für die Auflistung aller den aktuellen Anwender betreffenden, offenen Tasks.

PLFLOW bietet dafür in dem Package eine eigene Prozedur: `PL_FLOW.OpenWorkList`

Leider ist die Anwendung von `OpenWorkList` etwas komplex. Da sie für eine rein programmatische Ansteuerung entwickelt wurde, besteht der Aufruf aus Parametern für Sortierungen und einer komplizierten Filtersyntax. Retourneriert wird ein Cursor.

Glücklicherweise sind das alles Punkte, die APEX viel besser, schneller und schöner erledigen kann. Wir müssen nur das

```
where    p.id          = i.prin_id          -- link prin_id to activity_instance
        AND i.negation_ind = 'N'          -- only valid instances, not faked ones
        --AND (i.pati_id IS NULL OR i.pati_id IN (SELECT * FROM TABLE(CAST( :role_id_table AS int_table_type
)))) --only activities that participant may be able to perform
        AND (i.pati_id IS NULL OR i.pati_id=:P1_PARTICIPANT OR i.pati_id IN (SELECT * FROM TABLE(pl_flow.has_
roles(:P1_PARTICIPANT,'')))) --only activities that participant may be able to perform
        -- link possible current performer
        AND i.id          = f.acin_id (+)
        AND f.pati_id     = b.id (+) -- link performer to participant record
        AND f.state (+)   = 'CURRENT'
        AND (f.accepted IS NULL OR f.accepted <> 'N')
        --link possible assigned performer (role, ou or human)
        AND i.id          = f_assigned_role.acin_id (+)
        AND i.id          = f_assigned_human.acin_id (+)
        AND i.id          = f_assigned_ou.acin_id (+)
        --if someone is assigned, it has to be equal to the participant, or the participant has to have a role
that equals the assigned.
        --AND (f_assigned_role.pati_id IS NULL OR f_assigned_role.pati_id IN (SELECT * FROM TABLE(CAST( :role_
id_table AS int_table_type )))
        AND (f_assigned_role.pati_id IS NULL OR f_assigned_role.pati_id=:P1_PARTICIPANT OR f_assigned_role.pati_
id IN (SELECT * FROM TABLE(pl_flow.has_roles(:P1_PARTICIPANT,''))))
        --AND (f_assigned_ou.pati_id IS NULL OR f_assigned_ou.pati_id IN (SELECT * FROM TABLE(CAST( :role_id_ta
ble AS int_table_type )))
        AND (f_assigned_ou.pati_id IS NULL OR f_assigned_ou.pati_id=:P1_PARTICIPANT OR f_assigned_ou.pati_id IN
(SELECT * FROM TABLE(pl_flow.has_roles(:P1_PARTICIPANT,''))))
        --AND (f_assigned_human.pati_id IS NULL OR f_assigned_human.pati_id IN (SELECT * FROM TABLE(CAST( :role_
id_table AS int_table_type )))
        AND (f_assigned_human.pati_id IS NULL OR f_assigned_human.pati_id=:P1_PARTICIPANT OR f_assigned_human.
pati_id IN (SELECT * FROM TABLE(pl_flow.has_roles(:P1_PARTICIPANT,''))))
        -- link responsible participant to processinstance
        AND pati_resp.id (+) = p.pati_id
        -- join activity instance with activity
        AND acti.id        = i.acti_id
        AND acti.prce_id   = i.acti_prce_id
        AND acti.implementation = 'NO' --only select manual activities
        -- dont select future activities
        AND TRUNC(i.date_created) <= SYSDATE
```

Listing 7: Der Select-Teil aus `PL_FLOW.OpenWorkList`

```
where ACTI_PRCE_ID = wf_processes.id
and acti_id = wf_activities.id
```

Listing 8: Notwendige Anpassungen

```
CREATE OR REPLACE FORCE VIEW PLFLOW.WF_ACTIVITY_INSTANCES_V
AS
SELECT DISTINCT
  ai.ID,
  ai.prin_id,
  a.NAME,
  a.DESCRPTION,
  a.SPLIT,
  a.START_MODE,
  ai.pati_id,
  ai.pati_id_exclude,
  ai.state,
  NVL2 (T_in.ACIN_ID_TO, NULL, 1) AS START_ACTIVITY_PRCE_ID,
  NVL2 (T_out.ACIN_ID_FROM, NULL, 1) AS END_ACTIVITY_PRCE_ID
FROM PLFLOW.WF_ACTIVITY_INSTANCES AI,
  PLFLOW.WF_TRANSITION_INSTANCES T_out,
  PLFLOW.WF_TRANSITION_INSTANCES T_in,
  PLFLOW.WF_ACTIVITIES A
WHERE T_out.TRSI_ACTI_PRCE_ID_FROM(+) = Ai.PRIN_ID
AND T_out.TRSI_ACTI_ID_FROM(+) = ai.id
AND T_in.TRSI_ACTI_PRCE_ID_TO(+) = Ai.PRIN_ID
AND T_in.TRSI_ACTI_ID_TO(+) = ai.id
and AI.ACTI_ID = A.ID
```

Dann muss ein SQL-Statement für das Plug-in angepasst werden:

SQL for Nodes:

```
select a.id,
  replace(a.name, ' ', '_') as text,
  case when a.split is not null THEN 'rhombus'
    when a.START_ACTIVITY_PRCE_ID is not null then 'circle'
    when a.END_ACTIVITY_PRCE_ID is not null then 'circle'
    when a.start_mode = 'MANUAL' THEN 'round'
    when a.start_mode = 'AUTOMATIC' THEN 'box'
  end as Apperance,
  'f?p=&APP_ID.:&APP_PAGE_ID.:&APP_SESSION.::::P201_ID:'||a.id as link,
  a.state as style,
  replace(p.name, ' ', '_') as subgraph,
  replace(a.description, ' ', '_') as tooltip
from WF_ACTIVITY_INSTANCES_V a, wf_participants p
where a.prin_id = &P201_PRCE_ID.
and a.PATI_ID = p.id
```

Listing 9: SQL für Aktivitäten bei Workflows (laufenden Prozessen)

dahinter liegende SQL als Basis nehmen und etwas modifizieren.

In dem Sourcecode von PL_FLOW ist das Select-Statement in drei Teile zerlegt:

```
l_select_expr
l_from_expr
l_where_expr
```

Die ersten beiden Teile können so, wie sie sind, übernommen werden. Erst im Whe-

re-Teil (l_where_expr) gibt es Anpassungen (siehe Listing 7):

Diese Anpassungen bestehen darin, dass alle Zeilen mit einem CAST

```
--AND (i.pati_id IS NULL OR
i.pati_id IN (SELECT * FROM
TABLE(CAST( :role_id_table AS
int_table_type )))) --only acti-
vities that participant may be
able to perform
```

so korrigiert werden, dass auf den aktuellen Benutzer verwiesen wird:

```
AND (i.pati_id IS NULL OR
i.pati_id=:P1_PARTICIPANT OR
i.pati_id IN (SELECT * FROM
TABLE(pl_flow.has_roles(:P1_PARTICIPANT,'')))) --only activi-
ties that participant may be able
to perform
```

Dabei steht das Item :P1_PARTICIPANT für die ID jener Person, die gerade die Aufgabenliste bearbeitet. Also die (z.B.) Personalnummer des :APP_USER.

Leider fehlt in diesem SQL eine lesbare Beschreibung des Prozesses oder der Aktivität. Aber diese Informationen lassen sich über Joins anhängen (siehe Listing 8):

Der so entstandene Report kann vom Endbenutzer über APEX-Bordmittel beliebig formatiert werden.

Um die Applikation benutzerfreundlicher zu machen, sollte über einen Hintergrundjob (Job-Scheduler) periodisch eine E-Mail mit einem Link auf diese Applikation an jeden Benutzer geschickt werden – sofern er gerade offene Aufgaben hat.

Verwaltungsoberfläche: Ansicht der laufenden Prozesse

Zurück zur Administrationsoberfläche – dort fehlt uns noch ein Monitoring aller aktuell laufenden Geschäftsprozesse. Mit dem bisher Umgesetzten lässt sich das leicht ergänzen.

Die Prozessvisualisierung mittels Mermaid muss dazu ein wenig angepasst werden:

Zuerst ist eine neue View notwendig, die auf die Instanztabellen zugreift (siehe Listing 9).

Da sich nur die Activities ändern (vor allem im Status), muss nur das Statement für die Nodes angepasst werden. Hier kommt jetzt die volle Möglichkeit des CSS-Statements zum Tragen – je nach Status der Activity wird die Erscheinung automatisch angepasst.

Bei zusätzlichen Abfragen muss beachtet werden, dass alle Informationen zu laufenden Prozessen in den Tabellen WF_PROCESS_INSTANCES und WF_ACTIVITY_INSTANCES vorhanden sind. Die Stan-

WF_FORMAL_PARAMETERS

DATA_TYPE	VARCHAR, NUMBER, DATE
FOPA_INDEX	Position (1.,2....) in der PL/SQL-Prozedur
FOPA_MODE	IN, OUT oder INOUT
ATRI_ID	ID des Attributes, das referenziert wird

Tabelle 1: WF_FORMAL_PARAMETERS

WF_ACTUAL_PARAMETERS

DATA_TYPE	VARCHAR, NUMBER, DATE
FOPA_ID	Fremdschlüssel für WF_FORMAL_PARAMETERS
EXPRESSION	Eine SQL Expression wie sysdate, oder 1+1
ATRI_ID	ID des Attributes, das referenziert wird

Tabelle 2: WF_ACTUAL_PARAMETERS

```
myschema.send_mail(empfaenger in varchar2);
Grant execute on myschema.send_mail to PLFLOW;
```

Listing 10: Beispielprozedur, die angesteuert werden soll

```
INSERT INTO PLFLOW.WF_APPLICATIONS (
    NAME, PLSQL_PROC_NAME, PRCE_ID)
VALUES ( 'Sende Benachrichtigung' /* NAME */,
    'myschema.send_mail' /* PLSQL_PROC_NAME */,
    4711 /* PRCE_ID */ ) RETURN ID into V_APLI_ID
;
```

Listing 11: ...deren Definition in PLFLOW-Tabelle

```
INSERT INTO PLFLOW.WF_ACTIVITIES (
    .... )
VALUES ( 4711 /* PRCE_ID */,
    V_APLI_ID /* APLI_ID */,
    'Versende Benachrichtigung' /* NAME */,
    'AUTOMATIC' /* START_MODE */,
    'AUTOMATIC' /* FINISH_MODE */,
    'TOOL' /* IMPLEMENTATION */,
    1 /* PATI_QUERY */
);

INSERT INTO PLFLOW.WF_ATTRIBUTES (
    ....)
VALUES ( 4711 /* PRCE_ID */,
    'CHARACTER' /* DATA_TYPE */,
    'E-Mail Adresse' /* NAME */,
    '100' /* LENGTH */,
    'N' /* KEEP */,
    ) RETURN ID into V_ATRI_ID;
```

Listing 12: Der Prozessschritt

```
INSERT INTO PLFLOW.WF_FORMAL_PARAMETERS (
    ....)
VALUES (V_ATRI_ID /* ATRI_ID */,
    4711 /* PRCE_ID */,
    V_APLI_ID /* APLI_ID */,
    'VARCHAR2' /* DATA_TYPE */,
    '1' /* FOPA_INDEX */,
    'IN' /* FOPA_MODE */,
    'Adresse des E-Mail Empfängers' /* NAME */
) RETURN ID into V_FOPA_ID;

INSERT INTO PLFLOW.WF_ACTUAL_PARAMETERS (
    ....)
VALUES (V_ATRI_ID /* ATRI_ID */,
    V_ACTI_ID /* ACTI_ID */,
    4711 /* ACTI_PRCE_ID */,
    V_FOPA_ID /* FOPA_ID */,
    Null /* EXPRESSION */
);
```

Listing 13: Die Parameterdefinition als Verbindung zwischen Prozessschritt und Applikation

ardt Tabellen WF_ACTIVITY_INSTANCES enthalten nur die leeren Prozessmatrizen.

Bei der Ansicht aller laufenden Prozesse werden keine Formulare benötigt.

Ansteuern einer anderen Applikation aus dem Workflow heraus

PLFLOW kann beliebige andere Applikationen ansteuern. Im Kern erfolgt das über eine PL/SQL-Prozedur. Diese kann in einem anderen Schema etwas machen. Oder ein Webservice starten. Oder über den Job-Scheduler Kommandos auf Betriebssystem-Ebene ausführen. Oder Files schreiben beziehungsweise per E-Mail verschicken.

Zuerst wird ein Programm definiert. Das geschieht durch einen Eintrag in die Tabelle WF_APPLICATIONS. Wichtig sind die PRCE_ID und der Name des aufzurufenden Programmes (PLSQL_PROC_NAME). Diese Prozedur sollte in einem anderen Schema – nicht in PL_FLOW – stehen. Und natürlich muss PLFLOW diese Prozedur ausführen können.

Dann wird eine Activity mit dem Typ Implementation TOOL benötigt. Startmode und Finishmode sollten auf AUTOMATIC gestellt werden und in die Spalte APLI_ID kommt die ID aus der Tabelle WF_APPLICATIONS. Sobald diese Activity ausgeführt wird, startet sie die PL/SQL-Prozedur und wartet auf das Ergebnis.

Bleiben noch die Übergabeparameter. Jeder Übergabeparameter in der PL/SQL-Prozedur muss einzeln definiert werden.

Dafür gibt es zwei Tabellen WF_FORMAL_PARAMETERS und WF_ACTUAL_PARAMETERS (siehe Tabelle 1 und 2).

Letztere hat einen Fremdschlüssel (FOPA_ID) auf WF_FORMAL_PARAMETERS.ID.

Existiert eine Attributs-ID, wird der Wert vom Attribut genommen. Nur wenn das Feld leer ist, wird die EXPRESSION ausgewertet.

Dazu ein Beispiel:

Wir haben eine Prozedur, die eine vordefinierte Nachricht verschickt (*siehe Listing 10*):

Für diese erstellen wir einen formalen Eintrag (*siehe Listing 11*):

Dann haben wir eine Aktivität, die die Applikation verwendet und in einem Attribut die Empfängeradresse hat (*siehe Listing 12*):

Für das Ansteuern der Prozedur müssen dann nur noch die Parameter definiert werden (*siehe Listing 13*):

Fazit

Zugegeben, die Einrichtung von PLFLOW und die Entwicklung der Applikationen für die Administration beziehungsweise der Aufgabenlisten sind auch in APEX nicht ganz trivial – aber für jemanden, der mit APEX arbeitet, umsetzbar.

Die Vorteile, dass sich das Tool leicht ansteuern und in wenige Applikationen packen lässt und dabei nichts außerhalb von APEX braucht, überwiegen meiner Meinung nach.

Und wie sieht es mit der Performance aus? Die Engine ist prinzipiell für den Parallelbetrieb ausgerichtet. Da sie jedoch in der Datenbank läuft, können CPUs nicht so einfach hinzugefügt werden. Die Eignung für den Betrieb einer Enterprise-Lösung mit 10.000+ Workflows sollte also vorher evaluiert werden. Aber als Prozesswerkzeug für einige Dutzend Applikationen in kleineren Organisationseinheiten mit bis zu 150 Mitarbeitern sollte sie durchaus geeignet sein – und das ist die Größe, bis zu der Citizen Developer üblicherweise entwickeln.

MLDB und hat 2009 begonnen, eine konzernweite Low-Code-Applikationsplattform (self-service, automatische Authentifizierung, SQL-Schnittstellen zu Benutzerverzeichnissen und ein zentrales Plug-in / Code Repository) für Fachbereichsentwickler mittels APEX aufzubauen. Seitdem bildet er neue Entwickler in den Fachbereichen aus und unterstützt diese beim Aufbau ihrer eigenen Applikationslandschaften.

Sein aktuelles Projekt ist die erste kommerzielle SaaS-Plattform des Konzerns.

Quellen

- [1] <https://APEX.world>
- [2] Yeb Havinga (2005) <https://sourceforge.net/projects/plflow/>
- [3] Knut Sveidqvist (2014), Mermaid, <https://mermaidjs.github.io>

Über den Autor

Michael Weinberger ist Teamleiter bei VERBUND. Er arbeitet seit 2005 mit APEX/HT-



Michael Weinberger
Michael.Weinberger@verbund.com

Agil unterwegs in Infrastruktur-Projekten, eine sinnvolle Alternative?

Casimir Schmid, dbi Services

In der Software-Entwicklung genießt die agile Projektentwicklung heutzutage große Aufmerksamkeit, verspricht sie doch, dass durch sie auch komplizierte Aufgabenstellungen gemeistert werden können. Ist dies beim Aufbau und Betrieb von IT-Infrastrukturen ebenfalls sinnvoll? Erfahrene Leser überspringen Hinweise auf theoretische Grundlagen und gelangen über Erfahrungswerte aus der Praxis zur Schlüsselfrage: Wann agil, wann klassisch vorgehen?

Grundlagen

Als Einführung empfiehlt sich das Studium des offiziellen Scrumguides, der in viele Sprachen übersetzt als verständliche und umfassende Referenz von Scrum gilt [1].

Die Videos der Firma it-agile [2] sind kompakt gehalten und erklären auf anschauliche Art und Weise Scrum, Kanban sowie darüber hinaus weitere agile Themen.

Aus der Praxis

Agile Vorgehensweisen wie Scrum sind schnell erklärt, jedoch schwierig zu meistern, da sie die Kultur verändern, wie Dinge erledigt werden. Oftmals interpre-

tieren beteiligte Werte aus dem agilen Manifesto [3] und Vorgehensweisen sehr unterschiedlich und individuell. Die Einführung ist sehr aufwendig und benötigt viel Zeit. Vergleichbar ist dies mit Tanzen, Schach oder anderen sportlichen Disziplinen, wo Perfektion nur durch langes und hartes Training erreicht wird.

Agile Methoden führen zu schonungsloser Transparenz. Es ist anstrengend und entgegen der persönlichen Komfortzone, sich regelmäßig zu erklären und gleichzeitig offen für Veränderungen zu sein. Teammitglieder verändern und lernen unterschiedlich schnell und tief. Da dieser Lernprozess in Scrum elementar verankert ist und über die ganze Projektdauer anhalten wird, muss das Team durch einen agilen Spezialisten, einem Scrum Master, begleitet werden. Ziel ist es, sich nicht nur (während des Sprint Planning) zu fragen, durch welche Arbeitsschritte das Ergebnis noch wertschöpfender gestaltet werden kann. Ebenso soll man Effektivität und Effizienz von Methoden und Abläufen (in der Retrospektive) kritisch hinterfragen und nur solche ausüben, die positive Effekte hervorbringen. Konkret und nicht abschließend sind dies:

Bezogen auf die Beteiligten:

- Zufriedenheit der Endbenutzer
- Zufriedenheit der Team-Mitglieder (z.B. Anzahl Krankheitstage)
- gegenseitiger Respekt

Bezogen auf die zu erarbeitende Infrastruktur:

- Time-to-Market
- Effektivität der Lösung
- Skalierbarkeit und Umsetzungsdauer
- Wirtschaftlichkeit: Kosten für den Aufbau und Betrieb

Die Reflexion während Retrospektiven führt zu einem kontinuierlichen Verbesserungsprozess. Er basiert darauf, dass während des Projekts ständig dazugelernt wird. Es wird versucht, die Ursachen von Erfolg und Misserfolg zu ergründen und notwendige Anpassungen zu formulieren. Damit dies funktioniert, sind eine offene Atmosphäre und guter Teamspirit notwendig. So entstehen sachliche Diskussionen, die wiederum ermöglichen, positive Maßnahmen zu finden. Es gilt, einen individuell aufs Team bezogenen kleinen, optimalen Punkt zu treffen. Denn Retrospektiven sollen weder zu hart noch zu weich geführt werden. Es

ist wenig sinnvoll und für Beteiligte frustrierend, Ursachen zu ergründen und Maßnahmen vorzuschlagen, die entweder gar nicht oder nur halbherzig umgesetzt werden.

Bei allen Vorgängen in agilen Vorgehensweisen, bei denen Personen zusammenkommen, wird ersichtlich, dass dem Menschen eine zentrale Rolle zugeschrieben wird. Man wird sich bewusst, dass man nicht nur für Menschen, sondern auch von Menschen entwickelt. Man beginnt, dem Menschen gerecht zu handeln. Die Illusion ist weit verbreitet, dass es automatisch zu Projekterfolg und fantastischen Ergebnissen führt, auch wenn man agile Methoden nur dem Namen nach einführt, also nur oberflächlich und ohne die richtige Vorbereitung, gegen den Widerstand von Mitarbeitenden oder der Unternehmensführung. In der Realität zeigt sich jedoch, dass Erfolg nur durch harte Arbeit und kontinuierliche Bemühungen um Verbesserung der Zusammenarbeit, der Transparenz und der Denkweise im Unternehmen erreicht werden kann. Negativbeispiele sind die oberflächliche Einführung von:

- agilen Rollen: „Du bist ab sofort nebenbei Scrum Master“ ohne Schulung, Motivation und Prüfung der Eignung des Mitarbeitenden für diese anspruchsvolle Rolle.
- Meetings: Das Daily Scrum wird dazu benutzt, Schuldige auszumachen. Dabei ist die Schuldfrage irrelevant, es geht darum, sich auf Lösungen zu konzentrieren.
- Werkzeugen wie JIRA, die oft falsch verwendet werden und als Werkzeug allein nicht viel nützen, wenn Prozesse nicht richtig gelebt werden.
- Die Verwendung von Buzz-Words, ohne sich mit den agilen Prinzipien tiefergehend beschäftigt zu haben: „Wir betreiben MVP-Schneidung (Minimum Viable Product) für eine schnellere Time-to-Market“, obwohl die MVP-Pakete so riesig sind, dass durchschnittlich vier Monate Entwicklungszeit notwendig sind.

Um diese häufig vorkommende Illusion zu veranschaulichen, passt das Phänomen des „Cargo“-Kults, das besonders während des zweiten Weltkriegs in Neuguinea stark verbreitet war. Kriegsmaterial, das massenhaft von der US-Armee auf diese Inseln abgeworfen wurde (Fertigkleidung, Konservennahrung, Zelte, Waffen und an-

dere Ware), brachte drastische Änderungen des Lebensstils der Inselbewohner mit sich: Sowohl die Soldaten als auch die Einheimischen, die sie beherbergten, wurden mit Materialmengen regelrecht überschüttet. Oft wurden dafür eigene Wohnstätten und Nahrungsvorräte vernichtet sowie Landepisten und Flugplätze im Dschungel für die erwarteten Frachtflugzeuge gerodet. Mit dem Kriegsende wurden die Flughäfen verlassen und kein neues „Cargo“ wurde mehr abgeworfen. Darum bemüht, weiter Cargo per Fallschirm oder Landung zu Wasser zu erhalten, errichteten Kultanhänger Flugzeugmodelle in Originalgröße aus Stroh oder schufen Anlagen, die den militärischen Landebahnen nachempfunden waren, in der Hoffnung, neue Flugzeuge anzuziehen. Ebenso imitierten sie die Praxis, die sie bei den Soldaten, Seeleuten und Fliegern gesehen hatten. Sie schnitzten Kopfhörer aus Holz und trugen sie, als würden sie im Flughafentower sitzen. Sie positionierten sich auf den Landebahnen und imitierten die wellenartigen Landungssignale. Sie entzündeten Signalfeuer und -fackeln an den Landebahnen und Leuchttürmen. Die Kultausübenden nahmen an, die Ausländer verfügten über einen besonderen Kontakt zu den Ahnen, die ihnen als die einzigen Wesen mit der Macht erschienen, solche Reichtümer auszuschütten. Die Nachahmung der Ausländer verband sich mit der Hoffnung, auch den Einheimischen möge ein solcher Brückenschlag gelingen. Es wurde übersehen, dass diese Güter nicht einfach vom Himmel fielen, sondern in harter Arbeit hergestellt wurden. Die Konfrontation mit den vom traditionellen Leben so unterschiedlichen europäischen Gütern führte oft zu einem Zusammenbruch des ganzen Wertesystems der indigenen Völker. Soziale Strukturen wurden neu geformt, in der Hoffnung, das Paradies und die Erlösung im Diesseits zu erreichen.

