Nr. 6 | Dezember 2013 | ISSN 09 36-0360 | www.doag.org

DOAG News

Deutsche ORACLE-Anwendergruppe e.V.

So tickt das System

Tools zum Monitoring

- Nagios
- Enterprise Manager
- Ops Center

Monitoring mit Bordmitteln

- Eigene Tools entwickeln
- Die Top 10 der SQL-Befehle

Industrie 4.0

Interview mit Prof. Dr. Michael ten Hompel vom Fraunhofer Institut

OPITZ CONSULTING

Menschen.
Innovationen.
Lösungen.




Was kommt nach dem Oracle Warehouse Builder?

Oracle setzt beim Thema ETL zukünftig auf den Oracle Data Integrator.

Mit dem jüngsten Statement of Direction kündigt Oracle an, dass der OWB mit der Version 11g R2 das letzte große Release erfahren hat und Zertifizierungen gegen Datenbanken Releases größer 12c R1 seitens Oracle nicht mehr geplant sind.

- Was bedeutet das für Ihre Oracle Warehouse Builder Projekte?
- Können die bisher getätigten Investitionen gesichert werden?
- Ist der Oracle Data Integrator die einzige Alternative?
- Was kann man jetzt schon in laufenden OWB-Entwicklungen berücksichtigen?

OPITZ CONSULTING hilft Ihnen dabei, Ihre Fragen zu beantworten und die Umstellung Ihrer bewährten ETL-Prozesse auf den Oracle Data Integrator oder alternative Technologien effizient und möglichst investitionsschonend durchzuführen.

Gemeinsam finden wir mit Ihnen die passenden und individuell auf Ihre Bedürfnisse zugeschnittenen Lösungen und Vorgehensmodelle!

Erfahren Sie mehr über unsere Leistungen im Bereich Oracle Warehouse Builder unter www.opitz-consulting.com/owb_und_jetzt

Ihr direkter Ansprechpartner ist Jochen Wilms.
Telefon: +49 201 892994-0 · E-Mail: jochen.wilms@opitz-consulting.com

Unser Tipp: Die neuesten Oracle Schulungen

- **Oracle Database 12c: Administration Workshop**
  Nächste Termine: 12.05.2014 / 01.09.2014 / 15.12.2014
- **Oracle BI Publisher 11g R1**
  Nächste Termine: 24.03.2014 / 11.08.2014 / 15.12.2014

OPITZ CONSULTING ist zertifizierter Schulungspartner der Oracle University.

Mehr über unsere Schulungen und unser außergewöhnliches Schulungszentrum erfahren Sie unter www.opitz-consulting.com/schulungen

Jetzt anmelden!

ORACLE **Platinum Partner**

Specialized
Data Warehousing

Christian Trieb
Leiter der Datenbank Community

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

das Schwerpunktthema dieser Ausgabe ist „Monitoring“. Ein Thema, das wichtig ist – doch wie ich auch aus eigener Erfahrung bestätigen kann, oftmals eher ein Randthema bei der Konzeption von Datenbanken. Da stehen die Hype-Themen wie Hochverfügbarkeit, Performance oder Ähnliches weit häufiger im Fokus. Trotzdem wird das Monitoring von jedem benötigt, der eine Datenbank administriert.

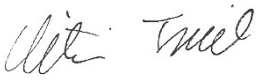
Für den Betrieb von Datenbanken ist es sogar existenziell, dass man über ein gutes Monitoring-Werkzeug verfügt. Es sollte einem die notwendigen, aber auch nicht zu viele Meldungen über den Zustand der Datenbanken liefern. Der Überblick muss schnell dargestellt und auf Wunsch müssen Details angezeigt werden. Das kann ein gut konfiguriertes grafisches Tool genauso leisten wie eine Sammlung von in der Praxis bewährten Skripten. In diesem Spannungsfeld bewegen sich die in diesem Heft beschriebenen Werkzeuge, Methoden und Vorgehensweisen.

Letztendlich kommt es immer noch auf den Menschen – den Oracle-Datenbank-Administrator – an, der gut ausgebildet die richtigen Schlüsse zieht und geeignete Maßnahmen aufgrund der eintreffenden Meldungen ergreift.