Es gibt durchaus Situationen, in denen agile Projekte abgebrochen werden. Gründe dafür können sein:

- Agile Werte werden von Beteiligten, insbesondere dem Management, nicht ausreichend verstanden und mitgetragen. Manager senken im agilen Umfeld Führung und Kontrolle auf ein Minimum und schaffen Rahmenbedingungen, die die Selbstorganisation der Teams unterstützen. Konkret können sie agile Werte authentisch vorleben und dabei agilen

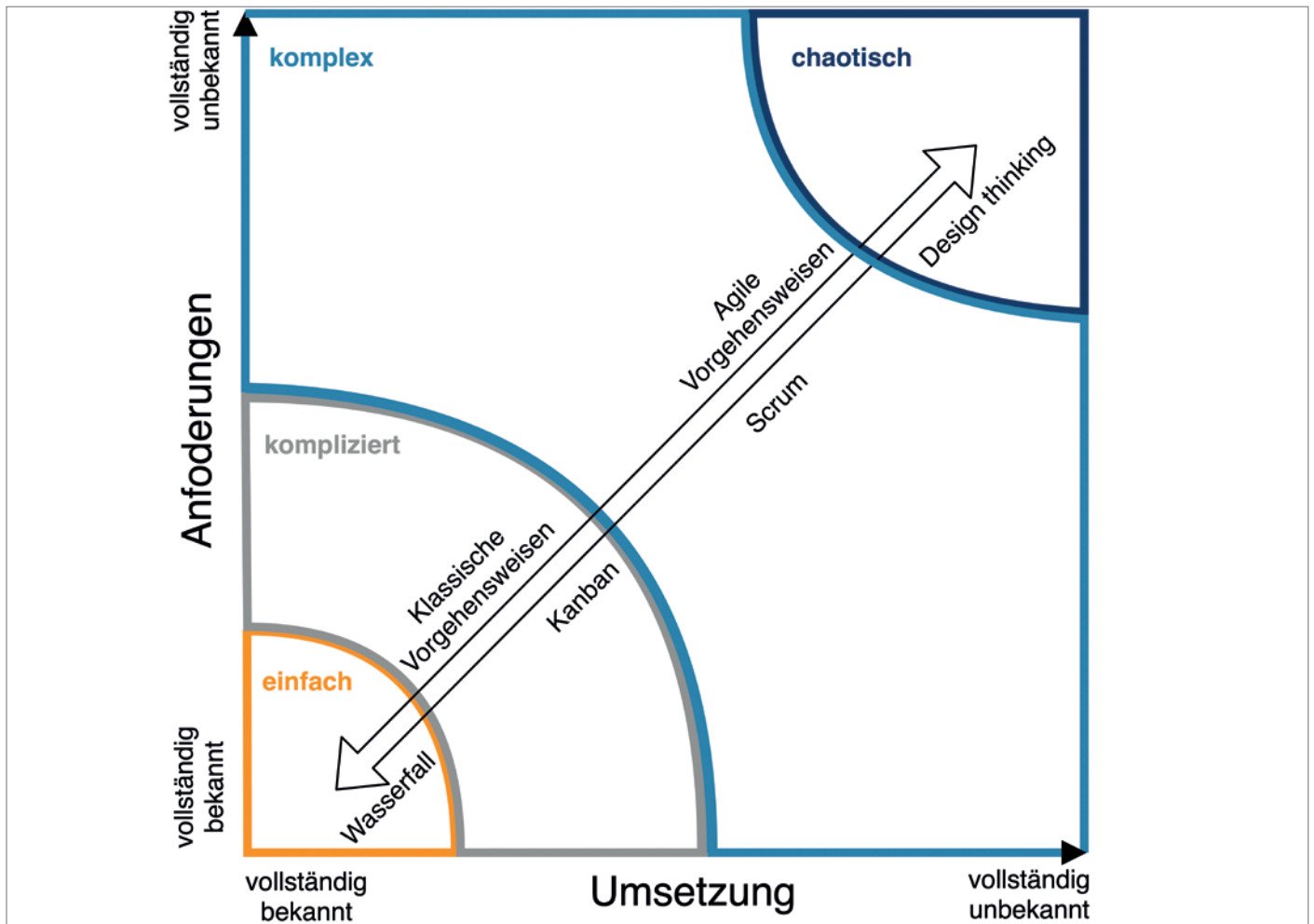


Abbildung 1: Adaption Stacey-Matrix (Quelle: Casimir Schmid)

Spezialisten wie dem Scrum Master behilflich sein. Es gilt zu fördern, nicht zu fordern. Dem Product Owner vermitteln sie Visionen auf verständliche Art und Weise und helfen, konkrete Arbeitspakete in den Backlogs abzuleiten.

- Es besteht eine zu geringe Chance auf ein Ergebnis. Ursache dafür kann eine fehlende oder unzureichende Analyse bei der Sprint-Planung sein. Es gilt, das Sprint-Ziel mit denjenigen Arbeitspaketen umzusetzen, die zu echten Werten und guten Erfolgchancen führen.
- Die Unternehmenskultur ist mit agilen Werten nicht vereinbar.

Im Vergleich zu Scrum ist Kanban in kurzer Zeit zu lernen, zeigt schnell Ergebnisse und kann risikoarm und in kleinen Schritten eingeführt werden. Auch Kanban wirkt kulturverändernd, weshalb sich beschriebene Effekte aus der Scrum-Praxis ebenfalls ergeben können, jedoch in signifikant abgeschwächter Form. Kanban setzt wie Scrum auf selbstorganisierende Teams. Je-

des Teammitglied ist unabhängig von seiner Position im Unternehmen mit seiner Meinung und seinen Verbesserungsvorschlägen willkommen. Das Team entscheidet selbst, wie es seine Arbeit erledigt, und wird ermutigt, selbstständig Probleme zu erkennen und diese zu beheben. Ein gutes Beispiel hierfür ist „Swarming“ aus der Kanban-Praxis: Alle Teammitglieder lassen ihre Arbeit augenblicklich liegen und schwärmen aus, um ein akutes Problem gemeinsam zu lösen und die Ursachen zu erkennen. Abschließend werden konkrete Arbeitspakete formuliert, mit dem Ziel, das Problem so nachhaltig zu beheben, dass es in Zukunft nicht mehr auftreten wird.

Wann agil, wann klassisch?

Zweifelsohne ist diese Frage schwierig und nicht eindeutig zu beantworten. Bei Softwareprojekten bedient man sich dafür häufig einer Adaption der Stacey-Matrix (siehe Abbildung 1) nach Ralph D. Stacey [4].

Je unbekannter Anforderung und Umsetzung sind, desto mehr eignen sich agile, je bekannter Anforderung und Umsetzung sind, desto mehr eignen sich klassische Vorgehensweisen.

In Infrastrukturprojekten werden im Wesentlichen Komponenten so zusammengesetzt, dass sich daraus ein oder mehrere Instrumente zur Wertschöpfung ergeben. Die Bandbreite reicht von statischen Komponenten mit klaren Anweisungen und gegebenen Erfahrungswerten zur Inbetriebnahme bis hin zu sehr dynamischen, gegebenenfalls eigenentwickelten Komponenten mit wenig Dokumentation und Erfahrungswerten. Vorhaben zum Aufbau von Infrastrukturen können grob in die Phasen Aufbau und Betrieb unterteilt werden, wobei der Aufbau mit zunehmendem Projektfortschritt in den Betrieb übergeht. Betrachtet man die Zusammenarbeit von Personen aus dem Infrastruktur-Bereich zu Lieferanten von Hard- und Software-Komponenten, ergibt sich ein weites Feld zwischen loser und enger Beziehung. Zu-

sammenfassend findet man sich als Mitarbeitender eines Infrastruktur-Projekts oft in vielen teils sehr unterschiedlich organisierten Schnittstellen einer Matrixorganisation zwischen unternehmens-internen und -externen Beteiligten wieder.

Ist die Beziehung eher lose und werden Komponenten nach klassischen Methoden entwickelt, eignen sich klassische Vorgehensweisen im Aufbau der Infrastruktur. Ist die Beziehung hingegen eng und wird nach agilen Ansätzen entwickelt, eignen sich eher agile Vorgehensweisen. Wichtig: Agile und klassische Methoden schließen sich nicht aus, sondern können und müssen einander aufgrund der vielfältigen Lieferanten häufig ergänzen.

Es sind also Mischformen und auf die jeweilige Komponente passende Vorgehensweisen denkbar. Als Veranschaulichung einige Beispiele:

- Architekturüberlegungen, beispielsweise zu einem Workflow-System, bei dem

Breite und Tiefe von Anwendungen, die integriert werden sollen, bestimmt werden soll. Da das Ergebnis offen ist, eignet sich die agile Vorgehensweise.

- Systemtuning, bei dem sich erst durch Instrumentalisierung unbekanntere Ursachen aufdecken lassen. Das Ergebnis ist völlig offen. Es bietet sich eine iterative, agile Vorgehensweise an.
- Bei Störungs- und Problembehebungen ist das Ziel sehr klar. Sie eignen sich für klassische Vorgehensweisen. Ermittelte Ursachen beziehungsweise technische Schulden können einem weiteren Projekt zufließen.
- Repetitive Aufgaben wie die Installation und Konfiguration von Software lassen sich ebenfalls mit klassischen Vorgehensweisen lösen. Um den zeitlichen Aufwand so gering wie möglich zu gestalten, werden sie häufig automatisiert.

Der Autor empfiehlt, zuallererst bei der Auswahl von Vorgehensweisen agil vorzugehen:

Konkret also in kleinen Schritten anzupassen, regelmäßig zu überprüfen und aufgrund gesammelter Erkenntnisse abzuändern.

Quellen

- [1] <https://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-German.pdf>
- [2] <https://www.it-agile.de/wissen/videos-1/>
- [3] <https://agilemanifesto.org/iso/de/manifesto.html>
- [4] https://en.wikipedia.org/wiki/Ralph_D._Stacey



Casimir Schmid

casimir.schmid@dbi-services.com



Performant unterwegs in heterogenen Datenbanken

Ales Zeman, Quest

Viele unterschiedliche Datenbanken gleichzeitig im Unternehmen nutzen? Das ist in vielen Unternehmen in Deutschland Alltag, stellt aber oftmals eine große Herausforderung dar. Denn wenn bereits der Betrieb einer einzigen Datenbank Ansprüche an Pflege und Erhaltung des Service-Levels stellt, die die Ressourcen gut in Beschlag nehmen, kann der simultane Betrieb mehrerer Datenbanken eine Unternehmens-IT an den Rand ihrer Kapazitäten bringen. Das liegt daran, dass die einzelnen Datenbanken für ihre Wartung jeweils eigene Werkzeuge bieten, die sich von denen anderer Anbieter unterscheiden. So müssen sich Datenbank-Administratoren (DBAs) nicht nur mit einer Vielzahl von Tools auskennen, sie müssen sie auch im laufenden Betrieb parallel im Auge behalten, damit Fehler oder Probleme sofort erkannt werden können. Dieser Prozess wird somit immer komplexer.

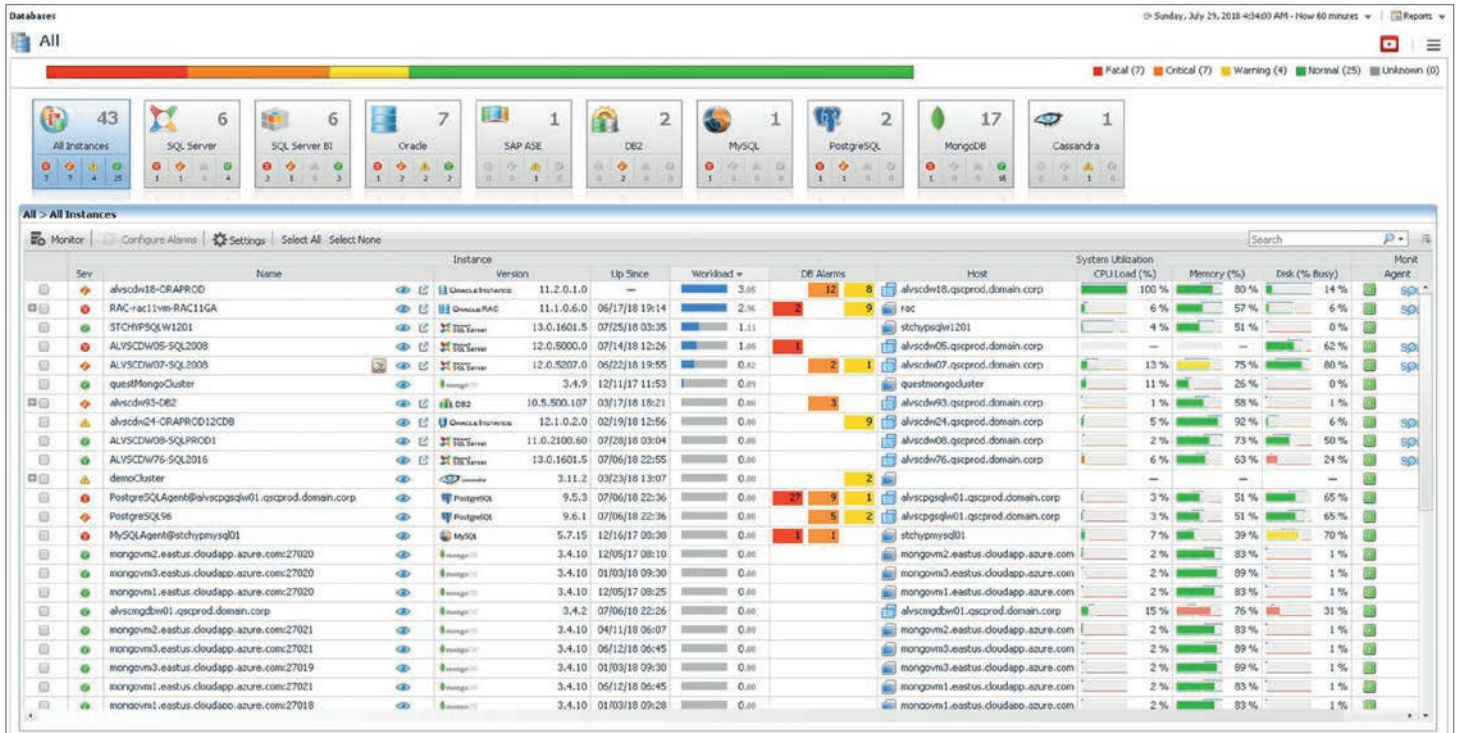


Abbildung 1: Performance-Monitoring mit Quest Foglight (Quelle: Quest)

Mehr Tools verschlingen mehr Zeit

Jedes Datenbank-Tool folgt somit einer eigenen Bedienungslogik und bietet einen für die jeweilige Datenbankplattform angepassten Funktionsumfang. Dieser kann sich von Datenbank zu Datenbank jedoch erheblich unterscheiden. Obgleich Spezialisierung durchaus einen Vorteil darstellen kann, ist dies bei der gleichzeitigen Verwaltung verschiedener Datenbankplattformen nicht der Fall. Datenbank-Administratoren müssen nicht nur zu jedem Tool geschult werden, auch das Wiederauffrischen des bereits erlangten Know-hows erfordert bei entsprechend vielen verschiedenen Werkzeugen mehr Zeit. Mit einer einfachen Schulung ist die Sache somit nicht erledigt.

Hinzu kommt, dass jeder Hersteller seine Produkte natürlich weiterentwickelt und sich diese Produktverbesserungen auch nicht analog zu anderen Anbietern verhalten. So wird der tägliche Umgang für DBAs immer komplizierter. Dabei kostet diese komplexe Arbeitsumgebung zusätzliche Zeit, die nicht gewinnbringend für das Unternehmen genutzt, sondern nur dafür aufgewendet wird, um die Datenbanken am Laufen zu halten. Sind alle DBAs ausgelastet, können Unternehmen nicht einfach neue einstellen, da der Arbeitsmarkt nicht imstande ist, diese Datenbank-Experten zu liefern.

Ein weiteres Problem ist, dass Administratoren sich bei einem begrenzten Zeitbudget auf eine Plattform konzentrieren müssen, was wiederum die Stabilität anderer Plattformen in Mitleidenschaft ziehen kann.

Tritt zusätzlich eine kritische Situation auf, können Fehler aufgrund der gleichzeitigen Verwendung mehrerer Plattformen übersehen werden. Zudem fördern unterschiedliche Benutzerschnittstellen nicht gerade die Übersichtlichkeit. So müssen sich IT-Verantwortliche mit der Frage beschäftigen, wie sie den Prozess der Verwaltung unterschiedlicher Datenbanken vereinfachen können, damit DBAs sich lohnenderen Aufgaben widmen können, wie der Weiterentwicklung ihrer Systeme.

Fünf Herausforderungen

Datenbankadministratoren sehen sich heute fünf wesentlichen Herausforderungen gegenüber. So setzt die Verwaltung unterschiedlicher Datenbankplattformen Administratoren unnötig unter Druck. Einige Unternehmen haben plattformspezifische Datenbankadministratoren, andere fordern die Einarbeitung der DBAs in neue Datenbanktypen, die aus Gründen der Kostenkontrolle, neuen Anwendungen, Unternehmenszusammenschlüssen oder Akqui-

sitionen nötig werden. Zusätzlich können verschiedene Datenbank-Editionen vorliegen, die aufgrund der Kostenersparnis angeschafft wurden. Die begrenzten Möglichkeiten der mitgelieferten Diagnose-Tools können zu einer Frustration der DBAs führen. Die Verwaltung großer Datenmengen kann eine weitere Herausforderung sein. Sie verursacht eine stark wachsende Anzahl von Datenbankinstanzen in der Organisation. Daher muss mit diesen großen und weitläufigen Umgebungen auch die Art der Verwaltung wachsen. Ferner sind verschiedene Datenbankstandorte eine weitere Herausforderung. Denn Datenbanken können sich sowohl On-Premises als auch in der Cloud befinden. Schließlich ist die Verwaltung der Performance mit minimalem Overhead die letzte Herausforderung. Den DBAs stehen viele APIs für diesen Zweck zur Verfügung. Sie müssen nun entscheiden, welches API sich für den Fall am besten eignet. Diese Auswahl kann durchaus problematisch sein, da einige APIs, die Datenbankhersteller zur Verfügung stellen, eine hohe Last auf Produktionssystemen erzeugen.

Einfach ist sicher

Wie können IT-Verantwortliche dieser Situation wirksam begegnen? Nun, der Kern des Problems ist eine immer kom-

pliziertere, gleichzeitige Verwaltung verschiedener Plattformen. Also liegt die Lösung darin, diese komplexe Umgebung zu vereinfachen. Dies kann durch die Einführung eines zentralen Datenbankmanagements geschehen. Damit ist es möglich, die Zersplitterung der administrativen Systeme zu vermeiden.

Wichtig beim Betrieb verschiedener Datenbanken ist allerdings nicht nur, dass sie fehlerfrei funktionieren, sondern auch, dass sie performant arbeiten. Gerade dieser Aspekt kann durch eine zentrale Betrachtung mit einem gemeinsamen Werkzeug für verschiedene, gleichzeitig genutzte Datenbanken adressiert werden. Denn wenn das Augenmerk der DBAs bei einer komplexen Umgebung darauf gerichtet ist, die Funktionsfähigkeit des Gesamtsystems als solches zu bewahren, bleiben Performance-Überlegungen meist außen vor.

Zentrale Werkzeuge, die verschiedene Datenbanken gleichzeitig umfassen, bieten ein Dashboard, auf dem die Stabilität, Integrität und Performance aller verwendeten Datenbanken gleichzeitig angezeigt werden. Dadurch ist eine permanente Überwachung leichter möglich.

Da die gängigen Datenbank-Formate, beispielsweise Oracle, SQL Server, DB2 oder Sybase, unterstützt werden, kann ein plattformübergreifendes Managementsystem die Datenbankpflege und Diagnose universeller durchführen als etwa die proprietären Management-Werkzeuge der einzelnen Datenbankanbieter (siehe *Abbildung 1*). Zusätzlich sind Unternehmen nicht mehr von einzelnen Herstellern abhängig, was die Verwaltung betrifft, und können ihre Administration bedeutend verschlanken.

Auswahlkriterien

Bei der Wahl eines unabhängigen Datenbankmanagementsystems sollten einige wichtige Punkte beachtet werden. So müssen Grundfunktionen wie die Verwaltung von Benutzern, Rollen, Infrastruktur und Dienstprogrammen vorhanden sein. Ferner sollten routinemäßige Wartungsaufgaben und Health-Check-Berichte durchgeführt werden können.

Eines der wichtigsten Kriterien ist außerdem, ob das Datenbankmanagementsystem dazu fähig ist, nicht nur die Leis-

tung der einzelnen Datenbanken – jede für sich – zu überwachen, sondern auch ein komplettes und lückenloses Bild der Gesamtleistung zu zeichnen.

Einen guten Mehrwert bieten Datenbankmanagementsysteme dann, wenn sie komplexe redundante Aufgaben für alle Datenbanken gemeinsam abarbeiten können. Hierzu zählen zum Beispiel Schemavergleiche, Ausführungen von Skripten, Erstellung und Verteilung von Berichten, die Optimierung des Batch-Processings von SQL-Statements oder die Verifizierung des Anwendungscodes.

Es sollte ferner selbstverständlich sein, dass das Management-Tool die Auswertung der Datenbanken in Echtzeit erledigt. So können DBAs ihre Systeme detailliert bis in die einzelnen Datenbankebenen aufschlüsseln und den Ablauf der Anwendungen schrittweise verfolgen. Natürlich ist eine Echtzeit-Auswertung auch für die Erkennung unregelmäßigen Verhaltens oder einer verringerten Leistung von Datenbanken eine Grundvoraussetzung. Wer Wert auf eine zukunftsfähige Lösung legt, sollte außerdem darauf achten, dass sich neue Plattformen beziehungsweise Datenbanken einfach in das bestehende Ökosystem eingliedern lassen.

So sollte die Lösung APIs für jede Datenbankplattform bieten, die eine ressourcenschonende und detaillierte Datensammlung sicherstellen. Zusätzlich wäre es von Vorteil, die Daten in eigenen, zentralen Repositories zu speichern.

Ein Plus für die Zusammenarbeit von Teams

Neben den Administrationsfunktionen sollte ein Datenbankmanagementsystem auch dazu in der Lage sein, die Arbeit von verschiedenen Datenbankadministratoren zu vernetzen. So können diese ihre Arbeitsschritte aufeinander anpassen und effizienter arbeiten. Dies ist beispielsweise beim Zugriff auf einen Datenpool im Rahmen eines Projekts der Fall. So können Aufgaben im Team schneller erledigt werden.

Benutzer effizient steuern

Eine weitere Aufgabe einer Managementplattform ist ferner die Steuerung von Benutzerrechten. Üblicherweise wird auch

das innerhalb der proprietären Tools der Datenbankhersteller erledigt. Jedoch können so etwa bei der Fluktuation des Personals oder auch beim Abteilungswechsel einzelner Mitarbeiter einige Anpassungen der Benutzerrechte unterbleiben, schlicht weil sie vergessen werden. Eine Managementplattform sollte diese zusätzliche Sicherheit bieten, denn dies ist auch ein Beitrag zur IT-Sicherheit und zum Datenschutz.

Fazit

Bei heterogenen und komplexen Datenbank-Umgebungen in Unternehmen sollte allein aus Gründen der Performance, der Datensicherheit und des Datenschutzes eine Evaluierung einer übergreifenden Datenbankmanagementplattform keine Frage mehr sein. Zumal die Anzahl der Aufgaben immer weiter steigt und das Angebot an Datenbankexperten auf dem Arbeitsmarkt knapp bleibt. Mit einem solchen Tool können IT-Verantwortliche ihre Systeme nachhaltig stabil und performant gestalten, während sie ihren DBAs die Freiheit geben können, zu tun, was tatsächlich strategisch wichtig ist.



Ales Zeman
Ales.Zeman@quest.com



Standardkonform oder doch Oracle-proprietär?

Jürgen Sieben, ConDeS GmbH & Co KG

Diese Folge erschien dem Autor unter dem Eindruck eines Vortrags auf der letzten DOAG-Konferenz heraus sinnvoll: Das Team von AskTom stellte Oracle-proprietäre und standardkonforme SQL-Schreibweisen vor und bezog, zum Beispiel zum Thema ANSI-Joins, eine Oracle-proprietäre Position. Eine Gegenrede.

Die Diskussion ist beinahe jedem Oracle-Entwickler bekannt. Seit Version 9 (und damit bereits seit mehreren Jahrzehnten) bietet Oracle für viele Techniken alternativ zu den proprietären auch standardkonforme Schreibweisen an. Ob nun, der Oracle-Empfehlung folgend, die standardkonforme Schreibweise oder aber, der eigenen Bequemlichkeit folgend, die langjährig geübte proprietäre Schreibweise verwendet werden soll, erregt die Gemüter. Wie wohltuend, dass in dem besagten Vortrag eine Lanze für die überkommene Oracle-Schreibweise gebrochen wurde, zum Beispiel bezüglich der Join-Schreibweise. Tenor: Auch mit der Oracle-propietären Schreibweise lässt sich alles programmieren und unter der Oberfläche wandelt der Optimizer ohnehin den ANSI-Join in die von Oracle bekannte Schreibweise um. Daher ergebe die neue Schreibweise keinen Sinn und keinen Vorteil.

Demgegenüber vertritt der Autor die Auffassung, dass man es sich so einfach (oder: so technisch) nicht machen kann. Die offensichtlichen Vorteile der ANSI-kompatiblen Schreibweise sollen nur kurz angerissen werden:

- Eine standardkonforme Schreibweise wird von mehr Entwicklern verstanden und erleichtert Migrationsprojekte
- Die explizite Angabe des gewünschten Join-Typs erleichtert das Verständnis, insbesondere bei den exotischeren Varianten NATURAL, CROSS oder FULL Join
- Der Full Join ist funktional in der Oracle-propietären Schreibweise nur durch eine Kombination von Outer Join, UNION-ALL-Abfrage und einer Anti-Join-Abfrage möglich

Weniger bekannt ist, dass auch die speziellere Form des sogenannten partitionierten Joins nur mit der Standardschreibweise möglich ist. Ein partitionierter Outer Join ist ein Join, der pro Partitionierungskriterium evaluiert wird. Dieser Join wird zum Beispiel bei Zeitreihenanalysen benötigt, um einen Outer Join zwischen den Tagen der Zeitreihe und mehreren Messobjekten für jedes Objekt getrennt evaluieren zu können. Sollen beispielsweise die Produktverkäufe dreier Produkte über eine Woche beobachtet werden, erwarten wir $3 \times 7 = 21$ Ergebniszeilen. Je nach

```
select t.time_id, p.prod_name,
       coalesce(p.amount_sold, 0) amount_sold
from times t
left join prod_sales p partition by (p.prod_name)
on t.time_id = p.time_id
```

Listing 1

```
SELECT channel_desc, TO_CHAR(SUM(amount_sold), '9,999,999,999') SALES$,
       RANK() OVER (ORDER BY SUM(amount_sold)) AS default_rank,
       RANK() OVER (ORDER BY SUM(amount_sold) DESC NULLS LAST) AS custom_rank
FROM sales, products, customers, times, channels, countries
WHERE sales.prod_id=products.prod_id AND sales.cust_id=customers.cust_id
AND customers.country_id = countries.country_id AND sales.time_id=times.time_id
AND times.calendar_month_desc IN ('2000-09', '2000-10')
AND sales.channel_id=channels.channel_id
AND country_iso_code= 'US'
GROUP BY channel_desc;
```

Listing 2a

```
SELECT ch.channel_desc,
       SUM(s.amount_sold) SALES$,
       RANK() OVER (ORDER BY SUM(s.amount_sold)) AS default_rank,
       RANK() OVER (ORDER BY SUM(s.amount_sold) DESC NULLS LAST) AS custom_rank
FROM sales s
JOIN times t      ON s.time_id = t.time_id
JOIN channels ch  ON s.channel_id = ch.channel_id
JOIN customers cu ON s.customer_id = cu.customer_id
JOIN countries co ON cu.country_id = co.country_id
WHERE t.calendar_month_desc IN ('2000-09', '2000-10')
AND co.country_iso_code = 'US'
GROUP BY ch.channel_desc;
```

Listing 2b

Datenlage werden aber weniger Zeilen ausgegeben, wenn an einem Tag Produkt A verkauft wurde, Produkt B aber nicht. In diesem Fall würde für Produkt B keine Zeile ausgegeben, weil der Outer Join in Produkt A bereits einen Treffer für diesen Tag gefunden hat. Um für alle Produkte die Verkaufszahlen an allen Tagen auszugeben, muss der Outer Join pro Produkt separat ausgewertet werden, was mittels partitioniertem Join gelingt wie in folgendem Codeausschnitt (*siehe Listing 1*):

Als Übung könnten Sie diese Abfrage ja einmal mit proprietärer Syntax formulieren.