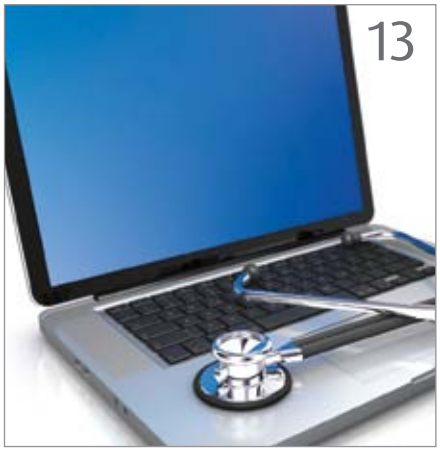
In dem vorliegenden Heft finden Sie genügend gute Informationen, damit Sie das „Monitoring von Datenbanken“ in Ihrer täglichen Arbeit verbessern können. Auch im kommenden Jahr wird das Thema bei der DOAG eine wichtige Rolle spielen, so zum Beispiel auf der Real World Performance Tour am 19. Februar 2014 in München und auf der DOAG 2014 Datenbank am 3. Juni 2014 in Düsseldorf.

In diesem Sinne wünsche ich Ihnen viel Spaß beim Lesen dieser Ausgabe. Darüber hinaus wünsche ich Ihnen jetzt schon gesegnete Weihnachten und ein gutes neues Jahr.

Ihr



Interview mit Prof. Dr. Michael ten Hompel, Leiter des Fraunhofer-Instituts, Seite 6



Monitoring von Oracle-Datenbank-Servern mit Nagios, Seite 13

Hardware- und OS-Monitoring für Oracle Server und Solaris, Seite 41

DOAG intern

Aktuell

Best Practice

Monitoring

Hidden Secrets

DOAG intern

Spotlight

Sonntag, 22. September 2013

In seiner Eröffnungs-Keynote auf der Oracle OpenWorld stellt Oracle-CEO Larry Ellison eine neue In-Memory-Option vor, die um den Faktor 100 schnellere Datenbank-Abfragen liefern soll. Quasi als Antwort auf die SAP-Datenbank HANA will Oracle bei der In-Memory-Technologie wieder das Maß der Dinge sein. Eine weitere Ankündigung betrifft die Oracle Database Backup Logging Recovery Appliance, ein neues Backup-Verfahren für die Datenbank. Die Keynote zur JavaOne, die parallel zur OpenWorld in San Francisco stattfindet, wird in diesem Jahr wieder an alter Stelle im Moscone Center gehalten. Peter Utzschneider, Oracle Vice President Java Product Management, geht unter anderem nochmals die Fakten durch, die im März 2014 mit Java SE8 zu erwarten sind. An oberster Stelle stehen die Lambda-Expressions, die mit einer neuen Syntax Java-Code besser lesbar und wartbar machen sowie durch Parallel-Ausführung eine bessere Performance bringen.

Dienstag, 24. September 2013

„Larry Ellison ist wegen des America´s Cup leider verhindert". Die angekündigte und von vielen Teilnehmern erwartete zweite Keynote des Oracle-Chefs muss Thomas Kurian, Oracle Executive Vice President of Product Development, übernehmen. Die Leute verlassen in Scharen die Halle. Es macht sich Unverständnis darüber breit, dass der Oracle-Chef sein Hobby über die Wertschätzung seiner Kunden stellt.

Mittwoch, 25. September 2013

Die DOAG-Vorstände Dr. Dietmar Neugebauer, Fried Saacke und Christian Trieb treffen sich am Rande der OpenWorld mit Tom Scheirsen, EMEA User Groups Relationship Manager von Oracle. Sie bekräftigen die Forderung der DOAG, dass das Programm der von Oracle organisierten internationalen Usergroup-Events zukünftig wieder deutlich auf die Anforderungen und Bedürfnisse der Anwendergruppen hin abgestimmt sein muss, damit sich der Aufwand für die Teilnehmer der Usergroups überhaupt lohnt.

Dienstag, 8. Oktober 2013

Das Organisations-Team der DOAG kommt mit Vertretern und Dienstleistern des Nürnberg Convention Centers zu einem Kick-off-Meeting zusammen, um die Durchführung der DOAG 2013 Konferenz + Ausstellung zu besprechen. Die Teilnehmer sind davon überzeugt, auch in diesem Jahr wieder optimale Rahmenbedingungen für die Jahreskonferenz der DOAG bieten zu können.

Mittwoch, 9. Oktober 2013

Dr. Frank Schönthaler, Leiter der Business Solutions Community, eröffnet in Berlin die DOAG 2013 Applications. 280 Anwender treffen sich auf der führenden Oracle-Applications-Konferenz in Europa drei Tage lang zu informativen Fachvorträgen, anregenden Diskussionen und interessanten Gesprächen.

Mittwoch, 24. Oktober 2013

In einer Telko mit Simon Ritter, weltweit verantwortlicher Java Technology Evangelist bei Oracle, schildert Fried Saacke, DOAG-Vorstand und Geschäftsführer, das Konzept des JavaLand, einer neuen DOAG-Veranstaltung am 25. und 26. März 2014 im Phantasialand in Brühl bei Köln. Simon Ritter versteht die Bedeutung des Events und dass Oracle dort vertreten sein sollte.