Ein weiterer, erheblicher Vorteil der standardkonformen Schreibweise liegt darin begründet, dass der SQL-Interpreter mehr Informationen über die gewünschte Funktion besitzt und daher fehlende Join-Bedingungen als Syntaxfehler markieren kann. Unerwartete kartesische Produkte sind so unmöglich, das Zählen der Join-Bedingungen entfällt. Umgekehrt ist durch die Schreibweise CROSS JOIN ein kartesisches

Produkt jederzeit möglich und gleichzeitig explizit als gewollt dokumentiert.

Relevanter ist aus Sicht des Autors aber, dass für die SELECT-Anweisung durch die Verwendung der ANSI-kompatiblen Schreibweise nun eine Trennung von Access- und Filterprädikat möglich wird: Der Join kümmert sich ausschließlich um die technische Verknüpfung von Zeilen zu einer neuen Relation, während die WHERE-Klausel sich ausschließlich um die Filterung der neuen Relation kümmert. Gerade im Umfeld von Datenwarenhäusern, deren Star-Schemata immer gleiche Join-Bedingungen nach sich ziehen, nimmt dies viel Komplexität aus der Anweisung.

Vergleichen wir an einem Beispiel aus der Oracle-Dokumentation die Bedeutung dieser „stilistischen“ Details (*siehe Listing 2a*).

Sehen Sie, was diese Abfrage macht? Haben Sie erkannt, dass die Tabelle PRODUCTS überhaupt nicht benötigt wird? Sehen Sie sich alternativ diese Fassung in *Listing 2b* an

```
select employee_id, last_name, salary
  from employees
 order by salary desc;
```

Listing 3

```
select employee_id, last_name, salary
  from (select employee_id, last_name, salary
        from employees
        order by salary desc)
 where rownum <= 5;
```

Listing 4

Hier ist auf den ersten Blick klar, dass nur zwei Filterkriterien vorliegen, und es wird unmittelbar einsichtig, welche Access-Bedingungen nur der Kombination der Daten dienen.

Die Argumentation, der Optimizer schreibe diese Formulierung ohnehin in die proprietäre Syntax um, erscheint nicht stichhaltig: Heißt das im Umkehrschluss, dass auch Spaghetticode in PL/SQL einem strukturierten Code vorzuziehen sei, weil der Compiler Hilfsmethoden ohnehin durch das Code Inlining auflösen und an die Aufrufstelle einfügen würde? Ist nicht umgekehrt die Tatsache, dass der Compiler eine strukturierte Arbeit ermöglicht und in eine – für die Datenbank optimierte – Form umsetzt, ein wesentlicher Vorteil, da der Entwickler nun sauber strukturierten Code erstellen kann, ohne die Performance nachteilig zu beeinflussen?

Dies wird an einem weiteren Beispiel deutlich: Eine Top-N-Analyse, zum Beispiel zu den Top-5-Verdienern, würde im einfachsten Fall so ausgeführt (*siehe Listing 3*).

Diese Abfrage sortiert nun die Mitarbeiter absteigend nach Gehalt. Ein Zugriff, zum Beispiel über das Netzwerk durch eine Anwendung, greift sich hiervon die ersten 5 Ergebniszeilen und schließt anschließend den Cursor. Nicht alle Anwendungsentwickler können direkt den Vorteil folgender Schreibweise benennen (*siehe Listing 4*).

Bekanntlich liegt der Vorteil dieser Schreibweise darin, dass der Optimizer nun ein ORDER BY STOPKEY genanntes Verfahren einsetzen kann, um die Anzahl der zu sortierenden Zahlen zu reduzieren. Hierbei werden – das für die Leser, die diese Option nicht kennen – nur die zufällig ersten fünf Zeilen der Tabelle sor-

tiert, die sechste und alle folgenden Zeilen jedoch nur daraufhin überprüft, ob deren SALARY-Wert größer als der momentan niedrigste Gehaltswert der fünf sortierten Zeilen ist. Ist dies so, wird das Gehalt integriert, ansonsten wird die Information ignoriert. Gerade bei größeren Tabellen ist dieser Vorteil eklatant, da stets nur fünf Zeilen sortiert vorgehalten werden müssen und sich die Sortierung immer nur auf diese wenigen Zeilen beschränkt. Lassen Sie uns nun ein wenig springen und die Schreibweise der Datenbankversion 12 für solche Abfragen verwenden (*siehe Listing 5*).

Diese Schreibweise ist nicht nur einfacher zu verstehen (und, ja, auch standardkonform), sondern wartet mit einem weiteren Vorteil auf: Der Optimizer schreibt diese Abfrage, abhängig von der optionalen Klausel ONLY oder WITH TIES, nun zu folgender Abfrage um (*siehe Listing 6*).

Dies wäre die Version für die Klausel WITH TIES, bei der auch mehr als 5 Zeilen als Ergebnis geliefert werden können, wenn mehrere Mitarbeiter auf Rang 5 des Gehaltsrankings lägen. Mit

```
select employee_id, last_name, salary
  from employees
 order by salary desc
 fetch first 5 rows [only|with ties];
```

Listing 5

```
select employee_id, last_name, salary
  from (select employee_id, last_name, salary,
              rank() over (order by salary desc) rg
        from employees)
 where rg <= 5;
```

Listing 6

der Klausel ONLY würde die analytische Funktion ROW_NUMBER verwendet. Diese Schreibweise kennen noch deutlich weniger Entwickler, und diese Form hat den Vorteil, dass nun überhaupt keine Zeilen mehr sortiert werden, sondern ein WINDOW-SORT-Verfahren verwendet werden kann, dass gegenüber dem ORDER-BY-STOPKEY-Verfahren noch einmal effizienter ist.

Warum also sollte man die ROW-LIMITING-Klausel verwenden, wenn man doch auch die Variante mit der analytischen Funktion schreiben könnte? Nehmen wir das Extrembeispiel zur Verdeutlichung: Hätten Sie die allererste Variante geschrieben, hätte die Datenbank keine Chance, eine Optimierung für Sie einzurichten. Erst durch die Verwendung der ROW-LIMITING-Klausel (oder der Unterabfrage) haben Sie der Datenbank mitgeteilt, dass Sie nur an den besten fünf Ergebnissen interessiert waren. Nachdem der Datenbank diese Information vorliegt, konnte sie nun also den besten Ausführungsplan berechnen. Und sollte in Datenbankversion 20 eine noch bessere Implementierung für dieses Problem bereitstehen, können Sie davon ausgehen, dass der Optimizer die modernste Version dieser Abfrage auch auf diese neue Technik hin optimieren wird.

An weiteren Beispielen lässt sich zeigen, dass die standardkonforme Schreibweise mit Vorteilen gegenüber der Oracle-proprietären Schreibweise verbunden ist: So lässt die Verwendung der COALESCE- gegenüber der NVL-Funktion nicht nur mehr als zwei Parameter zu, sondern implementiert auch noch einen Short Circuit, der dafür sorgt, dass der



2020
DOAG
Konferenz + Ausstellung

SAVE THE
DATE

17. - 20. Nov 2020



zweite (oder jeder weitere) Parameter nur noch dann evaluiert wird, wenn die vorhergehenden Parameter NULL-Werte enthalten. Insbesondere bei der Verwendung von PL/SQL-Funktionen als weitere Parameter profitieren Sie hier von einer verbesserten Performanz.

In die gleiche Kerbe schlägt der Ersatz der DECODE-Anweisung durch die konsistente Verwendung der CASE-Anweisung. Man mag argumentieren, dass die DECODE-Anweisung unschlagbar kurz und prägnant formuliert, was gefordert wurde, doch sind die Limitierungen dieser Funktion eben auch erheblich: In der auswertenden CASE-Anweisung lassen sich komplexe boolesche Ausdrücke und nicht nur einfache Wertvergleiche wie bei der DECODE-Anweisung durchführen. Zudem ist die CASE-Anweisung in PL/SQL verfügbar, die DECODE-Anweisung jedoch nicht.

Diese Beispiele zeigen, dass nicht nur wichtig ist, was „hinten herauskommt“, sondern auch, wie der Weg dorthin formuliert wurde. Zwar ist richtig: Wenn Sie alle Optionen und die jeweiligen Verbesserungen ken-

nen, können Sie natürlich die gleiche Performanz auch mit einer Oracle-proprietären Schreibweise erreichen; nichts (oder besser: kaum etwas) geht nur deshalb schneller, weil Sie eine standardkonforme Formulierung gewählt haben.

Und, um ehrlich zu sein, ist auch der Autor nicht in allen Belangen so konsequent, die jeweilige Standardversion für ein Abfrageproblem zu verwenden: So ist zum Beispiel die seit vielen Jahrzehnten bekannte CONNECT-BY-Abfrage auch im Vergleich zur ISO-kompatiblen Variante mit rekursiver WITH-Klausel immer noch unschlagbar elegant und leicht verständlich. Auch die DECODE-Anweisung hat die nützliche Eigenheit, NULL-Werte auf Gleichheit prüfen zu können, was in einigen Problemstellungen ein wirklicher Vorteil ist. Auch beim händischen Pivotieren von Daten macht die DECODE-Anweisung eine gute Figur, weil sie die eher triviale Fallunterscheidung dieses Abfragepatterns eben mit geringstmöglichem Code-Aufwand bewerkstelligt. Dadurch ist dieses Verfahren besser lesbar.

Dies sollte die eigentliche Messlatte zu sein: Sagen Sie der Datenbank so einfach, elegant und klar wie möglich, welche Daten Sie für eine gegebene Fragestellung benötigen. Wenn einmal eine Gewöhnung bezüglich der Verwendung standardkonformer Funktionen eingetreten ist, werden Sie feststellen: Ihre Abfragen profitieren. Daher sollten Sie der neuen Schreibweise eine Chance geben und sie nicht nur deshalb ablehnen, weil Sie seit Jahrzehnten andere Schreibweisen gewohnt sind.



Jürgen Sieben
j.sieben@condes.de

Der DOAG Projektpreis 2019 geht an fabe!

Marcos López, Redaktionsleitung Business News

Auf der DOAG 2019 Konferenz + Ausstellung wurde der Gewinner des DOAG Projektpreises vorgestellt.

Das Rennen machte die Non-Profit Organisation fabe. Die Entscheidung der Jury fiel dabei einstimmig aus. Fabe steht für for all a beautiful earth und ist eine innovative, mobile App, die den Menschen dabei helfen soll, gegen Klimawandel, Umweltzerstörung und Artensterben zu kämpfen. Dabei hat sich fabe drei r's auf die Fahne geschrieben: reduce (reduzieren), rescue (retten) und reconnect (besinnen).

Der Projektpreis wurde im Sommer 2019 ins Leben gerufen. Bis zum 16. September konnten Unternehmen ihre digitalen Projekte in den Ring werfen. Die Bedingungen: Umsetzung mit intensivem Einsatz von Oracle-Technologien und nicht älter als zwei Jahre. Der Preis: 2.000 Euro und drei Tickets für die DOAG 2019 Konferenz + Ausstellung. Der Projektpreis wurde in den Kategorien Applications und Technologie ausgelobt und

richtete sich explizit an Firmen aus dem deutschsprachigen Raum. Die Sichtung der eingereichten Beiträge und die letztliche Auswahl der Gewinner erfolgte durch die sechsköpfige Projektpreis-Jury, der Vertreter des DOAG-Vorstands und der DOAG-Communities sowie unabhängige Experten angehörten.

Die Gründer von fabe, Steven Feuerstein und Vincent Morneau, hatten sich bei der Umsetzung für eine PWA als Lö-

sung entschieden, einer Progressive Web App. Neben APEX wurde die Oracle Enterprise Datenbank in der Cloud unter anderem mit dem Tuning und Diagnostic Pack genutzt. E-Mails wurden in der Datenbank generiert und aus der Oracle Cloud gesendet. Das DB-Modell wurde mit dem SQL Data Modeler erzeugt, während der SQL Developer der entscheidende Faktor bei der Umgebungsentwicklung war. Die Technologie Oracle Rest Data Services diente schließlich zur Bereitstellung und Verwendung von Datenbankinhalten.

Fabe kann als Informationsquelle für viele Aktionen zum Artenschutz und gegen die Klimazerstörung benutzt werden. Dabei können Nutzer auch eigene Aktionspläne erstellen und durch die App und die Community motiviert werden, noch mehr zu tun und am Ball zu bleiben. Denn: „Wenn eine Million Menschen auf den Kauf von Wasser in Plastikflaschen verzichten, dann könnte dies auch die Produktion und den Handel beeinflus-

sen“, so die Macher in ihrem Projekt-Prospekt.

Für Niels de Bruijn, Vorstand Development bei der DOAG, war nach Sichtung der Einreichungen fabe der klare Favorit: „Ich finde es eine großartige Initiative, die uns inhaltlich alle angeht. Das Projekt wurde ausschließlich mit Freiwilligen umgesetzt – und mit aktueller Oracle-Technologie!“ Michael Paege, stellvertretender Vorstandsvorsitzender bei der DOAG, konnte sich dem nur anschließen: „Die Summe der Argumente für fabe war herausragend: Das weltumspannende Team mit seiner Basis in

Deutschland, die Ehrenamtlichkeit der Mitarbeiter, die Benutzerfreundlichkeit der App – und das Ganze mit modernster Oracle-Technologie.“ Herzlichen Glückwunsch, fabe!





Senkung der Betriebskosten durch Migration auf Open Source? Ja, aber ...

Grégory Steulet, dbi services

Als Partner zahlreicher Softwarehäuser, die proprietäre oder Open-Source-Software anbieten, wird dbi services immer wieder auf die verschiedenen angebotenen Softwarelösungen angesprochen. Zu unseren Partnern zählen unter anderem EnterpriseDB, Severalnines, Cloudera, Red Hat, MariaDB, aber auch Oracle, Microsoft und Dbvisit, die alle drei proprietäre Software herausgeben.

Der Aufstieg von Open-Source-Software

Seit einigen Jahren beobachtet dbi services bei vielen Unternehmen die zunehmende Popularität von Open-Source-Software, unabhängig von der Branche oder Größe des Unternehmens. Das bedeutet nicht, dass diese Unternehmen sich vollkommen von Anbietern proprietärer Software abwenden. In den meisten Fällen koexistiert Open-Source-Software mit proprietärer Software.

Heute setzen selbst Unternehmen, die früher niemals daran gedacht hätten (zum Beispiel im Bankensektor), ihre Anwendungsumgebungen auf anderen als AIX-Plattformen zu betreiben, Linux-Systeme ein – und das auch in Umgebungen, die im Hinblick auf Leistung und Verfügbarkeit keinerlei Kompromisse dulden. Die großen Softwarehäuser gelten nach wie vor als Garant für Langzeitbestand, allerdings treten die ursprünglichen Entscheidungskriterien angesichts aktueller Randbedingungen wie Time-to-Market, Agilität und Interoperabilität in den Hintergrund. Um Open Source ist ein regelrechter Hype entstanden, der nicht nur in wachsender Nachfrage, sondern auch in einem steigenden Angebot an Beiträgen bei internationalen Events wie

der DOAG-Konferenz (Konferenz der Deutschen Oracle-Anwendergruppe) zum Ausdruck kommt. Selbst Anwendergruppen wie die SOUG (Swiss Oracle User Group) zeigen sich bei ihren Veranstaltungen offen für Open-Source-Technologien, die nicht von Oracle stammen, um weitere Teilnehmer zu gewinnen.

Relationale Datenbanken als Beispiel

Open Source betrifft alle Schichten der Software-Infrastruktur und auch Datenbanken bilden hier keine Ausnahme. Viele unserer Kunden, die Oracle oder Microsoft verwenden, setzen heute auch PostgreSQL ein. Auch MySQL und seine Abspaltungen (Forks) wurden schon immer auf breiter Ebene genutzt (wenn auch teilweise ohne Wissen des Kunden), da diese Systeme in zahlreichen Softwarelösungen, wie beispielsweise Content-Management-Systemen (CMS), Dokumenten-Management-Systemen (DMS), aber auch Anwendungen im Bereich Customer-Relationship-Management (CRM), integriert sind. MySQL verliert in letzter Zeit zwar etwas an Popularität – zweifelsohne zugunsten von MariaDB und Percona –, behauptet sich jedoch laut db-engines.com im

Mai 2019 immerhin auf dem zweiten Platz der beliebtesten Datenbanken.

Unsere Erfahrung im Bereich von IT-Infrastrukturen hat uns gezeigt, dass hinter der Entscheidung für Open Source – sofern sie nicht auf branchenspezifischen Anforderungen beruht – häufig der Wunsch steht, die Investitions- und Betriebskosten zu senken. In diesem Fall dient die Perspektive einer eventuellen Nutzung von Open-Source-Software gerne als Argument bei den Verhandlungen mit den großen Softwarehäusern.

Beim Umstieg von einer proprietären Lösung auf eine Open-Source-Lösung sind tatsächlich Einsparungen möglich, ebenso wie beim Umstieg von einer proprietären Lösung auf eine andere proprietäre Lösung. Allerdings ist „Open Source“ nicht gleichbedeutend mit „kostenlos“. Dies gilt umso mehr, wenn es um Lösungen geht, bei denen es in hohem Maße auf Leistung und Verfügbarkeit ankommt. Aber wie hoch sind dann die tatsächlichen Kosten von Open Source?

Die Kosten von Open Source

Die Antwort auf die Frage nach den Kosten für Open Source lautet ganz klar „Je

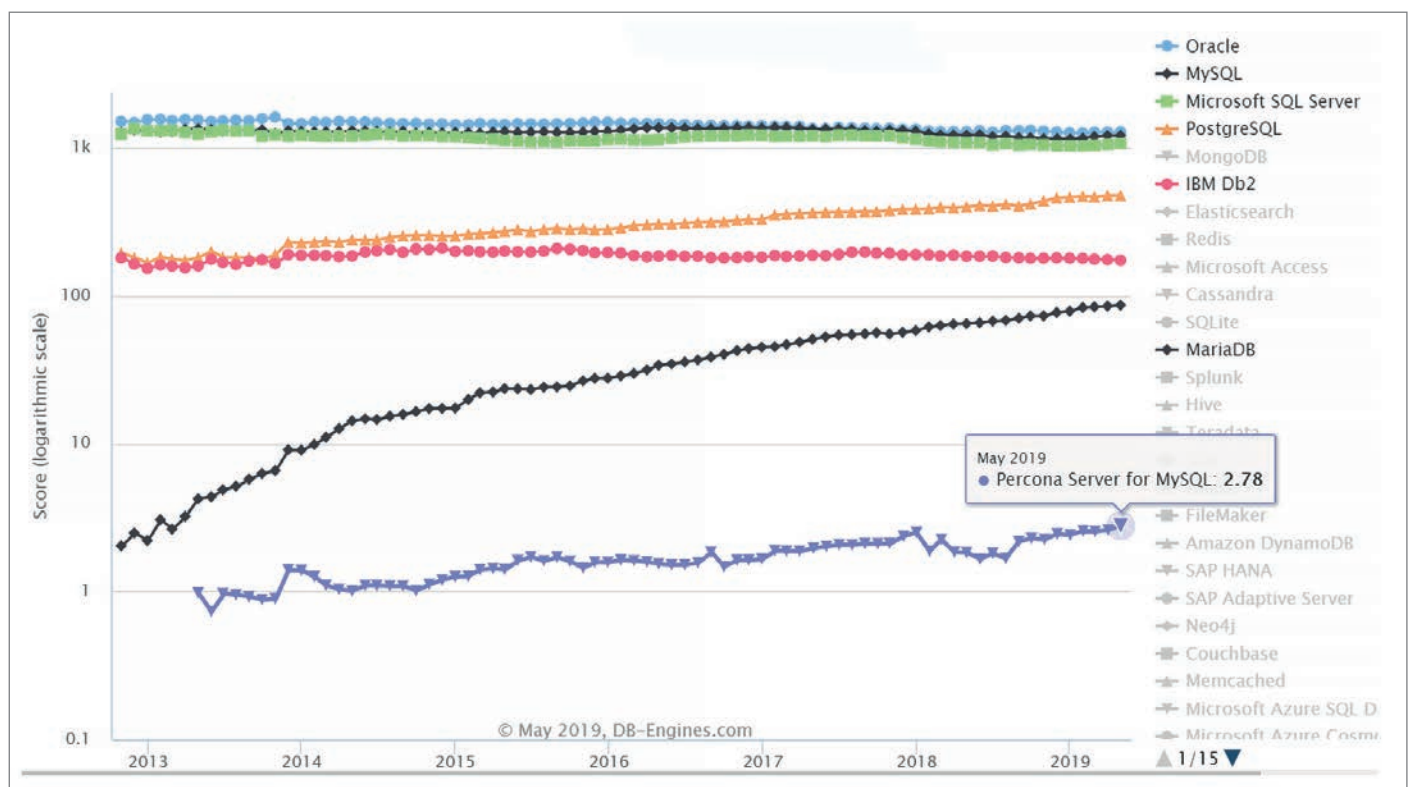


Abbildung 1: DB Engines Ranking (Quelle: Grégory Steulet)

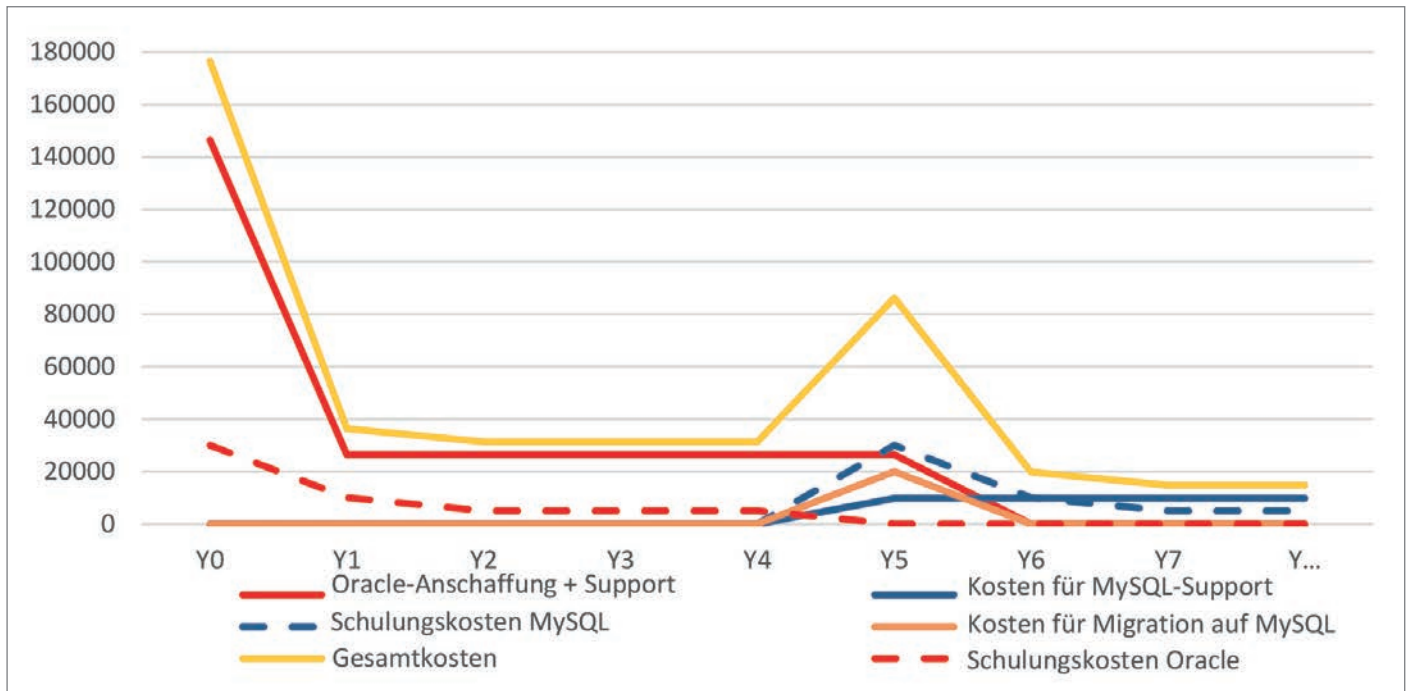


Abbildung 2: Kosten des Open-Source-Systems (Quelle: Grégory Steulet)

nachdem...". Manche Kosten müssen jedoch vollkommen kontextunabhängig in jedem Fall eingeplant werden, insbesondere die großen Kostenfaktoren:

- Kosten für Schulung der Teams
- Kosten für Migration
- Kosten für Support der Lösung

Open-Source-Systeme bringen nicht zu vernachlässigende Investitionskosten mit sich, die sich in den Folgejahren auch in den Betriebskosten und mittelfristig in einer gemischten Bilanz niederschlagen. Die durch den Einstieg in oder die Migration auf ein Open-Source-System bedingte Investition erreicht ihre Rentabilitätsschwelle meist erst nach einigen Jahren.

Hypothese: Migration von Oracle auf MySQL

Nehmen wir ein einfaches Beispiel, um die Migration von Oracle auf MySQL zu veranschaulichen. Wir wollen hierbei außer Acht lassen, dass die Funktionen von Oracle und MySQL nicht direkt vergleichbar sind. Auch PostgreSQL hätte für dieses Beispiel herangezogen werden können, insbesondere bei Nutzung von gespeicherten Prozeduren. Wie gehen also davon aus, dass keine weitere Funktionalität benötigt wird, die nicht im Leistungs-

umfang von MySQL enthalten ist, und dass die aktuelle Oracle-Infrastruktur auf zwei „Oracle- Prozessoren“ (Intel Xeon 4 Core x 0.5 Intel Core Factor (Oracle) = 2 Oracle-Prozessoren) beruht.

Eine Oracle-Prozessor-Lizenz kostet 47 500 \$ (Preisliste), insgesamt gehen wir also von einem Gesamtbetrag von 95 000 \$ aus. Bei dieser Installation müssen der Diagnostics Pack und der Tuning Pack hinzugefügt werden, mit einem Betrag von 2 x 7 500 \$ + 2 x 5 000 \$ = 25 000 \$. Die Anschaffungskosten belaufen sich also auf 120 000 \$. Dazu kommen die Kosten für den „Support im 1. Jahr“ in Höhe von 22 % des Betrags, also 26 400 \$. Die Gesamtkosten im ersten Jahr belaufen sich damit auf 146 400 \$, in den Folgejahren sind jeweils 22 % der Gesamtkosten für den Support einzukalkulieren, also 26 400 \$ (ohne Teuerungszuschlag, der von Oracle jedes Jahr berechnet wird).

Nach fünf Jahren wird die Entscheidung getroffen, die Datenbank auf ein Open-Source-System (Oracle MySQL oder MariaDB) zu migrieren. Es fallen keine Anschaffungskosten an. Die Supportkosten betragen 9 600 \$ (MariaDB TX) oder 5 000 \$ (Oracle MySQL 1-4 Sockets) und 3 000 \$ für eine Provisioning- und Monitoring- Lösung wie ClusterControl von Severalnines (für das Monitoring beinhaltet MariaDB Enterprise Monyog und MySQL Enterprise den MySQL Enterprise Monitor), mit

Kosten im Bereich von 5 000 \$ bis 12 600 \$ pro Jahr.

Dazu kommen selbstverständlich die eigentlichen Migrationskosten (die erheblich von der bestehenden Infrastruktur abhängen und hier pauschal mit 20 000 \$ veranschlagt werden) sowie eventuelle Kosten für die DBA-Schulung in MySQL. Zur besseren Vergleichbarkeit wird diese Schulung ebenso wie die Oracle-Schulung mit 30 000 \$ im ersten Jahr, 10 000 \$ im zweiten Jahr und 5 000 \$ in den folgenden Jahren angesetzt. Diese Kosten decken die für Selbstlernprogramme und externe Schulungen aufgewendete Zeit.

In unserem Fall wurde zur Sicherheit beschlossen, den Oracle-Support während der Migration für ein Jahr beizubehalten. Dadurch erhöhen sich die Betriebskosten im Vergleich zu den Vorjahren. Allerdings rechtfertigt sich diese Entscheidung in der Regel durch die Möglichkeit, während der Übergangszeit und in den ersten Monaten nach der Migration weiterhin auf die alte Infrastruktur zugreifen zu können. Dieser Posten sollte bei der Aufstellung des Migrationsbudgets unbedingt berücksichtigt werden.

Während des Migrationsjahres sind die Betriebskosten möglicherweise höher. Dies sollte im Jahresbudget entsprechend eingeplant werden. Ausgehend von unserem Beispielfall kann die Rentabilitätsschwelle rund zweieinhalb Jahre nach erfolgter Migration erreicht werden.

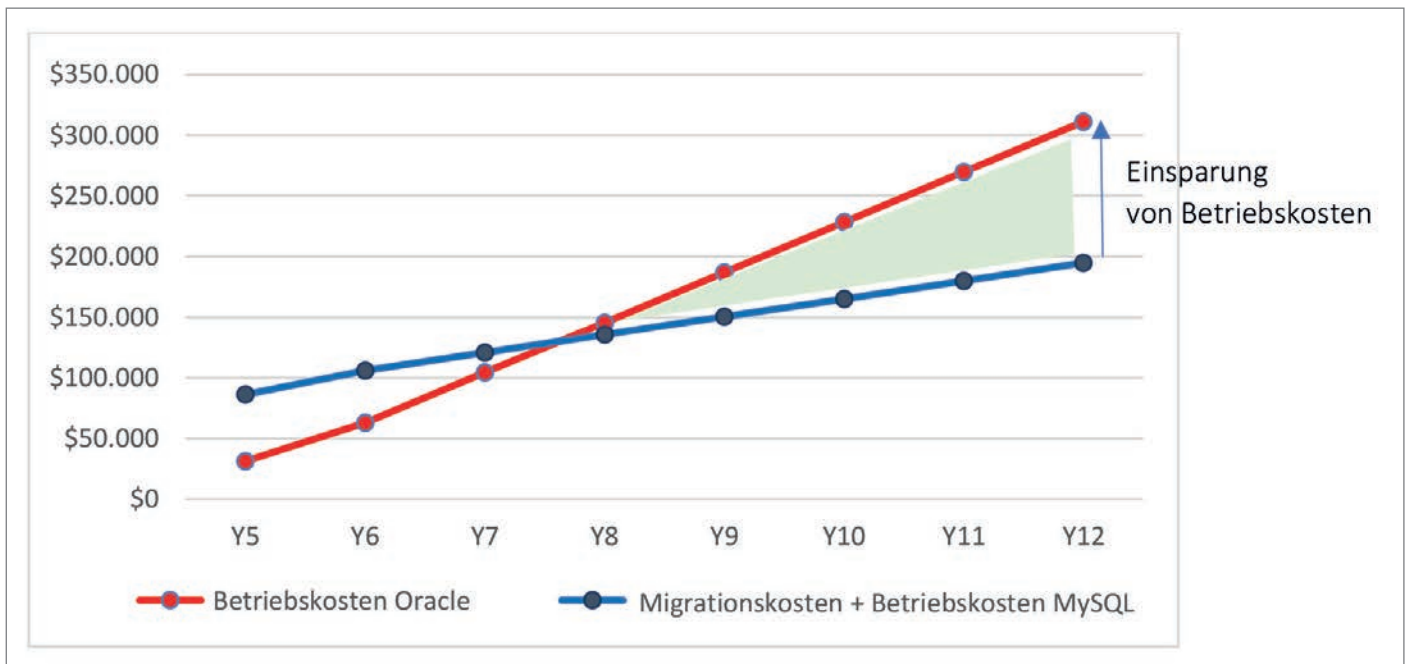


Abbildung 3: Rentabilitätsschwelle (Quelle: Grégory Steulet)

Ist Open Source zu kostenaufwendig?

Immer häufiger sind IT-Manager gefordert, mit weniger mehr zu leisten. Vor diesem Hintergrund ist die Wahrscheinlichkeit groß, dass die Geschäftsführung auf das IT-Management zugeht und die Kosten des Open-Source-Systems mit der imperativen Zielsetzung anprangert, diese Kosten nur wenige Jahre nach der Migration zu senken. Die Zeit vergeht und wir neigen dazu, in der Vergangenheit erzielte Einsparungen schnell zu vergessen und nur noch die laufenden Aufwände zu sehen.

Wie alles andere, gehen jedoch leider auch Open-Source-Systeme mit Kosten einher. Das Budget für IT-Produkte und IT-Dienstleistungen muss jedes Jahr angepasst und mit der Qualität, Sicherheit und Verfügbarkeit der Services in Einklang gebracht werden. Unterhalb einer bestimmten Schwelle wird die Servicequalität direkt beeinträchtigt. Das IT-Management hat die Aufgabe, immer wieder herauszustellen, wie wichtig es ist, dieses Qualitätsniveau zu wahren und die kritische Schwelle, ab der diese Qualität gefährdet ist, nicht zu unterschreiten. Ausfälle und mangelnde Verfügbarkeit schlagen sich ebenfalls als Kostenfaktor im Budget nieder und werden bei Datenverlusten oder Nichtverfügbarkeit der Dienste einzig dem IT-Management zugeschrieben.

Kurz zusammengefasst

Open Source wird allzu häufig mit Kostenfreiheit gleichgesetzt. Eine Migration auf Open-Source-Software ist jedoch keinesfalls kostenlos. Bevor Sie durch den Umstieg Betriebskosten sparen können, müssen Sie diese in einem ersten Schritt vielmehr erhöhen. Um die für eine Infrastruktur in der Produktion erforderliche Verfügbarkeit gewährleisten zu können ist, benötigen Sie Support, welcher – in jedem Fall – kostenpflichtig ist. Kaum ein IT-Manager käme auf die Idee, die IBM-Architektur P-Series oder eine Oracle-Datenbank ohne Support einzusetzen. Bei Open-Source-Software ergibt sich diese Situation jedoch recht häufig, sei es bei Nutzung einer PostgreSQL-Datenbank, des Betriebssystems Linux oder eines CRM-Systems wie SugarCRM. Leider bedingt der fehlende Support unweigerlich reduzierte Servicequalität.

Letztendlich entstehen bei der Migration auf Open-Source-Software zusätzlich indirekte Kosten durch Schulung, Migrationsaufwand und in den meisten Fällen doppelten Support während der Übergangszeit. Diese Investition wird aller Wahrscheinlichkeit nach erst nach einigen Jahren wirtschaftlich rentabel. Es ist also Vorsicht angeraten, in dieser Hinsicht trügerische Erwartungen anzuheizen. Bei Ihren Überlegungen sollten Sie sich jedoch auch von anderen Faktoren

leiten lassen. Open-Source-Software erlebt einen Hype sondergleichen. Mit einer Stellenausschreibung im Bereich der Open-Source-Technologie treffen Sie sicherlich auf höheres Interesse als mit einer Stellenausschreibung im Bereich proprietärer Technologie.

Kurz gesagt, Open Source ist wunderbar – aber nicht kostenlos.



Grégory Steulet
Grégory.Steulet@dbi-services.com

BUSINESS NEWS

NEWS



Agile Vorgehensmodelle in Serviceorganisationen

Carsten Wiesbaum, esentri AG
Ingo Sobik, esentri AG

Was für eine Assoziation springt Ihnen als erstes in den Sinn, wenn Sie an die Begriffe Deutschland und Service denken? Bei vielen Menschen ist es der Begriff „Servicewüste“, der das völlige Fehlen akzeptabler Dienstleistungen beschreibt. In diesem Artikel möchten die Autoren mögliche Gründe für diese Wahrnehmung der Gesellschaft aufzeigen und beschreiben, wie agile Denkweisen und Vorgehensmodelle helfen können, die Servicequalität zu verbessern.

Deutschland ist für einiges bekannt. Seien es unsere Autos, die deutsche Pünktlichkeit oder die Gewissenhaftigkeit, mit der Aufgaben erledigt werden. Ein Bereich, in dem die Deutschen aber so gar nicht zu glänzen scheinen, ist der Dienstleistungs- und Sektorservice. Nicht umsonst gab es die Wortschöpfung der „Servicewüste“, welche die gefühlte Qualität der Services in Deutschland beschreibt. Dies ist umso erstaunlicher, wenn man sich den aktuellen Anteil des Dienstleistungssektors an der Bruttowertschöpfung Deutschlands im Jahr 2018 anschaut. Dieser hat einen Anteil von knapp 68% [1]. Das produzierende Gewerbe, das die hierzulande und im Ausland geschätzten deutschen Produkte erzeugt, war 2018 lediglich für gut ein Viertel (25,8%) der deutschen Bruttowertschöpfung verantwortlich. Im Zeitraum von 1991 bis 2018 ist dieser Anteil sogar um 5% gefallen, wohingegen der Anteil des Dienstleistungssektors um etwa 6% gestiegen ist.

Der Trend scheint sich somit entgegengesetzt unserer historisch starken Wirtschaftsbereiche hin zu den eher schwach ausgeprägten Sektoren zu entwickeln. Grund genug sich darüber Gedanken zu machen, wie wir aus der Servicewüste eine Service-Oase machen können. Aber wie ist die Servicewüste überhaupt entstanden? Diese Frage ist weder einfach noch allgemeingültig zu beantworten. Im Arbeitsalltag haben die Autoren jedoch ein wiederkehrendes Muster bei Serviceorganisationen erlebt, welches ein Aspekt für den aktuellen Stand der Servicequalität darstellen kann.

Das Ziel von Serviceorganisationen ist häufig, einen möglichst stabilen Prozess zu erreichen. Dieser wird kontinuierlich auf Durchsatz und die Reduzierung der Kosten für die Verarbeitung eines Kundenanliegens optimiert. Bei dieser Optimierung sind die einzelnen Prozessschritte klar vorgegeben. Im Optimalfall liegen außerdem digitale und strukturierte Eingangsdaten vor. Somit kann die komplette Behandlung eines Kundenanliegens automatisiert werden. Häufig fehlen jedoch bestimmte Daten, oder sie liegen nicht in einer geeigneten Form vor. In diesen Situationen greifen die Mitarbeiter der Organisationen ein. Diese haben klare Vorschriften, wie Kundenanliegen abzuarbeiten sind. Es ist sogar häufig so, dass es klare Zeitvorgaben und zu erfüllende Quoten für jeden einzelnen Mitarbeiter gibt. Die Erreichung der Zielvorgaben und Quoten wird oft mit KPIs (Key Performance Indicator) überwacht und zur Bewertung von Leistungen (Mitarbeiter, Team, Abteilung) und weiteren Optimierungsmöglichkeiten genutzt. Als Ergebnis arbeiten Mitarbeiter oft, um den Prozess und seine KPIs zu bedienen. Dabei treten der Kundenfokus und die zufriedenstellende Bearbeitung des Kundenanliegens oft in den Hintergrund.

Die Grundprinzipien dieser Art der Unternehmensführung sind allgemein hin als Scientific Management [2] oder auch Taylorismus bekannt. Seine ersten Werke zu dem Thema veröffentlichte Frederick W. Taylor im Kontext des produzierenden Gewerbes zu Beginn des zwanzigsten Jahrhunderts. Dabei gab es eine direkte Korrelation zwischen der produzierten Menge und dem

möglichen Umsatz eines Unternehmens. Die Grundidee dabei war, dass es für die Produktion eines Guts einen optimalen Herstellungsprozess gibt. Diesen galt es zu finden, um die Produktionskapazitäten zu optimieren. Die Definition des Prozesses wurde in einem Planungsbüro verantwortet. Hier wurde der „One Best Way“ definiert und entsprechend der vorhandenen Ressourcen geplant. Jeder Schritt im Produktionsprozess wurde detailliert beschrieben und dessen Ausführung kontrolliert. Für jeden Produktionsschritt waren der Leistungsort und die dafür zur Verfügung stehende Zeit klar definiert.

Die beschriebenen Prinzipien haben zu Zeiten Taylors und innerhalb des produzierenden Gewerbes gut funktioniert, waren betriebswirtschaftlich einleuchtend und einfach zu vermitteln. Durch die damals langen Entwicklungs- und Lebenszyklen von Produkten konnten diese über Jahre produziert werden. So konnte der „One Best Way“ seine volle Wirkung erzielen. Zusätzlich gab es nur selten Änderungen an einem entwickelten Produkt, der Kunde musste akzeptieren, was er bekam und sich an das Produkt anpassen.

Nach Meinung der Autoren ist eine Problematik im Dienstleistungssektor, dass eben jene Prinzipien, die ein stabiles Umfeld und ein einheitliches Produkt in den Mittelpunkt stellen, genutzt werden, um Wertschöpfungsketten für Serviceprodukte zu optimieren. Damit werden nicht der Kunde und sein Anliegen in den Mittelpunkt gestellt, sondern ein nach innen gerichteter Standpunkt eingenommen. Es geht in ers-

ter Linie darum, Anliegen einheitlich und möglichst effizient zu verarbeiten. Gemessen werden Mitarbeiter nach Zeitvorgaben und der Anzahl abgearbeiteter Fälle. Ob das Kundenanliegen sinnvoll behandelt wurde und der Kunde zufrieden ist, geht dabei schon einmal gerne unter. Hinzu kommt, dass sich der Markt und die Erwartungshaltung der Kunden stark gewandelt haben.

Das Zeitalter von Plattformen und VUCA

Der Markt, in dem Unternehmen heutzutage agieren, hat sich stark verändert. Bei vielen der heutigen Produkte handelt es sich um Software, oder ein Großteil der Produktfunktionalität wird durch Software bereitgestellt. Produkte sind heutzutage oft Plattformen, die der Kunde an seine eigenen Bedürfnisse anpassen kann. Dies geschieht durch Erweiterungen des Produktes, zum Beispiel durch Apps auf einem Smartphone oder Skills bei Smart-Speakern. Für jede Aufgabe gibt es dabei nicht nur eine Möglichkeit der Erweiterung, sondern viele. Benötigt der Kunde eine Funktion auf seinem Smartphone für seine Aufgabenverwaltung, kann er einfach in den entsprechenden Store gehen und danach suchen. Sofort wird ihm eine Vielzahl an Apps angezeigt, die er nacheinander testen kann. Dabei installiert man schnell einmal mehrere Apps, probiert sie kurz aus und wählt dann die aus, die für die eigenen Zwecke am besten passt und am intuitivsten zu bedienen ist.

Der übliche Kunde hält sich heute nicht lange damit auf, die Bedienung einer App

zu erlernen oder seinen eigenen Arbeitsablauf daran anzupassen. Eine App kann schnell gewechselt werden und Probleme in der Bedienung oder Qualitätsmängel werden sofort abgestraft. Wir als Kunden binden uns nur noch selten an konkrete Marken und Produkte. Allgemein wird dieses Umfeld für Unternehmen oft als VUCA (Volatility, Uncertainty, Complexity, Ambiguity) beschrieben. Mit diesem neuen Anspruch an die Produkte von heute ist auch der Anspruch an einen qualitativ hochwertigen Service gestiegen. Wie bei den Produkten ist auch hier der Wechselwille der Kunden sehr hoch, und schlechte Erfahrungen haben oft Konsequenzen für ein Geschäftsverhältnis zwischen Dienstleister und Kunde.

Um dem Kundenverhalten zu entsprechen, haben sich mittlerweile in der Produktentwicklung agile Vorgehensmodelle als Standard durchgesetzt. Sie versprechen konstante und schnelle Lieferung von Funktionalitäten bei hoher Qualität. So wird versucht, immer nah am Kunden zu sein und diesen auf seiner Plattform zu halten. Nun stellt sich die Frage, ob agile Vorgehensmodelle möglicherweise auch Vorteile im Dienstleistungssektor haben könnten. Wir glauben, sie können.

Agile Vorgehensmodelle in Serviceorganisationen

Versucht man, agile Vorgehensmodelle in Serviceorganisationen zu etablieren, stellt sich zunächst einmal die Frage nach dem Produkt. Arbeitet man agil, steht das Produkt

immer im Zentrum der täglichen Arbeit. Das Ziel eines Teams ist es, stets ein hochwertiges Produkt zu entwickeln, das die Ansprüche des Kunden erfüllt und ihn als solchen erhält. Betrachtet man aber nun das klassische Dienstleistungsgeschäft, ist es zunächst schwer, ein konkretes Produkt zu identifizieren. Der Kunde hat in erster Linie ein Anliegen und kommt damit zum Dienstleister. Der Kontakt zwischen dem Dienstleister und dem Kunden kann über verschiedene Schnittstellen, wie zum Beispiel Servicemitarbeiter, eine App oder auch ein Self-Service-Portal, hergestellt werden. Aber was ist in diesem Augenblick das Produkt des Unternehmens? Sind es die Schnittstellen, mit denen der Kunde interagiert? Ist es die Kundenzufriedenheit, welche ein Resultat der Erfüllung eines Kundenanliegens ist?

Unserer Meinung nach sind die Produkte, auf die sich eine Serviceorganisation konzentrieren muss, die Geschäftsprozesse hinter den Schnittstellen. Wenn diese Geschäftsprozesse das Kundenanliegen wie erwartet mit hoher Qualität bedienen, bekommt der Kunde das, was er möchte und es entsteht eine starke Kundenbindung. Jeder Geschäftsprozess selbst besteht dabei aus mehreren Schritten, die durch unterschiedliche Personen, Personengruppen und unterstützende Systeme ermöglicht werden.

Somit wird auch klar, warum das Kundenerlebnis durch die klassischen Optimierungsansätze nach Taylor entwickelten „One Best Way“-Prozesse leidet und nicht mehr in die heutigen Märkte passt. Generell wurde stark nach innen gerichtet auf Effizienz opti-

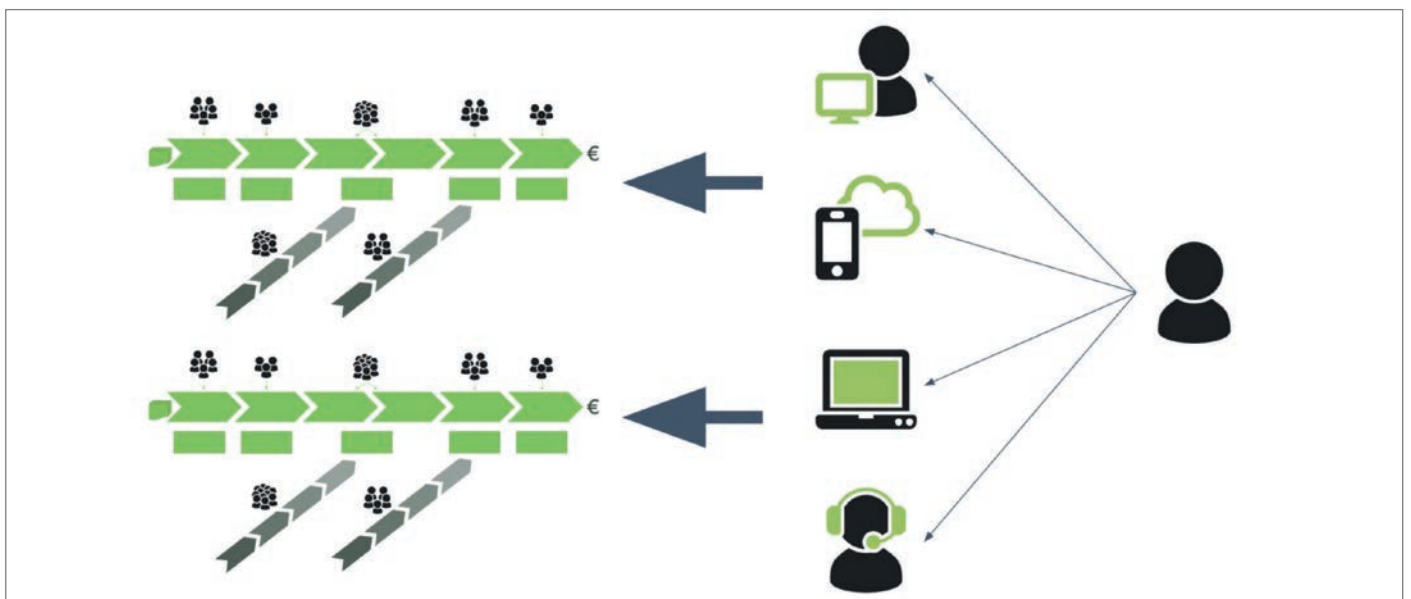


Abbildung 1: Interaktion eines Kunden mit einer Serviceorganisation

miert. Dabei werden häufig die Kunden ignoriert und in erster Linie interne KPIs bedient. Zusätzlich ist ein auf Effizienz ausgerichtetes Gesamtsystem häufig sehr starr und kann nicht so einfach auf von der Norm abweichende Gegebenheiten reagieren. Die resultierenden starren Geschäftsprozesse entsprechen damit nicht dem Verhalten heutiger Kunden und passen nicht in ein VUCA-Umfeld. Aber wie können agile Vorgehensmodelle hier helfen und was muss sich dafür ändern? Hier sehen die Autoren zwei Aspekte.

Einer dieser Aspekte ist die konstante Verbesserung und Automatisierung der unterstützenden Wertschöpfungsketten und Systeme. Das Ziel der Entwicklungen muss es hierbei sein, die Mitarbeiter und die Kunden selbst dabei zu unterstützen, Kundenanliegen möglichst zufriedenstellend zu erfüllen. Je nach Unternehmen und Kundenfokus kann die Ausprägung der gleichen Dienstleistung stark voneinander abweichen. Als Beispiel kann man sich hier Starbucks und eine Bäckerei am Bahnhof vor Augen führen. Beide verkaufen die Dienstleistung der Zubereitung einer Tasse Kaffee an ihre Kunden. Die Ausrichtung ist jedoch völlig unterschiedlich. Das eine Unternehmen bietet ein Erlebnis und einen Ort, an dem ganze Generationen zusammenfinden. Das andere verkauft die schnelle Koffeindosis, während man zwischen Straßenbahn und ICE hin und her hetzt. Je nach Ausrichtung des Unternehmens müssen die Geschäftsprozesse entsprechend optimiert sein und die unterstützenden Systeme deren Durchführung ermöglichen. Dieses Thema ist generell bereits durch agile Softwareentwicklung, sowie systemische agile Ansätze wie SAFe [3] abgedeckt und am Markt weit verbreitet.

Der zweite und für die Autoren interessantere Aspekt ist jedoch, die klassische Denkweise des „One Best Way“ aufzubrechen und in ein Zeitalter der konstanten kundenorientierten Anpassung und Optimierung auf Serviceebene selbst überzugehen. Durch den steigenden Grad der Automatisierung können bei Servicemitarbeitern freie Kapazitäten geschaffen werden. Diese sollten primär in die konstante Weiterentwicklung der Geschäftsprozesse zur Steigerung des allgemeinen Kundenerlebnisses investiert werden. Wir sehen hier eine Parallele zu dem Trend aus der Softwareentwicklung hin zu DevOps. Früher waren Teams oft mit repetitiven Aufgaben im Bereich von Konfiguration, Umgebungs-

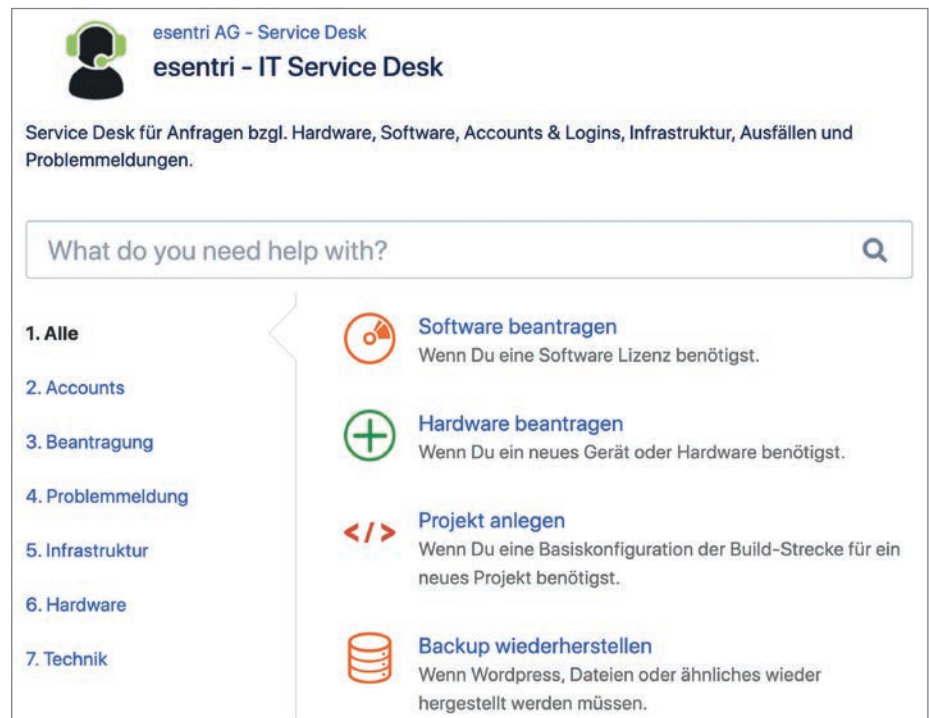


Abbildung 2: esentri IT Service Desk

management oder Lieferung neuer Softwareversionen beschäftigt. Heute ist man sich im Bereich der Softwareentwicklung jedoch einig: Sich wiederholende und fehleranfällige Aufgaben, welche Mitarbeiter oft langweilen, sollten automatisiert werden. Die geschaffenen Freiräume können dann genutzt werden, damit die Entwickler sich auf die Weiterentwicklung des Produktes konzentrieren und Mehrwert für den Kunden erzeugen können.

Ein ähnlicher Wandel ist im Bereich der Serviceorganisationen nötig. Die durch Automatisierung frei werdenden Kapazitäten sollten von Mitarbeitern der Serviceorganisation genutzt werden, um die Geschäftsprozesse als ihr eigenes Produkt kundenorientiert weiterzuentwickeln. Voraussetzung hierfür ist die Etablierung eines Systemdenkens innerhalb der Serviceorganisation. Die Geschäftsprozesse müssen abteilungsübergreifend betrachtet und anhand von durch den Kunden definierten Qualitätsmerkmalen ganzheitlich optimiert werden. Hierfür können bereits existierende agile Methoden und Werkzeuge wie Scrum, Kanban, Retrospektiven und Kreativmethoden genutzt werden.

Erfahrungen mit Agilität in Serviceorganisationen

Die Autoren glauben daran, dass mit der fortschreitenden Automatisierung, der Etab-

lierung einer agilen Denkweise und der Nutzung von agilen Methoden innerhalb von Serviceorganisationen die Entwicklung von der Servicewüste hin zur Service-Oase gelingen kann. Allerdings stehen wir hier noch am Anfang einer längeren Reise, auf der wir viel lernen müssen. So sind existierende agile Prinzipien nicht unbedingt direkt auf Serviceorganisationen anzuwenden. In dieser Hinsicht lernen wir aktuell viel in unserer eigenen Organisation und bei unseren Kunden. Im letzten Abschnitt wollen die Autoren noch einen Einblick in die Maßnahmen geben, welche wir aktuell in unserer eigenen Organisation nutzen.

Innerhalb der esentri AG ist unser Backoffice für die übrigen Mitarbeiter eine interne Serviceorganisation. Als Unternehmen mit einem IT-Fokus haben wir zunächst begonnen, einen Großteil unserer internen Backoffice-Prozesse zu automatisieren. Als neue Schnittstelle haben wir dazu die von Atlassian entwickelte Service Desk Applikation eingeführt. Von Haus aus ermöglicht diese Lösung eine strukturierte Aufnahme von Kundenanliegen und die Definition von spezialisierten Kanban-Workflows für die Mitarbeiter des Backoffice. Ein Vorteil dieses Set-ups ist, dass durch die strukturierte Aufnahme der Anliegen viele Aufgaben ganzheitlich automatisiert werden können. Ergänzend zum Service Desk werden noch weitere Softwarelösungen von Drittanbie-

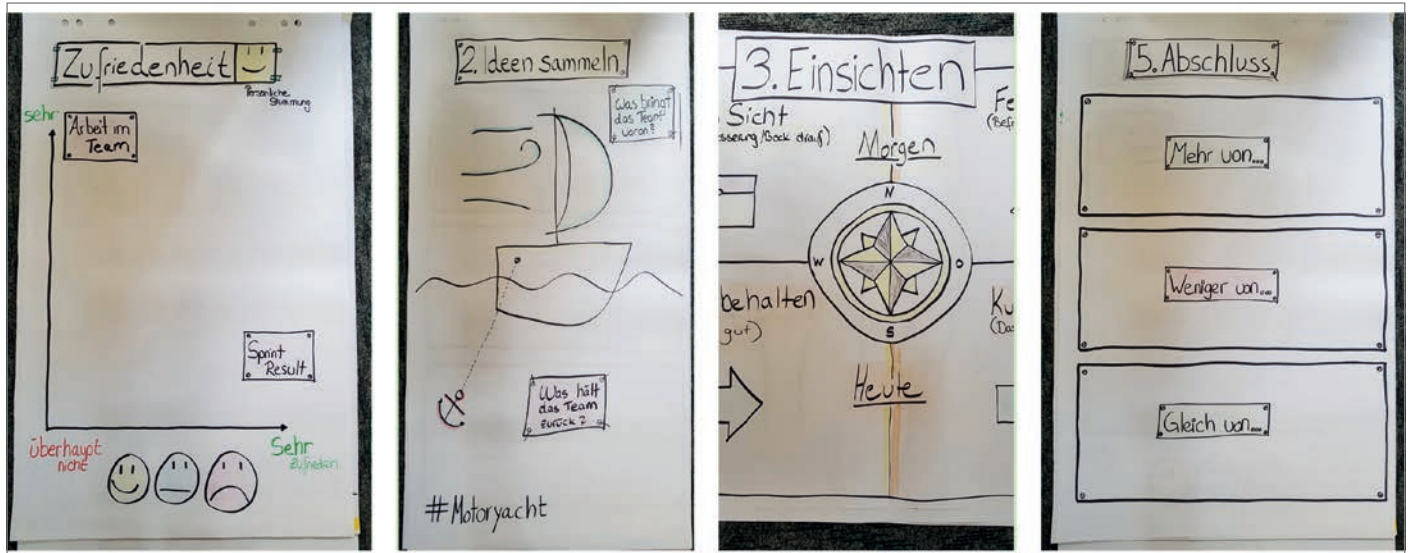


Abbildung 3: Backoffice Retrospektiven

tern genutzt, die den Mitarbeitern Self-Service-Portale bieten. So können zum Beispiel Änderungen der Personalstammdaten direkt von Mitarbeitern eingepflegt werden.

Die durch die Nutzung von Kanban erzielte Produktivitätssteigerung sowie die durch die fortschreitende Automatisierung frei werdenden Kapazitäten nutzen wir wiederum zur kundenorientierten Weiterentwicklung unserer Geschäftsprozesse. Hierfür haben wir ein internes agiles Coaching unserer Mitarbeiter etabliert und führen regelmäßige bereichsübergreifende Retrospektiven durch.

Die Erfahrungen sind hier ähnlich wie bei Teams im Bereich der Softwareentwicklung. Zunächst muss eine Atmosphäre innerhalb des Teams geschaffen werden, welche einen offenen Austausch über Herausforderungen und Potenziale ermöglicht. Dies kann nach unserer Erfahrung häufig mehr Zeit und Aufwand in Anspruch nehmen als innerhalb von IT-Projekten. Oft liegt dies in fehlendem Vorwissen zu agilen Methoden begründet und dem jahrelang antrainierten Fokus auf den eigenen isolierten Bereich. Um hier ein Umdenken zu ermöglichen, muss in entsprechende Trainings und in ein persönliches Coaching investiert werden. Nimmt ein Team erfolgreich die ersten Hürden, entsteht schon bald eine eigene Dynamik und die kundenorientierte Weiterentwicklung der Geschäftsprozesse findet statt.

Die Erfolgsfaktoren agiler Vorgehensmodelle in Serviceorganisationen

Auf Basis der bisherigen Erfahrungen sehen die Autoren drei konkrete Erfolgsfaktoren für

die Einführung von agilen Vorgehensmodellen in Serviceorganisationen. Zunächst einmal muss die Gewohnheit der lokalen Optimierung und Erfüllung von KPIs durch ein ganzheitliches Systemdenken ersetzt werden. Nur wenn eine Serviceorganisation die gesamte Wertschöpfungskette betrachtet und die unterstützenden Prozesse und Systeme synchronisiert weiterentwickelt und automatisiert, werden entsprechende Freiräume für die Organisation geschaffen. Mit den so gewonnenen Freiräumen kann eine Kultur der konstanten kundenorientierten Weiterentwicklung der Geschäftsprozesse durch die Mitarbeiter vorangetrieben werden. Damit dies Erfolg hat, bedarf es jedoch einer offenen und gleichberechtigten Kommunikation über Abteilungen und Bereiche hinweg.

Zusammenfassend ist ein großes Potenzial für Serviceorganisationen und die Servicequalität in Deutschland zu erkennen. Wir befinden uns jedoch am Anfang einer längeren Reise, auf der es noch viel zu entdecken und lernen gilt.

Quellen

- [1] statista (2020): Anteil der Wirtschaftszweige an der Bruttowertschöpfung* in Deutschland im Jahresvergleich 1991 und 2018. Statistisches Bundesamt, <https://de.statista.com/statistik/daten/studie/255082/umfrage/struktur-des-bruttoinlandsprodukts-in-deutschland/>
- [2] Frederick W. Taylor (2006, Nachdruck von 1911): The principles of scientific Management. Cosimo, New York, ISBN 1-59605-889-7
- [3] Scaled Agile, Inc: Scaled Agile Framework for Lean Enterprises, <https://www.scaledagileframework.com/>



Carsten Wiesbaum
carsten.wiesbaum@esentri.com

Carsten Wiesbaum ist schon immer fasziniert von Technologie und Software. Was ihm früh klar wurde: Dass Technologie und Software meist nicht Ursache oder Lösung für Probleme in Unternehmen sind und dass Menschen in ihrem Arbeitsalltag Erfüllung und Selbstverwirklichung finden können. Hierzu braucht es jedoch ein neues Denken in Unternehmen, welches auf agilem Mindset, Empowerment und Transparenz basiert. In dieser Entwicklung unterstützt er die esentri AG und andere Unternehmen seit vielen Jahren.



Ingo Sobik
ingo.sobik@esentri.com

Ingo Sobik unterstützt als Consultant und Agile Coach die Kunden der esentri AG bei ihrem digitalen Wandel. Dabei versteht er es als Digital Native, dass nachhaltige Weiterentwicklung im Unternehmen auch immer eine kulturelle Weiterentwicklung bedeutet. Er kombiniert fundiertes technisches Know-how mit einer ausgeprägten methodischen und sozialen Kompetenz. Dies qualifiziert ihn als Berater, Coach, Moderator oder Trainer für die Arbeit an den Schnittstellen zwischen Technik, Fach und Management. Als Vorstand der DOAG NextGen-Community engagiert er sich aktiv für die Förderung der IT-Talente von morgen.

Erfolgreiche Kommunikation in agilen Projekten ist eine Frage der Beziehung

Carsten Firus, MT AG

Wenn in Unternehmen agiles Projektvorgehen eingeführt wird, stellt das eine Herausforderung für viele Beteiligte dar. Nicht nur das Prinzip der Selbst- bzw. Teamverantwortung wird neu entdeckt. Es gilt eine gänzlich neue Denke zu etablieren, die sich weg von den gewohnten Wasserfall-Mustern bewegt. Dass dabei Irritationen auftreten ist voraussehbar und nachvollziehbar. Diesen kulturellen Wandel im Unternehmen und der neu zu justierende Mindset der Beteiligten gilt es entsprechend zu moderieren – und zu fördern. Dabei darf nie vergessen werden: Der Kundennutzen steht im Fokus der Aktionen. Der Autor zeigt in seinem Beitrag, worauf es ankommt.

„Agiles Vorgehen sichert Ihre Zukunftsfähigkeit.“ So oder ähnlich formulieren Vertreter der agilen Welt. Zugegeben, da ist allerhand dran. Denn agiles Vorgehen berücksichtigt manch einen Mechanismus, den Fredmund Malik bereits in „Strategie des Managements komplexer Systeme“ herausgearbeitet hat. Als Beispiele seien kurze Feedback-Zyklen, schrittweises Vorgehen und klare Kommunikation genannt.

Gleichzeitig ist festzuhalten, dass agiles Vorgehen kein Allheilmittel ist. Aufgrund ihrer biografischen Prägung fällt es verschiedenen Menschen durchaus schwer, sich auf ein agiles Vorgehen einzulassen. Zu sehr sind sie davon geprägt, dass es jemanden gibt, der ansagt, was wie zu tun ist und jemand anderes für die Ausführung zuständig ist. Selbstverantwortliches Arbeiten liegt nicht allen. In Organisationen und Unternehmen ist durch das Aufeinandertreffen solcher Menschen eine dementsprechende Kultur entstanden. Wer in einem solchen Umfeld allen ein agiles Vorgehen überstülpen möchte, provoziert Widerstand. Zwischen den Zeilen wird die Botschaft „Ihr seid nicht in Ordnung“ wahrgenommen – und das mag niemand gerne hören. Bevorzugt geschieht das, wo der Übergang in die agile Welt als ein klassisch geführtes und (im besten Sinne des Wortes) kontrolliertes Projekt geschehen soll. Die Wahrscheinlichkeit steigt, dass dieses Vorhaben in die Liste der gescheiterten Veränderungs-Projekte aufgenommen wird.

Genug der kritischen und warnenden Worte. Was ist in der Kommunikation zu beachten, um die Chance zu erhöhen, erfolgreich in der agilen Welt anzukommen und

von den Vorteilen eines agilen Vorgehens zu profitieren?

Wir befinden uns in einem komplexen Umfeld. Zu Beginn eines agilen Projektes sind die zu erfüllenden Anforderungen weitgehend unbekannt, im weiteren Verlauf ändern sie sich häufig und auch hinsichtlich technischer Herausforderungen herrscht ein gewisses Maß an Unsicherheit.

Agile Projekte greifen diese Unsicherheiten auf, indem mit zeitlich befristeten „Sprints“ von beispielsweise vier Wochen Dauer iterativ vorgegangen wird. Am Ende eines Sprints werden die Ergebnisse den Auftraggebern vorgestellt und unmittelbares Feedback eingeholt. Der Auftraggeber

soll von einem möglichst hohen Geschäftswert profitieren. Es geht darum, die 20% an Features zu entwickeln, die regelmäßig genutzt werden, und es geht darum zu vermeiden, Funktionalität bereitzustellen, die quasi nie verwendet wird. Darüber hinaus wird eine ständige Verbesserung des Produktes und der Kooperation angestrebt. Entsprechend des agilen Manifests wird mehr Wert auf direkte Kommunikation als auf das Erstellen von Dokumenten gelegt. Darin zeigt sich eine Haltung zueinander auf Augenhöhe und von Vertrauen geprägt.

In der Literatur wird immer wieder auf die Menschenbilder gemäß Theorie X/Y verwiesen, eine Unterscheidung, die auf Douglas Mc-

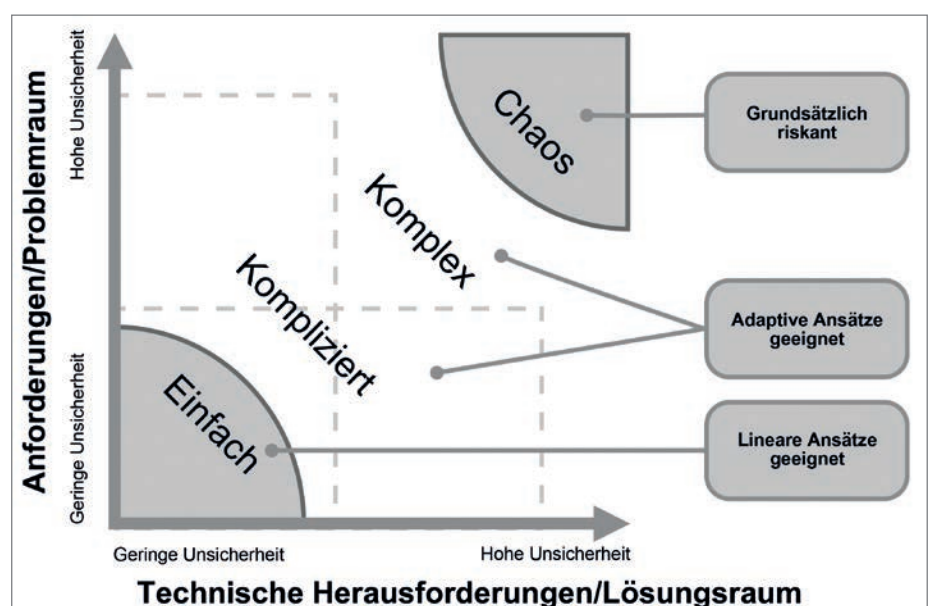


Abbildung 1: Unsicherheits- und Komplexitätsmodell nach Stacey – Quelle: Carsten Firus nach PMI und Agile Alliance S. 14

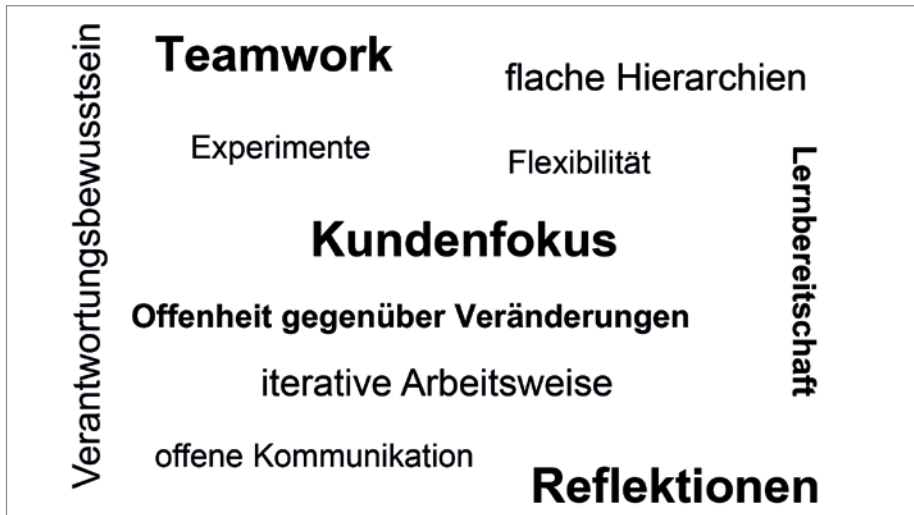


Abbildung 2: „Was macht eine agile Haltung aus?“ – Antworten während eines Vortrags im Rahmen der DOAG Konferenz und Ausstellung 2019 – Quelle: Carsten Firus

Gregor zurückgeht. Nach Theorie X sind Menschen extrinsisch motiviert und packen nichts von sich aus an. Erst wenn sie ausreichend starke Anreize (Belohnung oder Druck) von außen erhalten haben, legen sie los. Anders die Theorie Y. Demnach sind Menschen von sich aus motiviert, das heißt intrinsisch, sie wollen etwas erreichen und sich entwickeln. Wenn wir heute kleine Kinder beobachten, fällt auf, dass wir als Y geboren werden: Neugierig machen sie sich auf den Weg, entdecken die Welt, probieren Neues aus, fallen hin, stehen auf, hören zu, beginnen zu reden, nehmen hemmungslos Kontakt zu anderen auf. Doch die folgenden Prägungen mit Regeln, Vorschriften und Anreizen sorgen oftmals dafür, dass sich Menschen als X weiterentwickeln.

Das wirkt sich darauf aus, wie teamintern und mit Kunden über Ziele, Bedürfnisse, Änderungen, Herausforderungen und technische Lösungen gesprochen wird. Allgemein gilt, dass vertrauensvoll miteinander geredet wird, sodass offen Fehler thematisiert werden können und Lösungen zu deren zukünftigen Vermeidung gesucht werden (anstatt sich in wechselseitigen Schuldzuweisungen zu verstricken). Dazu gehört ein gutes Maß an Mut und Empathie, die Bereitschaft zuerst zu verstehen, bevor man verstanden wird (vgl. das gleichnamige Prinzip von Stephen Covey in „Die 7 Wege zur Effektivität“). Um Prozesse lösungsorientiert nach vorne zu bringen, werden Fragen gestellt, die

- ernst gemeint sind und nicht als versteckte Aufforderungen getarnt sind.

Beispielhaft seien die folgenden, durchaus bekannten Fragen genannt:

- Wie?
- Was?
- Wie aufwendig?
- Wie viel?
- Wer braucht welche Hilfe?
- Was leitet dich?
- Woran machst du fest, dass...?
- Was brauchst du, um ...?
- Aus welchem Grund?

Wahrnehmung ist subjektiv. Jeder nimmt die eine Wirklichkeit aus seiner ganz speziellen Perspektive und mit seiner einzigartigen

- den anderen (nicht nur in seiner Wahrnehmung und Professionalität) ernst nehmen,
- auf Lernen und Erkenntnisgewinn aus sind und die

biografischen Prägung wahr. Damit werden Informationen gefiltert, nicht alles dringt ins Bewusstsein. So ergeben sich die unterschiedlichen Aussagen über eine sachliche Auseinandersetzung. Aus der subjektiven Wahrnehmung heraus gestaltet jeder seine Wirklichkeit und deutet das Erlebte in seinem Sinn. Menschen machen sich die Welt, wie sie ihnen gefällt. Deshalb sind offene Fragen so wichtig. Sie rechnen mit der berechtigten unterschiedlichen Wahrnehmung und Deutung. Über Gemeinsamkeiten und Unterschiede sprechen zu lernen gehört zu den wichtigen Grundlagen.

Das spiegelt sich auch im Feedback wider. Feedback wird gegeben, indem beschrieben wird, was wahrgenommen wurde, wie sich die Wahrnehmung in einem ausgewirkt hat und welchen Wunsch man für die Zukunft hat. Das ist subjektiv und überlässt dem Gegenüber die Freiheit, sich zum Wunsch zu positionieren. Bewertungen werden auf ein Minimum reduziert. Zentral bleibt die Ausrichtung auf den Geschäftswert für den Kunden, das gemeinsame Ziel.

Mit etwa sieben Jahren ist die Prägung der Art und Weise, wie Menschen kommunizieren abgeschlossen – außer man hat begonnen, bewusst das eigene Verhalten und die Kommunikation zu reflektieren und aktiv zu gestalten. Um es plastisch zu formulieren: Muster, die im Sandkasten angewendet wurden, finden heute Einzug ins Projektgeschäft. Nicht alle sind förderlich. Sie sind jedoch biografisch nachvollziehbar. Deshalb sind die Retrospektiven, die regelmäßige, bewusste Reflektion der Zusammenarbeit, ein zentrales Element im agilen Alltag. Mit ihrer Hilfe können sich Beteiligte entwickeln, können hinderliche

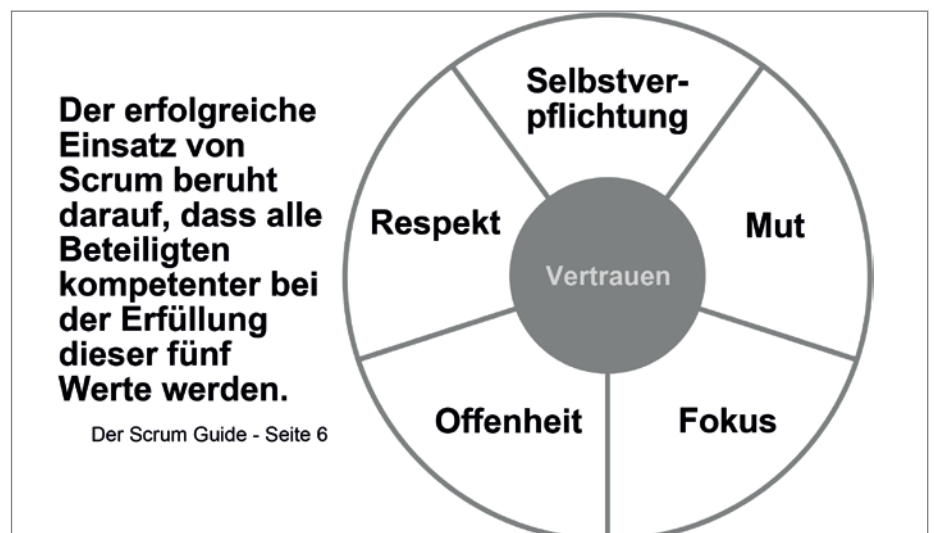


Abbildung 3: Fünf Werte in Scrum – Quelle: Carsten Firus in Aufnahme des Scrum-Guides

Muster erkannt und durch förderliche (schrittweise) ersetzt werden. Allen Beteiligten sollte bewusst sein, dass solche Reifeprozesse nicht mit dem Umlegen eines Lichtschalters zu vergleichen sind. Es sind Veränderungen, die mit mühsamer Arbeit und einigen Rückschlägen verbunden sind. Nur weil wir etwas vom Kopf her verstanden haben, gelingt es uns noch lange nicht, es fortan in jeder Situation entsprechend abzurufen und zu leben. Manch einer wird dabei sein eigener ärgster Feind sein. Sich selbst und anderen gegenüber barmherzig zu sein und Prozesse (!) der Entwicklung zuzugestehen ebnet den Weg.

Nicht alles, was in einer Retrospektive zutage tritt ist im Rahmen des Teams weiter zu bearbeiten. Es werden immer wieder biographische Prägungen einzelner deutlich werden, die in Einzelgesprächen aufgearbeitet werden sollten. Dort ist der Platz, um beispielsweise „schrägern“ Verhalten Vorgesetzten gegenüber auf den Grund zu gehen. Manchmal hilft es, Ursachen in der Vergangenheit zu entdecken. Wichtig ist jedoch, dass meist reaktive und unbewusste Verhalten durch ein neues, der aktuellen Situation gerechtes Verhalten zu ersetzen. Die eigene Haltung, das Denken und das Verhalten zu reflektieren hilft, ausgehend von den eigenen und fremden Bedürfnissen her, neue adäquate Verhaltensweisen zu entwickeln.

Agile Teams sind oftmals crossfunktional zusammengesetzt. Darin kann nicht jeder alles (auf Experten-Niveau), sondern jeder hat ein oder mehrere Spezialgebiete, während er in den anderen Gebieten genug Fachwissen und Kompetenzen mitbringt, um zu unterstützen. Es kann auch sein, dass alle zunächst als Spezialisten beginnen und sich im Lauf der Zeit durch den engen Austausch mit den Teamkollegen in die Breite entwickeln. Diese funktionsübergreifende Zusammenarbeit gelingt vor allem dann, wenn allen Teammitgliedern klar ist, dass es zur Zielerreichung auf einen selber und auf die anderen ankommt. Es braucht die unterschiedlichen Perspektiven, damit das gesamte Projekt zum Erfolg werden kann. Wo stattdessen einer stets das letzte Wort haben möchte oder an den Ideen der anderen immer etwas auszusetzen hat, wird unweigerlich Sand ins Getriebe kommen. Es kann zu Konflikten kommen, die den Projektfortschritt gefährden. Deshalb ist es hilfreich, wenn eine Retrospektive von jemandem geleitet wird, der außerhalb des Projekt-Teams und der sonst bestehenden Linienhierarchie steht. Sollte keine geschulte Person inner-

halb des Unternehmens zur Verfügung stehen, kann auch auf Supervisoren zurückgegriffen werden, die in der Begleitung von Teams erfahren sind [1].

Eine agile Haltung macht in der Kommunikation und bei der Anwendung von bekannten Methoden den Unterschied. Im Brainstorming ist jede Idee erlaubt. Es wird mehr über Lösungen gesprochen als über Probleme. Jeder darf zu Wort kommen. Freiheit und Verantwortung gehören zusammen. Das Team steht zusammen dafür ein, dass das Ziel erreicht wird. So neu ist also das, was als „agiles Mindset“ beschrieben wird gar nicht. In der Management-Literatur finden sich solche Aspekte schon seit Jahrzehnten. Innerhalb der agilen Welt setzt man sich allerdings sehr bewusst für solch eine Haltung und diese Verhaltensweisen ein bzw. sind sie vorausgesetzt.

Das Verhältnis zu Kunden verändert sich radikal. Sie sind Geschäftspartner. Die Betonung liegt auf „Partner“. Es geht darum, für den Kunden wertvollen Nutzen zu stiften. Auf die Leistung folgt die Rechnungsstellung und -begleichung, so dass auch an die Lebensfähigkeit des leistenden Unternehmens gedacht ist. Als eine der größten Umstellungen für Auftraggeber hat sich die ständige Verfügbarkeit für das entwickelnde Team herausgestellt. Wenn die Anforderungen nicht zu Beginn feststehen, werden sie im Laufe des Projekts auf die nötige Detailtiefe gebracht. Das bedarf der Abstimmungen, der Kommunikation. Dafür entlastet es davon, im Vorfeld treffgenaue Formulierungen für das Lastenheft finden zu müssen. Die Zeit wird nicht mehr vor Arbeitsaufnahme des Dienstleisters in die Erstellung von Dokumenten, sondern während der Erstellungsphase in die direkte Kommunikation mit dem Auftragnehmer investiert. Weiter sind für die sogenannten „Reviews“ am Ende eines Sprints regelmäßig Vertreter des Auftraggebers abzustellen, die ihr Augenmerk darauf richten, ob die richtige Software und diese richtig erstellt wird. Denn der Auftraggeber ist selbst in der Pflicht dafür zu sorgen, dass die Software den nötigen Nutzen stiftet. Das schrittweise Vorgehen ermöglicht gerade den klaren Fokus auf die wichtigsten Funktionalitäten, die den späteren Anwendern einen Mehrwert liefern.

Hat sich die Zusammenarbeit zwischen Auftraggeber und Dienstleister etabliert, ist das Vertrauen gewachsen, so lassen sich auch die Verträge derart gestalten, dass dem agilen, schrittweisen Vorgehen Rechnung getragen wird. In *„Der agile Festpreis. Leitfaden für wirklich erfolgreiche IT-Projekt-Verträge“* wird Walter Ja-



Der grüne Faden für Ihre Digitale Evolution

Wir bei PROMATIS folgen einem selbst entwickelten grünen Faden:

Mit professioneller Beratung und innovativen Digitalisierungslösungen schaffen wir exzellente Geschäftsprozesse: agil, bedarfsgerecht, intelligent und zukunftssicher. Nachhaltige Qualität und Wirtschaftlichkeit sichern wir durch kontinuierliche Verbesserung der eingesetzten Verfahren, Produkte und Services.

Mit unserer Digitalisierungskompetenz und unseren Best Practice-Lösungen begleiten wir Sie auf Ihrer Reise in die Oracle Cloud.

PROMATIS Gruppe
Pforzheimer Str. 160
76275 Ettlingen
+49 7243 2179-0
www.promatis.de

Ettlingen | Hamburg | Berlin | Münster
Wien | Zürich | Denver

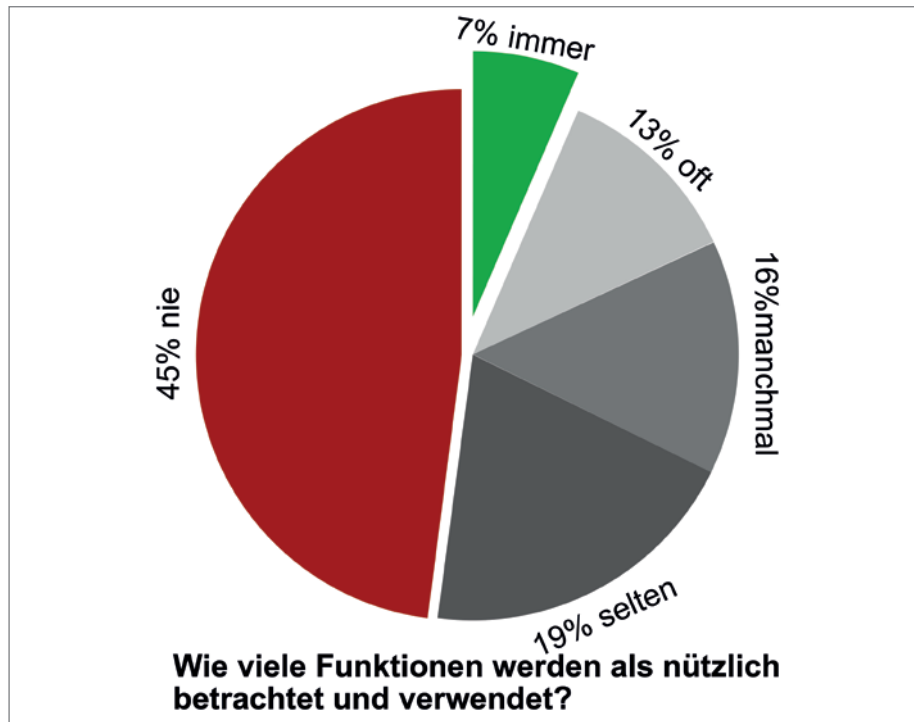


Abbildung 4: Verwendung vorhandener Funktionalität – Quelle: Carsten Firus nach Ebert S. 8

burek als ein „beeideter und gerichtlich zertifizierter Sachverständiger für Informationstechnik und Telekommunikation“ vorgestellt (S. 5) und später zitiert: „Time & Material ist die beste Vertragsform, wenn auf beiden Seiten absolute Ehrlichkeit herrscht.“ (S. 234). Mit anderen Worten: Können Auftraggeber und -nehmer einander vertrauen, braucht keiner aus Angst vor Übervorteilung einen Festpreisvertrag mit konkreten Angaben zu den vereinbarten Umständen abschließen. Es reicht aus, sich auf Prozessschritte für das Vorgehen Scrum (oder ein anderes agiles Vorgehen) zu einigen.

Auch was Führung angeht verändert sich in der Praxis einiges, in der Theorie hingegen weniger. Denn Stichworte wie „Servant Leadership“ oder „Die Führungskraft als Coach“ existieren bereits seit Jahren. Doch im agilen Umfeld werden sie mit Leben gefüllt. Eine hierarchisch geprägte Führungskraft ist herausgefordert, Selbstorganisation des Teams zuzulassen. „Jede Gruppe ist eine Fundgrube voll Kreativität und Talent... Sie müssen ein Umfeld schaffen, in dem die Team-Mitglieder bereit sind, etwas zu riskieren, zu wachsen, Verantwortung zu übernehmen und kreativ zu sein... Bevollmächtigung heißt also, Gruppen bei der Entwicklung ihrer Fertigkeiten und Kenntnisse zu helfen und sie beim praktischen Gebrauch ihrer Begabungen zu unterstützen.“ Unerfahrene Teams werden punktuell versuchen, Verantwortung wieder an die Führungskraft zurück zu delegieren.

Dann hat der Vorgesetzte der Versuchung zu widerstehen und Entscheidungen (soweit möglich) zurückzugeben. Hier helfen Fragen, die das Team darin unterstützen, die eigene Freiheit und Verantwortung zu übernehmen. Beispiele: Worum geht es genau? Was habt ihr schon probiert? Wie weit seid ihr gekommen? Wo genau kommt ihr an eure Grenzen? Was hindert euch? Aus welchem Grund kommt ihr nicht selbst zu einer Entscheidung? Bei Bedarf kann Hilfestellung gegeben werden, indem Entscheidungsprozesse moderiert werden, ohne die Entscheidung zu beeinflussen. Mit

einem solchen Vorgehen demonstriert die Führungskraft ihr Vertrauen ins Team und gewinnt dadurch Freiheit, sich um die eigenen strategischen Themen zu kümmern. Im wirklichen Leben wird dieser Prozess von manchen Führungskräften als ein herber Kontrollverlust erlebt, weil weniger direkter Einfluss geltend gemacht wird. Unterstützung erfährt die Führungskraft, wenn unternehmensweit geregelt ist, wer was bis wohin entscheiden darf.

Fazit

Scrum und andere agile Vorgehen sind für komplexe Verhältnisse entwickelt worden. Was hinter dem agilen Manifest und den zugehörigen Prinzipien steckt, sind Prozesse, die auf Vertrauen aufbauen und Vertrauen stärken. Niklas Luhmann schreibt: „Vertrauen braucht man zur Reduktion einer Zukunft von mehr oder weniger unbestimmt bleibender Komplexität.“ Dabei stellt er Vertrauen nicht als DIE Lösung der Herausforderung Komplexität dar. Auch das Vertrauen beinhaltet ein Risiko, jedoch eines, mit dem es sich oftmals leichter leben und arbeiten lässt. Komplexität ist ein Risiko, das nicht aufgelöst werden kann.

„Vertrauen ändert alles... In einer komplexen Welt könnte alles, dem wir nicht vertrauen, eine Bedrohung darstellen. Uns interessieren die wahren Intentionen der anderen und je transparenter wir sie wahrnehmen, desto glaub- und vertrauenswürdiger können wir sie einschätzen. Wenn wir vertrauen, muss unser mentaler Apparat viel weniger auf der Hut sein, viel weniger Risiken abschätzen; das entspannt und er kann sich produktiveren Aufgaben zuwenden.“



Abbildung 5 - Quelle: Carsten Firus in Aufnahme von Pfläging und Hermann S. 9

Deshalb kommt es darauf an, im täglichen Miteinander die eigene Grundhaltung zu sich selbst, den anderen Beteiligten und zum Ziel zu reflektieren – sich selbst und anderen auf Augenhöhe zu begegnen, getreu dem Motto: „Ich bin okay – und du bist es auch.“ Das zeigt sich in der Wortwahl und im Tonfall. Es wirkt sich auf das Team aus, verändert das Führungsverständnis und die Kundenbeziehung. Mit einfachen Rezepten der Kommunikation ist es nicht getan. Es ist mitunter harte (biografische) Arbeit, dorthin zu gelangen. Doch die Mühe lohnt sich: Am Ende stehen Produkte, die genau das können, was gebraucht wird. Die Kosten wurden minimiert, weil viel weniger von dem entwickelt wurde, was niemand braucht und was noch wichtiger ist: Man hat Partner gewonnen, denen man vertraut. Es ist deutlich geworden, worauf man sich verlassen kann, wenn man zusammenarbeitet. Erfolgreiche Kommunikation ist eine Frage der Beziehung.

Quellen

- Fredmund Malik: Strategie des Managements komplexer Systeme. Ein Beitrag zur Management-Kybernetik evolutionärer Systeme, Bern 10. Aufl. 2008
- PMI & Agile Alliance: Praxisleitfaden Agilität, Newton Square, Pennsylvania 2017
- Agiles Manifest und 12 Prinzipien agiler Softwareentwicklung, www.agilemanifesto.org, 2001
- Jeff Sutherland und Ken Schwaber: Der Scrum Guide. Der gültige Leitfaden für Scrum: Die Spielregeln, 2017
- Stephen Covey: Die 7 Wege zur Effektivität. Prinzipien für persönlichen und beruflichen Erfolg, Offenbach 11. Aufl. 2008
- Andreas Opelt, Boris Gloger, Wolfgang Pfarl und Ralf Mittermayr: Der agile Festpreis. Leitfaden für wirklich erfolgreiche IT-Projekt-Verträge, München 3. Aufl. 2018
- Niklas Luhmann: Vertrauen, Stuttgart 4. Aufl. 2009 (Nachdruck)
- Ulf Brandes, Pascal Gemmer, Holger Koschek und Lydia Schültken: Management Y. Agile, Scrum, Design Thinking & Co.: So gelingt der Wandel zur attraktiven und zukunftsfähigen Organisation, Frankfurt 2014
- Kenneth Blanchard, Donald Carew, Eunice Parisi-Carew: Der Minuten Manager schult Hochleistungsteams, Hamburg 2002
- Christof Ebert, Systematisches Requirements Engineering. Anforderungen ermitteln, dokumentieren, analysieren und verwalten, Heidelberg 6. Aufl. 2019
- Niels Pfläging und Silke Hermann: Komplexithoden. Clevere Wege zur (Wieder)Belebung von Unternehmen und Arbeit in Komplexität, München 2015

- [1] Dann sollte darauf geachtet werden, dass Supervisoren einem Berufsverband wie beispielsweise der Deutschen Gesellschaft für Supervision oder der

European Association for Supervision and Coaching angehören.

- [2] Kenneth Blanchard u.a. Der Minuten-Manager schult Hochleistungsteams, S. 101
 [3] Luhmann: Vertrauen, S. 19
 [4] Brandes u.a. Management Y, S. 126



Carsten Firus
 Carsten.firus@mt-ag.com

Carsten Firus arbeitet als Berater bei der MT AG und ist Supervisor mit Schwerpunkt Organisationsentwicklung, Theologie und angehegender Wirtschaftsinformatiker. Er bringt über 15 Jahre Erfahrung der Führung und Veränderung in unterschiedlichen Rollen und Organisationen mit. Begegnungen auf Augenhöhe und miteinander gemeinsame Wege zu gehen sind zwei seiner besonderen Anliegen. Er führt regelmäßige Kommunikationstrainings unter besonderer Berücksichtigung einer agilen Haltung durch.

DOAG 2020 Datenbank

25. und 26. Mai 2020 in Düsseldorf





Führung im Kontext der agilen Arbeit

Daniel Reinold, Visual Braindump

Wie beeinflusst das Thema Führung den Kontext der agilen Arbeit? Ist Führung in diesem Kontext überhaupt notwendig und welche Voraussetzungen empfehlen sich in Verbindung mit unserem unternehmerischen Handeln? Lesen Sie, warum diese Fragen oft im Raum stehen, warum die Antwort darauf oft schwer fällt und wie wir mit dem richtigen Verständnis im Kontext der agilen Arbeit führen (lassen) können. Dieser Artikel möchte zur persönlichen Schlussfolgerung anregen und mittels plakativer Beispiele die eigene Haltung hinterfragen. Gleichzeitig will er Mut machen, die Diskussion über das Thema ergebnisorientiert zu führen.

Die hartnäckige These

„Führung ist in der agilen Arbeit überflüssig.“ Auf diese oder ähnliche Aussagen treffe ich immer wieder, insbesondere in methodisch gesehen jungen Teams, die voller Elan am Beginn der Umsetzung ihres Auftrages stehen – oder von Teams, die sich vom Unternehmen allein gelassen fühlen. Dieses Gefühl ist meiner Erfahrung nach der subjektive Eindruck, der mangels Kommunikation und produktiven Austausches zwischen einem Team und dem restlichen Unternehmen entstehen kann. Leider begegne ich ebenfalls in dieser Verbindung immer wieder Führungskräfte, die dadurch

selbst ins Zweifeln geraten und schlussendlich die Phrase „Führung ist in der agilen Arbeit überflüssig“ ebenfalls wie die sprichwörtliche „Sau durchs Dorf“ treiben.

Daneben gibt es die weniger sichtbare, aber größere Gruppe derjenigen, die ihre funktionierende Art der Führung auf die neuen Gegebenheiten agiler Arbeit anpassen möchten. Dieser Artikel soll einen Weg mit Beispielen aufzeigen, um die Themen Führung und agile Arbeit zu verbinden.

Da ich beide benannten Gruppen berücksichtigen möchte, fange ich mit dem vermeintlich negativen Szenario an, bevor ich auf die allgemeine Situation eingehe.

Die grundsätzliche Frage der Führung im Kontext agiler Arbeit

Die Führungsfrage im Kontext agiler Arbeit begleitet mich in meiner beruflichen Tätigkeit seit nunmehr neun Jahren. Unter den Kunden waren die Sun Microsystems, für die ich als Tier3-Techniker arbeitete, große Banken und Versicherer, wie die Allianz AG, für die ich nach wie vor als Trainer tätig bin, sowie diverse Unternehmen aus IT, Forschung und Entwicklung, darunter Giesecke und Devrient in München, bei dem ich diverse agile IT-Projekte leiten durfte. In der Medienwelt, der ich noch heute punktuell beratend zur Seite stehe, nahm ich eine vornehmlich coachende Rolle ein.

Nach meinem letzten Umzug nach Nürnberg hatte ich Gelegenheit, die Badges einschlägiger Veranstaltungen der letzten vier Jahre zu zählen, bei denen ich direkt oder als Gast involviert war: 28 Veranstaltungen, die zwar nicht alle ein purer Erfolg, aber durchaus erkenntnisreich waren. Während dieser Zeit entstand auch der starke Bezug zum Thema Führung. Gerade zu Beginn der sichtbaren agilen Szene wurde das Führungsthema mit gerührten Blicken und verständnisvollem Kopfnicken begleitet.

Ich möchte Ihnen anhand von Beispielen aus meiner Erfahrung zeigen, wie es zur negativen Haltung gegenüber Führung kommt:

Beispiel 1: Teams starten als agile Versuchsblase und man lässt sie erst einmal machen. Schnell fühlt sich das Team durch diese Abkapselung isoliert, auch nach ersten positiven Stimmungen und Befürwortung. Dies gilt insbesondere dann, wenn organisatorische und technische Abhängigkeiten bei der Lösungserarbeitung auftreten, bei denen das Team Unterstützung benötigt. In der Folge entstehen „kleine gallische Dörfer“, denen man nicht zu nahe kommen sollte. Die Wiedereingliederung ist mitunter aufwendig, begleitet von einem erhöhten Konfliktpotenzial. Das gilt im Übrigen für beide Seiten. Dies ist ein Grund für die Führungsdiskussion und sicherlich ein Problem, das mittels Führung behandelbar ist.

Beispiel 2: Den Einsatz diverser Teams in Verbindung mit der agilen Arbeit erlebte ich als die Verteilung auf vielen Inseln. Stellen Sie sich ein kleines Archipel in der frühen Moderne vor. Der Austausch untereinander, Wissenstransfer, die Identifikation identischer Themenfelder – all diese Punkte werden zu oft stiefmütterlich behandelt, leider zu oft verschleppt und erst erheblich später mit dem verbundenen Aufwand bearbeitet. Gerade die Emergenz ist ein wesentlicher Beitrag der skalierenden agilen Arbeit. Auch wenn die Inseln noch so schön anzuschauen sind, ist es wichtig, schnellstmöglich funktionierende Verbindungsrouten zur Kommunikation untereinander zu etablieren. Wo liegt hier die Verantwortung? Wer wurde befähigt? Welche Mittel wurden genutzt? Wird dem Thema genug Bewusstsein entgegen gebracht? Führung kann hier durch Rat und Tat zur Seite stehen. Nicht immer geht es um die operative Umsetzung. Eine Empfehlung verbunden mit einer Schirmherrschaft oder einfacher Unterstützung zum Aufbau des Netzwerkes kann hier ein wertvoller Schritt voran sein.

Beispiel 3: Dann gibt es noch die Budget-Variante in der Liste der Beispiele. Schnell entstehen Teams in Größen von weit über zehn Spezialisten, die ihre betreuenden agilen Rollen an den Rand der Verzweigung führen. Sei es, dass alle mit relevanten Tätigkeiten versorgt sein wollen oder sie Unterstützung und Führung fordern, die durch eine Person nur mangelhaft abgedeckt werden kann. Die agilen Rollen werden ihren Aufgaben nicht gerecht oder es entsteht ein Flickenteppich. Irgendwann wird punktuell unterstützt, auch in vermeintlichen Kompetenzfeldern der bestehenden Rollen. Natürlich wird auch solch ein Beispiel mit Rufen begleitet. Es wäre schon immer klar gewesen, dass das mit der agilen Arbeit nicht funktionieren könne. Gerade Führungskräfte mit Budget-Verantwortung sollten sich bewusst werden, welche Auswirkung eine mangelhafte Unterstützung nach sich ziehen kann.

Die Reihe ist nahezu beliebig lang erweiterbar. Wenn wir dabei das Thema agiles Arbeiten in den Beispielen herausdenken, handelt es sich immer um ein Alltagsproblem der Führung – ob agil, projektorientiert oder in der Linienarbeit. Die Benennung einzelner Ausprägungen macht es leichter, mit erhobenem Finger darauf zu zeigen und Lager zu bilden. Leider hilft dieses Vorgehen niemandem weiter.

Im agilen Arbeiten ist Führung wichtig und richtig, gerade in der Phase einer Transformation, in der Unternehmen Agilität erstmalig erleben, oder um Personal an agiles Arbeiten heranzuführen und Fehler im angepassten Vorgehensmodell zu beheben. Führung ist auch nötig, um die nächste Qualitätsebene der agilen Arbeit erreichen zu können oder um in skalierende agile Modelle zu wechseln. Sie merken: also eigentlich immer.

Die Herausforderung: Führung ist weder ein Management von Ressourcen noch die Vorgabe und Kontrolle kleinteiliger Ziele. Zudem wird mit der agilen Arbeit eine starke Taktung eingeführt, in der wertvolle Umsetzungsziele erreicht, getestet, mit Feedback versehen, angepasst und eingesetzt werden.

Sinnbildlich: Stellen Sie sich eine Galeere mit vielen Rudern vor, die von einer fähigen Mannschaft bedient werden. Die Taktung der Ruderschläge wird über eine Trommel vorgegeben. Die Trommel folgt einem Rhythmus, um einen optimalen

Ablauf herbeizuführen. Nun erscheint eine Person mit einer zweiten Trommel, die nach Belieben einen zusätzlichen, vielleicht sogar unregelmäßigen Takt einführt. Das Chaos der Ruderschläge können Sie sich sicher vorstellen.

Gleichzeitig ist Fingerspitzengefühl im Bereich des Handlungsspielraumes gefragt. Die Aufstellung breit gesteckter Visionen in Verbindung mit der Vergabe eines breiten Handlungsspielraumes kann zur Herausforderung werden, sofern das Personal diese Art der freien Arbeit nicht kennt bzw. damit nicht umzugehen weiß. Zudem ist Führung keine ausschließliche Einzelkämpfer-Aufgabe mehr. Je nach Situation ist es nötig, die Führung (temporär) in die Hände anderer Mitspieler*innen zu geben, die eine technische oder fachliche Führung situativ für sich beanspruchen. Dann gilt es, die nötige Befähigung zu kommunizieren, Vertrauen zu haben, integer zu handeln und gegebenenfalls zu unterstützen. Wir erklären zum Beispiel auch unserem Zahnarzt nicht, wie er was zu tun hätte. Wir vertrauen diesem Spezialisten und messen am Ergebnis. Die technische und fachliche Führung für diesen Bereich ist nicht in unserer Hand. Es handelt sich um keine dauerhafte Führungsaufgabe, somit ist die Führungskraft nicht aus dem Spiel. Sie wechselt in eine coachende Rolle, die mit ihren weiteren Fertigkeiten eine wertvolle Unterstützung bildet. Im Beispiel hören Sie sich Handlungsoptionen an, hinterfragen, bewerten und unterstützen dann einen der Vorschläge.

Dazu fördern und nutzen wir die Potenziale der „T-Shaped People“. Bei diesen handelt es sich um Spezialist*innen, die ebenfalls über ein Wissen in anderen Disziplinen verfügen. Wobei diese anderen Disziplinen nicht vollends ausgeprägt sind. Die Mischung aus Breiten- und Spezialistenwissen erlaubt, das große Ganze zu erkennen und mit anderen Disziplinen zusammenzuarbeiten. Das „T“ visualisiert diese Verbindung sehr schön. Der Fuß steht für die Tiefe der Spezialisierung, das Dach für die geringere, aber vorhandene Breite in anderen Disziplinen. Ich favorisiere zudem das π -Shaped-Modell, das neben den T-Shaped-Eigenschaften auch das Thema Führung als weiteren Fuß berücksichtigt. Dabei geht es nicht darum, das Thema Führung mit aufzuzwingen, sondern ein Verständnis dafür zu schaffen und eben auch kurzfristige Führungsaufgaben zuzulassen – aber eben richtig.

Sinnbildlich: Führung ist heute weniger „eine eigentlich sehende Person mit verbundenen Augen durch einen Raum führen“ (Blinden gelingt das meist auch ohne Hilfe), als mehr „einem Menschen das Fahrradfahren beibringen“. Mehr loslassen, das Neue und Ungewohnte erleben lassen, bei Bedarf mit Rat und Tat zur Seite stehen.

Warum tun wir uns bei Führung im agilen Vorgehen schwer?

Bereits während meiner Ausbildung und dem späteren Studium gab es Schwerpunkte, wie die Personalführung oder das Personalmanagement. Diese brachten mir zwar den Menschen als Ressource näher, ließen aber die heute relevanten Ebenen der Führung außer Acht. Tatsächlich hatten wir das Personal vorrangig wie Material berechnet und auf dem Papier in Tätigkeiten oder Projekte verteilt. Man könnte meinen, dass Personal immer an der Hand geführt und bei Bedarf in eine Richtung geschickt werden müsse. Ist dem so? Einschlägige Studien belegen eine bessere Ausbildung als zu Beginn dieser Art des Personalmanagements.

Das Gleiche gilt für die private Nutzung von IT, die sich inzwischen auch in die älteren Semester ab 65 Jahren erstreckt. Eine Veränderung in den persönlichen Wünschen, mehr Selbstverwirklichung und sinnvolle Beschäftigung sind logische Konsequenzen. Zuletzt mündete dies in Begriffen wie „Generation X, Y und Z“, die in den Medien die Runde machten. Wir, die wir faktisch und je nach Modell in eine dieser Generationen fallen, sollen uns auf einen Umgang mit diesen Generationen vorbereiten. Wir stecken bereits mitten in diesen Ausprägungen. Wir (25-65 Jahre) sind ein Teil davon [1].

Nachträgliche Schulungen während meines Berufswegs behandelten das Thema Führung bereits mit diversen situativen/partizipativen Ansätzen, so zum Beispiel Variationen des Personalgesprächs, die Erarbeitung und Beteiligung an der Auswahl von persönlichen Zielen, wie auch Team- und Abteilungsziele. Im Nachgang bleibt ein fader Beigeschmack. Denn mir bleibt ebenso die Maximierung der Auslastung der „Ressource Mensch“ im Kopf, wie auch diverse Techniken zur Beeinflussung. Dass diesem Inhalt der nötige Zeitgeist fehlt, bestärken auch aktuelle Bücher, wie zum Beispiel „Führung? Führung!“ von Herbert Henzler (ehem. McKinsey).

Daneben ist mir durchaus bewusst, dass es keine einheitliche Sicht auf die agile Ar-

beit gibt. Erst im vergangenen Jahr hatte ich die Gelegenheit, bei einem angesehenen Dienstleister ein Event aktiv begleiten zu können. Ein Beitrag des Gründers eröffnete den Tag. Mir ist selten ein Mensch begegnet, der die individuellen Fähigkeiten der Kollegen so genau kannte. Darunter gab er auch hier und da eine private Floskel zum Besten. Dazu kommt, dass dieser Mensch zu führen versteht. Er prägt eine starke Vision, begeistert sein Umfeld und weiß individuelle Beiträge und Kritik wertzuschätzen. Er übergibt Verantwortung, ohne das nötige Interesse zu verlieren, hat ein außerordentliches politisches und wirtschaftliches Gespür und vieles mehr.

Das Motto der Veranstaltung versprach auch eine Antwort auf die zunehmende Herausforderung agiler Arbeit. Im Eröffnungsvortrag führten zwei Thesen zu einer anschließenden Diskussion zwischen uns, in der die unterschiedliche Sicht agiler Arbeit deutlich wurde.

These 1: Agile Arbeit ist konfus, der nötige Überblick fehlt.

Tatsächlich wurde als plakatives Beispiel ein Badezimmer dargestellt, in dem die Armaturen so angebracht waren, dass diese zu großen Teilen unbenutzbar waren oder zu einer Überflutung führen mussten. „Da hat jemand agil gearbeitet.“ Wirklich? Leider nein. Für mich vielmehr ein offensichtliches Gegenbeispiel. Ein (falscher) Plan wurde bis zuletzt verfolgt. Oder ein Monteur, der mangels Feedback und vielleicht fehlender Fachkompetenz offensichtliche Fehler machte. Das ist das Gegenteil agiler Arbeit. Kein Kundentest, keine Befähigung, keine Team- oder Kundenkommunikation. Wenn dies im genannten Beispiel der Fall war, fehlte die notwendige Führung, die im Rahmen ihres Verantwortungsbereiches nicht entgegengewirkt hat.

These 2: Die Kosten agiler Arbeit seien kaum berechenbar. Zudem: Schließlich wisse man nie, was man am Ende erhalte.

Dazu eine Gegenaussage eines Bereichsleiters, in dessen Umfeld ich ebenfalls im vergangenen Jahr als „agiler Coach“ im Einsatz war: „Seit der agilen Arbeit mit unseren Teams sind all die Projekte unter meiner Schirmherrschaft erfolgreich.“ Nun, ein klarer Fall der Sichtweise: In der agilen Arbeit bestreben wir die stetige Auslieferung ein-

setzbarer Ergebnisse. Das gilt für einen Gesamtauftrag wie für einzelne Funktionen, die zweitverwendet werden können. Vielleicht wurde eine ursprüngliche Vision tatsächlich am Ende der Projektlaufzeit erfüllt. Das Ergebnis unterschied sich von der ursprünglichen und konkreten Vorstellung einiger Bereichsvertretungen. Diese ließen sich durch entsprechende Team- und Kundenfeedbacks überzeugen. Für mich ist das ein indirektes Lob an die fachlichen Führungskräfte der Kompetenzfelder. Schließlich geht es nicht darum, wer „recht“ hat, sondern um ein unternehmerisches Handeln.

Personalkosten sind in beiden Fällen der agilen Arbeit leicht berechenbar, nachdem ein interdisziplinäres Team geschaffen wurde und eine erste Einarbeitung überwunden ist. Stabile Teams führen zu Kosten pro Planungszyklus, reifen in ihren Schätzungen und Prognosen gegenüber ihren Fähigkeiten und vertreten die Planung der Teilziele konsequent. Nie ist mir eine Prognose leichter gefallen. Leider erlebe ich oft Führungskräfte, die mit der Berechnung dieser Kosten betraut sind, sich aber nicht auf die Veränderungen, die agile Arbeit mit sich bringt, einlassen.

An dieser Stelle erwähne ich einen dritten Kommentar, den ich außerhalb dieser Begegnung wahrgenommen habe: „In der agilen Arbeit gibt es keine Planung.“ Tatsächlich habe ich auch in meiner Projektkarriere selten eine derart bewusste und präzise Planung erlebt. Lediglich die Planungsart derselben unterscheidet sich wesentlich. Sie erlaubt bewusste Anpassungen und vermeidet überflüssige Detailplanung in weiter Zukunft.

All diese (gelebten) Missverständnisse führen zu entsprechenden Herausforderungen, auch beim Thema Führung. Entweder beschäftigt uns die Arbeit, gegen diese Missverständnisse vorzugehen – oder wir stellen das Dasein der Führungsrolle an sich infrage. Es fehlt ebenfalls an der entsprechenden Aufklärungsarbeit in Verbindung mit einer transparenten und konstruktiven Anleitung durch methodisch geschultes Personal. Leider wird hier zu oft beim methodischen Personal gespart. Leider wird ebenso oft Personal vermittelt, das lediglich eine minimale praktische Erfahrung vorweisen kann (wenn überhaupt).

Ein weiteres Beispiel ist die gelebte Konkurrenz zwischen Rollen im agilen Kontext und Führungskräften, die (durchaus wertvolle) Ideen immer wieder spontan in den

Teams priorisieren, ohne die Arbeitsweise zu berücksichtigen. Diesen Kampf kann eine Führungskraft nur verlieren. Damit meine ich weniger in personeller Konsequenz, sondern hinsichtlich des zielführenden Ergebnisses, auf das wir hinarbeiten. Am Ende wird ein Umsetzungsteam in eine reaktive Arbeit verfallen. Die Rolle im Team, welche die Planung bisher übernommen hat, verwaltet die entgegen genommenen Aufgaben.

Diese Gerüchte, Selbstzweifel und Fehlbesetzungen führen zu einem fragmentierten Zustand im Verständnis agiler Arbeit. Zudem waren relevante Quellen für agile Führung lange Zeit Mangelware in unserem Sprachraum. Dem ist nicht mehr so. Tatsächlich entwickelt sich dieser Bereich mit zunehmender Geschwindigkeit.

Heute blicke ich mit Hoffnung auf viele verfügbare Ansätze, zum Beispiel auf die Werte und Prinzipien des agilen Manifests sowie auf Bestrebungen, den sieben Arten der Verschwendung in Lean („Muda“) eine achte hinzuzufügen: „Ungenutztes Potenzial von Personal“. Daneben nehmen erste Schulsysteme 2019 das Thema „Achtsamkeit“ zur Stärkung der Selbstwahrnehmung und Empathie in ihr Pflichtprogramm auf (leider nicht in Deutschland). Es gibt noch viele weitere Ansätze, die auch hierzulande in Barcamps, Meetups und immer öfter auch auf den größeren Konferenzen diskutiert werden. Darunter finden sich auch praxisbezogene Berichte, die immer seltener beschönigt werden. Dazu gehören auch neue Trends, die mehr oder weniger unerprobt zum Besten gegeben werden, aber Ihnen mehr Möglichkeit geben, einzelne Punkte in Ihren persönlichen Werkzeugkoffer aufzunehmen.

Wie das nötige Verständnis geschaffen werden kann

Zunächst ist es wichtig, dass Führungskräfte die agile Arbeit an sich verstehen. Das gilt ebenfalls für Umsetzungsteams und die zugehörigen agilen Rollen. Dazu gehört nicht nur das Lesen einschlägiger Quellen, sondern auch die praktische Fortbildung in Verbindung mit der Erfahrung agiler Arbeit in ihrer Anwendung. Ich sehe die Führungskraft als aktive Begleitung im jeweiligen Team. Daher muss besonders sie das methodische Handwerk verinnerlichen, um in der agilen Arbeit „mitspielen“ zu können.

Sinnbildlich: Stellen Sie sich vor, eine Gruppe Kinder entscheidet sich, von

Dame zu Schach zu wechseln. Sie beginnen zögerlich mit dem Spiel. Eines der Kinder hält aber an Dame fest und spielt dieses auch, ebenfalls auf dem Schachbrett, da die anderen Kinder das Damebrett weggelegt haben. Wie gut mag das funktionieren? Dieses Beispiel ist zugegebenermaßen etwas plakativ, aber der sinnbildliche Fall, der leider noch zu oft gelebt wird.

Diverse Schulungsanbieter bieten Fortbildungen zur agilen Arbeit an. Entsprechende Fortbildungen sind essenziell, auch wenn die Inhalte an die spezifischen Anwendungsfälle anzupassen sind. Die Fortbildung hilft Ihnen, Sinn, Unsinn und Abweichung in Ihrem Praxis-Umfeld zu unterscheiden. Oder sie ermöglichen, bewusste Abweichungen in Ihrem Unternehmen zu unterstützen und gegebenenfalls in die allgemein gelebte agile Arbeit aufzunehmen oder zu verwerfen (falls Sie sich dazu entscheiden, achten Sie bei der Auswahl des Trainings [1-5 Tage, je nach Anbieter und Inhalt] auf ausreichende Praxisübungen).

Bewusst trenne ich hier zwischen dem Verstehen und der direkten Mitwirkung einer Führungskraft in agilen Teams. Gerade disziplinarische Führungskräfte dürfen aus meiner Sicht keine der agilen Rollen in Teams besetzen. Eine Besetzung führt zu einer schädlichen Auswirkung im Teamgefüge. So würde zum Beispiel ein Scrum Master als Führungskraft einen zu starken Einfluss auf Teamentscheidungen mit sich bringen (gewollt oder ungewollt). Das Gleiche gilt für Product Owner in Scrum oder Kanban-„Master“ in Verbindung mit der disziplinarischen Führung (es ist schwer, über entsprechende Kanban-Rollen zu sprechen, wenn es sie doch eigentlich gar nicht geben sollte).

Teams halten die Eigenverantwortung für eine Funktion, ein Produkt, einen Prozess oder eine Dienstleistung in ihren Händen und verteilen sie auf das gesamte Team. Je nach momentanem Bedarf übernimmt in reifen Teams die Fachlichkeit, die Methodik oder die Technik situativ die Führungsrolle innerhalb dieses Teams. Als Führungskraft in Person ist es Ihre Aufgabe, diese Dynamik zu unterstützen und Ihre Erfahrung situativ zu teilen. Hier passt wieder eines der Sinnbilder dazu: Helfen Sie Ihrem Team dabei, das Radfahren zu erlernen.

Daneben wird agile Arbeit und Führung nicht nur für IT-Technologie oder auf

Teamebene verwendet. Es gilt für jeden Einsatzbereich: Es geht um eine Erarbeitung einer Vision in Eigenregie. Dazu erfolgt eine planerische Justierung in kurzfristigen Intervallen in Verbindung mit Nutzer-/Kundenfeedback in regelmäßigen Zyklen. Die Vermittlung zwischen Teams, die (Neu-)Justierung in Verbindung stehender Unternehmensbereiche und die Produktstrategie selbst treten für Führungskräfte stärker in den Vordergrund. Bereits bis zu dieser Stelle ist der Führungsbedarf recht hoch. Es mag nach einem „Damit mache ich mich doch selbst überflüssig!“ klingen. Dazu möchte ich Ihnen zwei Gegenargumente nennen:

1. Das ist tatsächlich ein höheres Ziel diverser Rollen im agilen Kontext. Fakt ist: Diese Rollen haben – irgendwann – etwas mehr Zeit für andere Tätigkeiten, wie zum Beispiel das schier unendliche Potenzial der Fortbildung ihrer Kollegen*innen und die unternehmensinterne Kommunikation. Der Bedarf versiegt nicht.
2. Sie sind Führungs-Spezialist und erhalten – irgendwann – mehr Zeit für weitere Optimierungen im Thema Führung. Oder Sie erhalten die Chance, andere Personen explizit auszubilden, Ihre Erfahrungen in andere Bereiche zu tragen usw. Der Bedarf versiegt nicht.

Über den Wirkungskreis hinaus blicken

Mit der Einführung und dem Ausbau der agilen Arbeitsweise erhöht sich das Risiko, angrenzende Unternehmensbereiche zu überumpeln. Veränderliche Planungen und die erhöhte Taktung der Auslieferungen, die sich in der agilen Arbeit ergeben, erhöhen das Risiko, Takt und Anschluss zu verlieren. Plötzlich sind Unternehmensprozesse, die in ihrer Dauer über diese Teamzyklen hinausgehen, Verhinderer des Teamfortschritts. Hier gibt es verschiedene mögliche Reaktionen, die konträr zu einer unternehmerischen Haltung stehen. So seien als Beispiele das Abwarten, die Anpassung der Teamarbeitsweise an Prozesse oder das Suchen nach Schlupflöchern genannt.

Aus diesem Grund ist es wichtig, dass sich bei der Einführung agiler Arbeit die Veränderung nicht nur auf einzelne Teams beschränkt. Unmittelbar nach der Einführung ergibt sich eine Dynamik, mit der weitere Bereiche eine Chance erhalten, unterstützende Prozesse zu vereinfachen, zu

verkürzen und somit mit höherer Reaktionsfreude eine konstruktive Zusammenarbeit zu wahren. Ich möchte mit dieser Aussage keine Bewertung einzelner Bereiche forcieren. Alle Bereiche leisten einen Beitrag zur Unternehmensvision. Sicherlich wurden bisher auch werthaltige Ergebnisse erzielt. Davon auszugehen, dass bei der Einführung agiler Arbeit in einem Unternehmen nur punktuelle Bereiche berührt werden und alle anderen wie immer weitermachen können, ist dagegen schlichtweg naiv. Dabei geht es nicht darum, dass alle nach dem gleichen agilen Modell arbeiten sollen. Es geht darum, sich darauf einzulassen und Unterstützung herzustellen. Insbesondere Führungskräfte stehen vor der zunehmenden Herausforderung, Personal in dieser Transformation zu unterstützen. Gegenrufe entstehen durch Unsicherheit und Unsicherheit durch mangelnde Aufklärung.

Daneben sind eine stärkere Teamvertretung und Transparenz auf taktischer und strategischer Ebene nötig. Akut verhängende Themen des Teamfortschritts werden gesammelt und übergreifend bearbeitet. Synergien werden erkannt und Optimierungen übergreifend herbeigeführt. Eigene Best Practice können auf andere Teams angewendet werden. Führungserfahrung wird geteilt und in Verbindung mit der gelebten Agilität optimiert. Strategische Unternehmensziele werden anhand operativer und taktischer Erfahrungen diskutiert, konkretisiert oder adaptiert.

Die Umsetzung dieser Aufgabe erfolgt zum Beispiel durch das Führungsinstrument OKR („Objectives and Key Results“). Aus einer übergeordneten Vision leiten umsetzende Rollen für sie relevante Schlüsselergebnisse ab. Dabei sind diese abgeleiteten Ziele in maximal drei Monaten erreichbar, messbar und auf fünf Ziele limitiert. Diese Aushandlung, Beteiligung und Frist erlauben eine starke Transparenz und eine rasche Reaktion auf veränderte Bedingungen. Die Ziele sollen ambitioniert, greif- und erreichbar sein. Eine Taktung, die sich sehr leicht mit der rhythmischen Vorgehensweise agiler Arbeit verbinden lässt.

Führungsaufgaben agiler Arbeit

Lassen Sie mich die Führungsaufgabe mit einer Scrum-Teamrolle verdeutlichen: der Scrum-Master-Rolle. Sie arbeitet nicht aktiv auf technischer oder fachlicher Ebene mit. Sie ist rein methodisch bestrebt, die Qualität des Teamergebnisses zu erhöhen.

Gleichzeitig wahrt sie die Einhaltung des Scrum Frameworks. Sie ist also methodisch gesehen in einer Führungsrolle. Die Scrum-Master-Rolle ist bewusst eingeschränkt in ihrer Entscheidungsbefugnis und ihrem Wirkungskreis. Auf operativer Ebene wirkt die ausschließliche Führungskraft unterstützend auf einzelne Personen ein. Sie unterstützt individuell bei fehlendem Know-how, der Besetzung einzelner Rollen oder als Netzwerk-Partner*in gegenüber anderen Unternehmensbereichen. Sie agiert ebenfalls als Sparringspartner für die agilen Rollen der Teams. Hierzu möchte ich ein Beispiel aus meiner Erfahrung mit Ihnen teilen.

Eine Führungskraft erhielt neben ihren übergreifenden Aufgaben den Auftrag, einmal am Tag Zeit mit den Teams zu verbringen. Die Aufgabe lautete in etwa wie folgt: „Versuche, einmal am Tag 30 Minuten im Raum deiner Teams zu verbringen. Keine Einmischung, nur da sein und ‚in das Team hineinhören‘.“ Mehr war es nicht. Die Führungskraft setzte sich an unterschiedlichen Tagen mit Laptop oder Tablet wechselweise in die Büros der sechs verschiedenen Teams. Sie ging dort ihren Tätigkeiten nach, die in Einzelarbeit zu erledigen waren. Mal geschah dies auf einem freien Platz, mal auf einer Couch, die vom Unternehmen in einem der Büros zur Verfügung gestellt wurde. Was geschah?

In den ersten Tagen gab es verwunderte Blicke, Gespräche wurden leise geführt, zaghafte und vereinzelte kurze Unterhaltungen zwischen allen Parteien. Alle fühlten sich nicht so ganz wohl in dieser Konstellation. Vier Wochen später: Die Führungskraft ist nun nicht mehr jeden Tag in den Büros. Dafür besonders häufig an Freitagen oder während und nach Teambesprechungen, die in den Büros abgehalten werden. Was ist geschehen? Bereits in der zweiten Woche merkten die Teams, dass die Führungskraft nicht zur Kontrolle oder als Entscheider in ihrer Fach- oder Methodenkompetenz erscheint. Ebenfalls kam es öfter zu spontanen Anfragen an die Führungskraft: „Mit wem könnten wir...?“, „Was denken Sie über...?“ „Was gibt es Neues zu...?“ und vieles mehr. Zudem hatte sich die Führungskraft dazu entschieden, regelmäßige Veranstaltungen mit den Teams durchzuführen, in denen Austausch jeder Art und eine übergreifende „News aus dem Unternehmen“ verkündet wurde. Warum? Die Führungskraft hatte bisher wenig Zu-

gang zur kompletten Belegschaft in ihrem Bereich, da es keine Gelegenheit dazu gab. Es gab hier und da Personalgespräche oder Feedbacks zu Teamergebnissen oder ähnliche Austauschformate mit großen Abständen zueinander.

Aus dem ersten gegenseitigen „Beschnuppern“ wurde eine Gelegenheit für Austausch geschaffen. Die Führungskraft kannte nun die Kolleginnen und Kollegen. Sie arbeitete hart und erfolgreich daran, alle Namen zu verinnerlichen. Viele Eigenheiten und besondere Fähigkeiten des Personals waren ihr bekannt und dies funktionierte auch andersherum. Heute wird die Führungskraft als Individuum wahrgenommen, das in einigen Bereichen einen großartigen Erfahrungsschatz aufweisen kann und ausgezeichnet vernetzt ist. Es wird somit weniger als Vergabe- und Kontrollorgan wahrgenommen. Dieses vermeintliche Feindbild wird in manchen Umgebungen, sogar auf einschlägigen agilen Veranstaltungen, leider noch immer gefördert.

Stattdessen erreicht eine Führungskraft Bereiche im Unternehmen, die anderem Personal unbekannt oder sogar vermeintlich verschlossen sind. Sie ist in der Lage, Türen zu öffnen. In beide Richtungen. Im Gegenzug erhält sie aktive und ehrliche Unterstützung aus den Teams bei ihren Aufgaben. Schließlich ist eine Führungskraft stellenweise (und oft unnötigerweise) alleine mit Themen betraut, die durch Teamunterstützung leichter zu lösen sind. Sie ist nun ein Teil des Teams („die Schnittstelle“) mit ihrer eigenen Handlungskompetenz, greifbaren Ergebnissen am Tagesende, und sie fühlt sich auch im agilen Kontext wertgeschätzt. Viele Kolleginnen und Kollegen wissen einfach nicht, was Führung eigentlich ist oder macht. Manchen macht das Angst. Ich sehe das als zusätzliche Gelegenheit, Führung gemeinsam und vielleicht auch neu zu gestalten.

Bei diesem und anderen Beispielen ist es letztendlich egal, um welchen Führungsstil es sich handelt. Genauso, wie sich Teammitglieder in ihre Rollen im Team einfinden müssen, müssen sich Führungskräfte in ihre Teaminteraktion (neu) einfühlen. Dabei ist es weniger erforderlich, sich von vorhandenen agilen Rollen abzugrenzen, sondern rollenspezifische Überschneidungen als Chance zu sehen, mit den eigenen Mitteln partnerschaftlich zu agieren. So ist es weniger eine Frage des Führungsstils, sondern

des Tuns. Passives Abwarten hängt Führungskräfte ab.

Zudem gibt es viele Ergänzungsmöglichkeiten neben der individuellen Personalführung. Insbesondere in den genannten Themen Netzwerk, Kommunikation und Vermittlung sehe ich die Herausforderung der Führungskräfte im Allgemeinen stärker als bisher. Diese neue Sicht in Verbindung mit dem Wissen über die eingesetzten agilen Standards ermöglichen das Mitspielen in der agilen Arbeit – und zwar in beide Richtungen. Frei nach Kaizen („Veränderung, zum Besseren“) geht es darum, offen für Veränderung zu bleiben, diese gemeinschaftlich zu erarbeiten und einen steilen Fluss beizubehalten.

Und zum Schluss noch ein Tipp aus der Praxis zur direkten Anwendung. Springen Sie auf den Zug der neuen Kommunikation der Teams mit auf. Oft werden mit der Einführung agiler Arbeit neue Kommunikations- und Customer-Relationship-Management-Tools eingeführt. Wenn Sie verstehen, wie diese funktionieren und wenn Sie diese mitnutzen können, werden Sie jederzeit aktuelle Informationen über Brennpunkte, Erfolge und Ideen erhalten. Als Ergänzung zu einem persönlichen Kontakt bekommen Sie so eine aktuelle Sicht in nahezu jedem Detaillierungsgrad.

Quelle

[1] https://digitalcollection.zhaw.ch/bitstream/11475/1379/1/Rigor_Lorena_W.BA.BO.GM.pdf



Daniel Reinold

daniel@visual-braindump.de

Daniel Reinold ist Wirtschaftsinformatiker und IT-Business-Manager mit Praxiserfahrung im klassischen Projektmanagement (PMBOK/PRINCE2) und den agilen Vorgehen, Ansätzen und Frameworks (SCRUM, Kanban und Design Thinking). Im Bereich der Wissensarbeit unterstützt er nicht nur mit einem großen Methodenkoffer, sondern auch durch eine ausgeprägt unternehmerische Haltung in strategischen Belangen. Daniel Reinold ist Gründer und Geschäftsführender Gesellschafter von Visual Braindump, einem Unternehmen mit dem Ziel, Wissensarbeit in Teams mittels visueller Techniken zu unterstützen.

Data Analytics 2020

28. & 29. April in Düsseldorf
AUF DEM WEG ZUM DATENGETRIEBENEN UNTERNEHMEN



Der Handel und die Auswirkungen der Digitalisierung

Jens Horstmann, Trevisto AG

Die Digitalisierung stellt für die überwiegende Mehrheit der Händler eine große Herausforderung dar. Dieser Aussage stimmten von 583 in einer Studie der Bitkom Research zwei Drittel aller befragten Händler zu. Dabei sind die Chancen der Nutzung von Daten für neues Geschäft im Handel immens. Durch den Einsatz von Big-Data-Technologien, datengetriebenen Analysen und Methoden der künstlichen Intelligenz wird der Händler wieder mehr Zeit gewinnen, sich um den Kunden zu kümmern. Aufgaben wie Handelsprognosen, Preis- und Sortimentgestaltung können durch intelligente Algorithmen abgelöst werden – und sie werden es bereits.

Seit Jahren erfährt der Einzel- und Großhandel zunehmende Konkurrenz durch Online-Käufe. Die Kunden bestellen ihre Waren vermehrt im Internet. Im Jahr 2018 belief sich der E-Commerce-Anteil am Einzelhandelsumsatz auf rund 8,4 Prozent [1]. In einem progressiven Szenario könnte der Anteil der Online-Umsätze am Einzelhandel bis zum Jahr 2024 auf 17,0 Prozent ansteigen [2]. Dies gilt für alle Güter des täglichen Bedarfs bis hin zu langlebigen Produkten und Investitionsgütern. Berücksichtigt man, dass im Lebensmittel Einzelhandel der Online-Verkauf in der Breite noch kaum eine Rolle spielt, ergibt sich für den restlichen Einzelhandel (ohne Lebensmittel) bereits heute ein Online-Anteil am Umsatz von rund 15 Prozent [3]. Unternehmen wie Amazon, Zalando und PayPal zeigen am Branchenbeispiel Handel sehr eindrücklich die Potenziale, aber auch, wie stark sich die Kombination von hybriden, also digitalen und physischen, Diensten – bestehend aus digitalem Marktplatz, sicherem Online-Bezahlen und physischer Logistik – auf traditionelle Geschäftsmodelle auswirkt [4]. Daraus folgt: Der stationäre Handel hat den Wandel mitgemacht – oder er wird nicht mehr im Markt präsent sein.

Ein Beispiel für den Existenzkampf im Handel sind Filialisten der Fashion-Branche: Die Insolvenzanmeldungen der Modehäuser Gerry Weber, Vögele, Strenesse oder K&L allein im Jahr 2019 zeugen von einer nicht bewältigten Umwälzung durch die Digitalisierung des Handels [5]. Zwar kommen im jeweiligen Einzelfall weitere Grün-

de für die Insolvenz hinzu, jedoch einte alle genannten traditionellen Mode-Anbieter der Umbruch, dass Menschen zunehmend im Internet einkaufen. Immer geöffnet, ein größeres Sortiment, Einkauf von nahezu jedem Ort, Warenverfügbarkeit, schnelle Logistik sind die Vorteile des Online-Handels.

Ein Weg aus dieser Abwärtsspirale ist der Erlebnis-Einkauf; er bietet neben Beratung, Ausprobieren und Anfassen einen Grund dafür, warum der Online-Handel den stationären Handel nicht ganz ersetzen kann. Das Einkaufserlebnis attraktiver zu machen und die Zahlungsbereitschaft der Kunden für Freizeiterlebnisse auszuschöpfen, ist zumindest in Consumer-Märkten ein Trend [6]. Der Großhandel als Mittler zwischen zahlreichen Herstellern von Waren und ihren Verwendern in Industrie, Handwerk und Einzelhandel benennt in einer Studie als größte Herausforderungen der Digitalisierung IT-Sicherheit, IT-Anwendungen (Software) und IT-Kosten neben Personal und Qualifikation [7].

Der Mangel an qualifiziertem Personal wird auch in der jüngsten Bitkom-Umfrage als Grund für die Schwierigkeiten des Handels im Umgang mit der Digitalisierung genannt. 71 Prozent der 504 befragten stationär und online tätigen Groß- und Einzelhändler gaben an, dass sie große Probleme bei der Personalbeschaffung haben, wenn es darum geht, Mitarbeiter mit digitalen Fachkenntnissen zu gewinnen. Die Herausforderung der Digitalisierung für sich genommen rangiert dagegen mit 65 Prozent nur an zweiter Stelle, was mit der Investitionsbereitschaft einhergeht: 45 Prozent

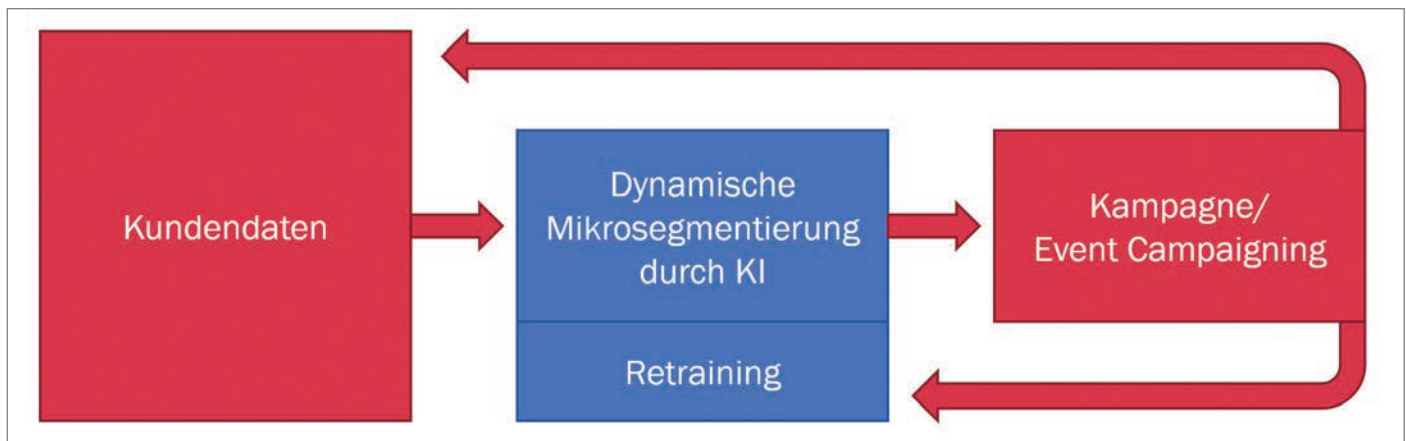
der Unternehmen investierten 2019 mehr in Digitalisierung [8].

Digitalisierung weist das Potenzial auf, zu mehr Wachstum und Profitabilität beizutragen. Digitale Geschäftsmodelle hängen jedoch nicht nur von der IT und einer schnellen Verfügbarkeit des Internets ab, sondern nach wie vor auch von der Nähe zum Kunden und der Kenntnis seiner Bedürfnisse – eine klassische Stärke traditioneller Handelsunternehmen.

Big Data im Handel

Einer stärkeren Nutzung von Daten für eine genauere Ansprache kommt dabei zukünftig eine besondere Bedeutung zu. Dabei ist diese Entwicklung nicht neu, denn der gut aufgestellte Handel war schon immer ein Vorreiter in Sachen Datennutzung für die Vermarktung. So analysierte das Einzelhandelsunternehmen Dodenhof mit seinen Einkaufszentren in Ottersberg und Kaltenkirchen seine Kundenbasis bereits vor mehr als sechs Jahren: Durch die Auswertung der Daten von 450.000 Kunden sowie 150.000 Transaktionen bei 140 Mailings im Jahr konnte eine achtmal höhere Response-Quote und Conversion-Rate als der Branchendurchschnitt erzielt werden [9].

Die dafür zugrunde gelegte Basis liefert Big-Data-Technologie. Zukünftig werden Handelsunternehmen ihre Absatzmengen genauer planen und ihre Kunden individualisierter, zur richtigen Zeit, im richtigen Kanal mit einem für sie relevanten Angebot ansprechen können, indem sie die Potenziale von Künstlicher Intelligenz (KI) für



Quelle: Trevisto AG

ihr Unternehmen nutzen. Doch die Potenziale von KI bleiben nicht lediglich auf den Absatz und das Kundenmanagement beschränkt.

Der Einsatz von Künstlicher Intelligenz (KI) in der Wertschöpfungskette des Handels

Künstliche Intelligenz ist eine Schlüsseltechnologie der Digitalisierung, die Wirtschaft und Gesellschaft grundlegend verändern wird. Betrachtet man die gesamte Wertschöpfungskette des Handels, von der Erzeugung der Waren, der Veredelung, dem Transport und der Lagerung bis zum Verkauf und Verbrauch, so ergeben sich in nahezu allen Stufen Mehrwerte durch den gezielten Einsatz von Künstlicher Intelligenz.

Wie sieht das in der Praxis aus? Unternehmen in der Produktion und im Handel von Frischeprodukten für den Lebensmittelhandel stehen vor der Herausforderung, dass durch das Mindesthaltbarkeitsdatum die Logistik deutlich komplexer ist: Zu viel Ware bedeutet teurer Verderb oder Rabattaktionen, welche die niedrige Marge im Lebensmitteleinzelhandel (LEH) nicht hergibt. Zu wenig Ware bedeutet, dass man die Kundenbedürfnisse nicht befriedigt und damit sowohl Umsatz verliert als auch Kundenzufriedenheit einbüßt.

Die Firma Köckerling mit Stammsitz in Westfalen produziert Schnittware im Obst- und Gemüsebereich. Kunden hierfür sind neben dem LEH vor allem Gastronomiebetriebe und Großküchen. Diese schätzen die einfachere Verarbeitung dieser Produkte in ihren Küchen mit vielen ungelerten Kräften. Das Problem ist aber, dass die Produkte nicht langfristig bestellt werden – üblich ist eine Bestellung am Tage der Lieferung.

Das heißt, dass ca. 90 Prozent der Bestellungen erst am Tag der Produktion und Auslieferung hereinkommen – und die beginnt gegen 11 Uhr. Nun könnte man die Produkte auf Vorrat produzieren, wenn das Mindesthaltbarkeitsdatum von vier Tagen dem keinen Strich durch die Rechnung machen würde. Eine Vorproduktion und Lagerung ist nicht möglich, da es sich um leicht verderbliche Waren handelt und die Laufzeit bis zum Ende der Mindesthaltbarkeit bei Handelskunden gerne beispielsweise für Warenpräsentation in Anspruch genommen wird. Es gilt also: je frischer, desto besser. Demzufolge ist eine Just-in-time-Produktion also die Regel. Dazu kommt auch, dass die Rohware nicht unbegrenzt haltbar ist und einen Zulauf von der Bestellung zum Lager – wenn sie aus dem Ausland kommt – von bis zu sieben Tagen hat.

Hier kommt dann die KI ins Spiel. Wenn man auf der einen Seite in der Lage ist, die Absatzmenge des nächsten Tages und den Rohwarenverbrauch der nächsten sieben Tage genau vorherzusagen, dann ist dies der Schlüssel für einen wirtschaftlichen und nachhaltigen Betrieb.

Die Firma Köckerling führt ein derartiges System gerade ein: Zusammen mit der Trevisto AG aus Nürnberg wird ein System entwickelt, das einerseits die Zwischenprodukte vor der Verpackung prognostiziert, andererseits den Verbrauch an Rohware der nächsten sieben Tage zur Optimierung des Lagers und des Einkaufs. In die Prognose fließen sowohl allgemein verfügbare Daten ein – Ferienzeiten, Wochentage, Feiertage und Wetterdaten – als auch die Absatzzahlen auf verschiedenen Produktionsebenen der letzten drei Jahre.

Die ersten Ergebnisse sind beeindruckend. Die Vorhersage auf Ebene der Zwi-

schenprodukte vor Verpackung erreicht eine Genauigkeit von bis zu 98,5 Prozent. Bei den Top-20-Produkten ist keine Prognose schlechter als 90 Prozent. Damit kann zu Schichtbeginn ohne Risiko mit voller Kraft die Produktion begonnen werden. Die Auswirkungen sind deutlich. Die klassische Hektik bei Eingang der großen Bestellungen ist deutlich gesunken, die Arbeit läuft effizienter und die Fehlerquote hat sich verringert. Gleichzeitig ist die Anzahl verspäteter Lieferungen gesunken, genauso wie die Überproduktion, die dann entsorgt werden müsste.

Die Prognose des Rohwarenverbrauchs ist naturgemäß nicht so genau. Die Wetterabhängigkeit einiger Produkte und die sinkende Präzision von Wettervorhersagen, je weiter sie in der Zukunft liegen, machen die Prognosen hier deutlich schwieriger. Trotzdem sind die Ergebnisse sehr gut. Die Genauigkeit erreicht bei den wichtigsten Produkten wie Paprika, Salaten und Tomaten 80 bis 90 Prozent. Mehr als genug, um den Einkauf besser zu steuern und so zwei wirtschaftliche Ziele zu erreichen: die deutliche Erhöhung der Verfügbarkeit bei gleichzeitiger Vermeidung von Verderb. Wenn also eine Firma wie Köckerling bei anderen regionalen Großhändlern kurzfristig einkaufen will, wird dem Verkäufer klar sein, dass gekauft werden muss, und es werden wegen der sofortigen Verfügbarkeit höhere Preise aufgerufen.

Das Ergebnis: Köckerling erreicht mit KI in der Einkaufs- und Produktionssteuerung weniger Verderb, bessere Einkaufspreise, weniger Personalkosten in der Produktion und zufriedener Kunden. Die Investitionskosten hielten sich auch in engen Grenzen, sodass der erwartete ROI (Return on Investment) bei ca. sechs Monaten liegt.

Gewinnung von Customer Insights durch Datenanalyse mit KI

Um sich zukünftig einen Wettbewerbsvorteil zu erarbeiten oder auszubauen, müssen Unternehmen konsequent kundenorientiert agieren. Das bedeutet heute vor allem, permanent neue Customer Insights aus Daten zu generieren, um diese in die Entwicklung und Vermarktung von Produkten und Dienstleistungen einfließen zu lassen und durch permanentes Lernen aus (Kunden-)Daten die Kundensegmentierung stetig zu verbessern.

Betrachten wir es historisch, so hatten wir früher im Einzelhandel eine sehr präzise persönliche Ansprache. Der Einzelhändler kannte bei seinen Stammkunden die Bedürfnisse und Vorlieben meist genau und konnte bei Neukunden schnell Muster erkennen, wenn er sie ansprach. Dadurch war er in der Lage, sehr schnell ein entsprechendes Angebot zu unterbreiten. Nach und nach ist diese Ansprache – egal ob analog oder später digital – verloren gegangen. Selbstbedienung, gedruckte Werbung und andere Änderungen im Handel haben hier die Zielgruppen zusammengefasst und die Ansprache eher global werden lassen.

Mit der Massenwerbung im industriellen Zeitalter und dem Aufkommen des Versandhandels, später dem Onlinehandel, wurde der Trend immer stärker – pauschale Ansprache aus Kostengründen, mit dem Pferdefuß, zu breit zu streuen. Dann wurden Zielgruppen entwickelt, Kundensegmente erarbeitet und diesen Affinitäten für Produktgruppen zugeordnet. Die Gruppen werden immer detaillierter und die enormen Datenmengen aus dem Online-Reich der Unternehmen lassen die Treffgenauigkeit immer weiter ansteigen. Allerdings haben Zielgruppen zwei entscheidende Nachteile: Zielgruppe oder Segment ist immer noch eine Schublade, die zwar immer kleiner geworden ist, aber den individuellen Menschen nur unvollständig wiedergibt. Nur weil jemand ein MacBook hat und Autozeitungen liest, muss er noch lange keinen BMW fahren.

Die Erarbeitung von immer mehr Segmenten ist zudem aufwendig und teuer. Der Trend zum Segment-of-One ist im Marketing seit Jahrzehnten ein geflügeltes Wort. Die Mass Customization, also die individualisierte Massenfertigung, wie beispielsweise beim Adidas-Schuhkonfigurator (create your own – miadidas) zu sehen, trägt diesem Trend Rechnung. Damit ist sie jedoch nur für große Unternehmen gewinnbringend einsetzbar.

Auch hier kommt die KI ins Spiel. Mit KI ist es möglich, eine noch nie dagewesene Mikrosegmentierung durchzuführen. Die Segmentgröße geht tatsächlich bis auf einen Kunden herunter, und dies ist nicht nur für Großunternehmen machbar, sondern auch für Handelsunternehmen aller Stufen in der Wertschöpfungskette. Das heißt, wenn wir mit einem solchen System Kaufimpulse generieren, dann kann es sein, dass bestimmte Produkte nur einem Kunden angeboten werden.

Der Zeitpunkt ist auch definitiv passend für solche Systeme. E-Mails und Werbebanner bis hin zu ganzen Websites lassen sich inzwischen automatisiert individuell erstellen. Zwei Kunden können auf die gleiche Website gehen und sehen etwas Unterschiedliches – nicht nur auf den Displayflächen: Die eigentliche Website unterscheidet sich. Das Gleiche gilt für Newsletter. Diese lassen sich heute individuell aus Bausteinen zusammensetzen, maßgeschneidert für den jeweiligen Empfänger. Basis dafür ist die Analyse der historischen Kundendaten der letzten Jahre. Daraus lässt sich mit moderner KI genau bestimmen, welche Artikel für einen Kunden interessant sind – oder eben nicht.

Das Besondere an der KI-gestützten Verkaufsimpuls-generierung ist der permanente Lernvorgang. Aus dem Erfolg oder Misserfolg der verschiedenen Aktivitäten trainiert sich das Netz immer weiter und passt so auch die Erkenntnisse über die Kunden permanent an. Dadurch werden die Ergebnisse immer besser und auch Änderungen an den Interessen frühzeitig erkannt – ob ein Kind in die Familie gekommen ist, der Wohnort gewechselt hat oder einfach eine extrinsisch motivierte Interessenänderung, wie beispielsweise die Änderung des Ausgabeverhaltens für hochwertige Konsumgüter aufgrund einer signifikanten Gehaltserhöhung, eingetreten ist. Die KI wird es erkennen und darauf reagieren.

Am Ende sind wir wieder bei dem Kolonialwarenhändler vom Anfang der Geschichte des Warentauschs, der bei jedem Kunden genau wusste, was er ihm anbieten kann. Nur können wir dies heute in der individuellen Massenansprache mit neuen Technologien datengestützt fundieren.

Fazit

Die Digitalisierung im Handel ist heute ohne den Einsatz von KI nicht mehr denkbar. Die gute Nachricht dabei ist, dass diese zukunfts-trächtige Technologie in der Umsetzung nicht ausschließlich den großen Playern vorbehalten ist. Die aus KI resultierenden Mehrwerte sind gerade bei bereits

vorhandenen (Kunden-)Daten in Handelsunternehmen vergleichsweise einfach zu heben. Darüber hinaus können nicht nur Verkäufe eigener Produkte und Dienstleistungen laufend beobachtet, sondern ebenso Preise und Aktionen des Wettbewerbs analysiert werden. Was für den Händler von früher in Zeiten von Tante-Emma-Läden galt, gilt heute mehr denn je: Die neuen Technologien erlauben eine systematische Generierung von Customer Insights und ermöglichen durch permanentes Lernen einen immer differenzierteren Austausch mit dem Kunden bis hin zu individuellen Echtzeit-Angeboten für den Kunden in allen Kanälen. Der ständige Dialog mit dem Kunden wird zukünftig datengestützt möglich – One-to-One-Marketing wird Realität.

Quellen

- [1] statista 2019, <https://de.statista.com/statistik/daten/studie/38846/umfrage/anteil-des-online-handels-am-einzelhandel/>, abgerufen am 06.12.2019
- [2] ibi research an der Universität Regensburg, <https://www.marketing-boerse.de/news/details/1902-anteil-e-commerce-im-handel-bis-zu-17-prozent/152835>, abgerufen am 06.12.2019
- [3] Ebd.
- [4] Bayerisches Staatsministerium für Wirtschaft, Landentwicklung und Energie (Hrsg.): Dr. Heiko Wrobel, Alexander Köhler, Estella Cäsar et. al.: Der Großhandel in Bayern – Marktstruktur und Digitalisierung, München, aktualisierte Fassung Februar 2019, S. 14.
- [5] Vgl. FAZ (2019): Modehändler Forever 21 meldet Insolvenz an, FAZ, Nr. 228, 01.10.2019, S. 19
- [6] Vgl. deutschlandfunkkultur.de/zukunft-des-handels-beim-shoppen-zaehlt-das-erlebnis vom 07.10.2016
- [7] Vgl. Bayerisches Staatsministerium für Wirtschaft und Medien, Energie und Technologie (2016): Der Großhandel in Bayern Marktstruktur und Digitalisierung, Juni 2016, S.70. Aktualisierte Fassung Februar 2019.
- [8] Vgl. Bitkom (2019): <https://www.bitkom.org/Presse/Presseinformation/Digitalisierung-sichert-deutschen-Handelsunternehmen-Zukunft> vom 24.09.2019
- [9] Vgl. Franziska von Lewinsky (2013): Wie aus Daten Kommunikation entsteht, in: Absatzwirtschaft, Sonderausgabe dmexco 2013, S. 30.



Jens Horstmann

jens.horstmann@trevisto.de

Jens Horstmann gründete gemeinsam mit Jürgen Engler 2009 die Trevisto AG. Zuvor war er u.a. bei Oracle Deutschland tätig. Die Trevisto AG ist ein IT-Dienstleister mit Hauptsitz in Nürnberg. Mit seiner Expertise bietet das Unternehmen seinen Kunden Mehrwerte in den Bereichen Big Data, Business Intelligence, Analytics und Künstliche Intelligenz.

Wir begrüßen unsere neuen Mitglieder

Persönliche Mitglieder

- Anis Briki
- Rene Rütter
- Udo Martin
- Christoph Benjamin Krämer

Firmenmitglieder DOAG

- Stadtwerke Rosenheim GmbH & Co. KG,
Christian Fußstetter
- HANSAINVEST Hanseatische Investment GmbH,
Frank Hilgendorf
- Database Consulting sp z o.o.,
Rafał Stryjek
- Creditplus Bank AG,
Oliver von Düsterloh



Termine Q1/2020

März

03

04.03.2020

Regionaltreffen Berlin/Brandenburg

Michel Keemers
Berlin

12.03.2020

Regionaltreffen NRW

Martin Schmitter, Torsten Rosenwald
Köln

17. - 19.03.2020

JavaLand 2020

in Brühl bei Köln

19.03.2020

CloudLand Preevent

in Brühl bei Köln

31.03.2020

Berliner Expertenseminar zum Thema Oracle Performance Diagnostics and Tuning

Richard Foote
Berlin

April

04

16.04.2020

Regionaltreffen Nürnberg

Martin Klier
Nürnberg

23.04.2020

Regionaltreffen München/Südbayern

Andreas Ströbel
München

28. & 29.04.2020

Data Analytics 2020

in Düsseldorf

Mai

05

05. - 07.05.2020

APEX Connect 2020

in Brühl bei Köln

07.05.2020

Regionaltreffen Rhein-Neckar

Frank Stöcker
Mannheim

13.05.2020

Regionaltreffen München/Südbayern

Andreas Ströbel
München

21.05.2020

Regionaltreffen Nürnberg

Martin Klier
Nürnberg

25. & 26.05.2020

DOAG Datenbank 2020

Düsseldorf

Impressum

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

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

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

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

Redaktion:

Sitz: DOAG Dienstleistungen GmbH
(Anschrift s.o.)
ViSdP: Mylène Diacquenod
Redaktionsleitung Red Stack Magazin:
Martin Meyer

Redaktionsleitung Business News:
Marcos López
Kontakt: redaktion@doag.org
Weitere Redakteure (in alphabetischer Reihenfolge): Wolf G. Beckmann, Carsten Firus, Frank Hoffmann, Jens Horstmann, Markus Lohn, Marcos López, Christian Luda, Daniele Massimi, Martin Meyer, Marco Mischke, Daniel Reinold, Stefan Seck, Casimir Schmid, Frank Schneede, Michael Schulze, Ingo Sobik, Jürgen Sieben, Grégory Steulet, Günther Stürner, Michael Weinberger, Carsten Wiesbaum, Ales Zeman

Titel, Gestaltung und Satz:

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

Fotonachweis:

Titel: © Daniil Peshkov | <https://de.123rf.com>
S. 5: © Martin Klier | DOAG
S. 7: © camera work | <https://stock.adobe.com>
S. 10: © Markus Gann | <https://de.123rf.com>
S. 16: © golubovy | <https://de.123rf.com>
S. 22: © Olivier Le Moal | <https://de.123rf.com>
S. 29: © dizanna | <https://de.123rf.com>
S. 34: © Christopher Titze | <https://de.123rf.com>
S. 37: © Tithi Luadthong | <https://de.123rf.com>
S. 47: © belchonock | <https://de.123rf.com>
S. 52: © buchachon | <https://de.123rf.com>
S. 61: © Bakhtiar Zein | <https://de.123rf.com>
S. 64: © alphaspirt | <https://de.123rf.com>

S. 69: © DOAG | <https://www.doag.org>
S. 70: © mariok | <https://de.123rf.com>
S. 74: © youichi4411 | <https://de.123rf.com>
S. 84: © Andriy Popov | <https://de.123rf.com>
S. 93: © gmast3r | <https://de.123rf.com>

Anzeigen:

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

Druck:

WIRmachenDRUCK GmbH,
www.wir-machen-druck.de

Alle Rechte vorbehalten. Jegliche Vervielfältigung oder Weiterverbreitung in jedem Medium als Ganzes oder in Teilen bedarf der schriftlichen Zustimmung des Verlags.

Die Informationen und Angaben in dieser Publikation wurden nach bestem Wissen und Gewissen recherchiert. Die Nutzung dieser Informationen und Angaben geschieht allein auf eigene Verantwortung. Eine Haftung für die Richtigkeit der Informationen und Angaben, insbesondere für die Anwendbarkeit im Einzelfall, wird nicht übernommen. Meinungen stellen die Ansichten der jeweiligen Autoren dar und geben nicht notwendigerweise die Ansicht der Herausgeber wieder.

Inserentenverzeichnis

B4BMEDIA.NET AG www.e3zine.com	S. 49	MuniQsoft Training GmbH www.muniqsoft-training.de	S. 3	Robotron Datenbank-Software GmbH www.robotron.de	S. 21
DOAG e.V. www.doag.org	U 2, U 3, S. 67, 83, 89	PROMATIS software GmbH www.promatis.de	S. 81	Trivadis AG www.trivadis.com	U 4
MuniQsoft Consulting GmbH www.muniqsoft-consulting.de	S. 25				



APEX *connect*
by DOAG

5. - 7. Mai 2020
in Brühl



WIR LEBEN DIGITALISIERUNG



Die Digitale Transformation macht vor keiner Branche halt und verändert Wertschöpfungsketten und Strukturen auch Ihres Unternehmens. Sie müssen sich neuen Herausforderungen stellen. In andere Richtungen denken. Ihr Geschäftsmodell anpassen und weiterentwickeln, damit Sie auch in Zukunft wettbewerbsfähig bleiben. Wir sind mittendrin im Geschehen und gestalten partnerschaftlich Ihren Weg ins Zeitalter der Digitalisierung. Sprechen Sie mit uns.

m.trivadis.com/digitalisierung | info@trivadis.com

BASEL | BERN | BRUGG | BUKAREST | DÜSSELDORF | FRANKFURT A.M.
FREIBURG I.B.R. | GENÈVE | HAMBURG | LAUSANNE | MANNHEIM
MÜNCHEN | STUTTGART | WIEN | ZÜRICH

trivadis