Dienstag, 29. Oktober 2013

Fried Saacke, DOAG-Vorstand und Geschäftsführer, und Wolfgang Taschner, Chefredakteur der DOAG News und der Java aktuell, vertreten die DOAG auf der EclipseCon Europe 2013 in Stuttgart. Sie machen Werbung für das JavaLand und spüren bei vielen Gesprächen eine Begeisterung für die neue Veranstaltung.

Mittwoch, 30. Oktober 2013

Das Programm-Komitee der Java-Community beschließt ein sehr interessantes und spannendes Programm für die JavaLand-Konferenz, das einen guten Mix aus renommierten Speakern aus dem deutschsprachigen und internationalen Raum enthält. Es ist keine einfache Aufgabe, aus den mehr als 400 eingereichten Vorträgen die 60 besten auszuwählen.

Fotos: Wolfgang Taschner

Fried Saacke (links) und Dr. Frank Schönthaler (rechts) beim Interview mit Prof. Dr. Michael ten Hompel

„Die Cloud-Umgebung kann sehr hilfreich sein …“

In der Informationstechnologie kommen einschneidende Änderungen auf die Unternehmen zu. Fried Saacke, DOAG-Vorstand und Geschäftsführer, Dr. Frank Schönthaler, DOAG-Vorstand und Leiter der DOAG Business Solutions Community, und Wolfgang Taschner, Chefredakteur der DOAG News, sprachen darüber mit Prof. Dr. Michael ten Hompel, Leiter des Fraunhofer-Instituts für Materialfluss und Logistik IML.

Sie haben die vierte industrielle Revolution angekündigt. Was ist mit Industrie 4.0 zu erwarten?
Prof. Dr. ten Hompel: Ich war das mit der vierten industriellen Revolution nicht allein, habe mich aber voll dahinter gestellt. Hinter Industrie 4.0 steht für mich das „Internet der Dinge“, das mit dem Voranschreiten der cyber-physischen Systeme schon im Entstehen ist. Es geht nicht mehr nur um einige Daten, die an einer Palette befestigt sind, sondern um wirklich intelligente Systeme. Das eröffnet uns ganz neue Möglichkeiten.

Welches sind aus Ihrer Sicht die wichtigsten Herausforderungen, die sich daraus für die Informationstechnologie ergeben?
Prof. Dr. ten Hompel: Zunächst einmal beschäftigen wir uns bereits seit fünfzehn Jahren mit dem „Internet der Dinge“. In dieser Zeit ist sehr viel neue Technologie entstanden. Leider erhält diese angewandte Forschung in Deutschland oft nicht die notwendige Anerkennung. Die wichtigste Herausforderung liegt jetzt darin, intensiv in die Entwicklung dieser cyber-physischen Systeme einzusteigen und das Feld nicht den Amerikanern oder Koreaner zu überlassen.

Sind diese Herausforderungen an die Industrie gerichtet oder eher an die Politik?
Prof. Dr. ten Hompel: Es ist in erster Linie eine Herausforderung für die Unternehmen. So, wie am Beispiel von Oracle die gesamte Entwicklung in den USA stattfindet, sollte die Entwicklung der vierten industriellen Revolution aus Deutschland kommen. Bei den cyber-physischen Systemen halten wir die Technologie in den Händen und müssen jetzt schnell zu fertigen Produkten kommen, bevor das die anderen tun.

Neue Informations- und Kommunikationstechnologien sind stets auch Treiber für vollkommen neuartige Geschäftsprozesse. Können Sie uns hierzu einige aus Ihrer Sicht wichtige Beispiele aus der Logistik nennen?
Prof. Dr. ten Hompel: Wir haben die richtigen Prozessoren, wir beherrschen das Energie-Harvesting und wir haben die Display-Technologien. Es fehlt allerdings an den Applikationen und an der Integration beispielsweise in die Oracle Engineered Systems.

Betrachten Sie das als Bedrohung, wenn die amerikanische Software-Industrie so stark ist im Vergleich zu der unsrigen?
Prof. Dr. ten Hompel: Gerade in den letzten Jahren sehe ich eine vertane Chance. Wir haben hier Unternehmen, die jedes für sich hochintelligente Applikationen entwickeln können. Ich sehe allerdings keine adäquate Wertschätzung dafür. Deshalb besteht die Gefahr, dass wir wieder einmal die grundlegende Idee und die entsprechende Technologie dafür haben, aber den Schritt nicht schaffen, gemeinsam das große Ganze zu schaffen. Ich befürchte, dass wir es ähnlich wie in der E-Commerce-Zeit verpassen, das große Rad zu drehen. Damals konnten es sich die entscheidenden Leute auch nicht vorstellen, dass eines Tages mehr als fünf Prozent des Versandhandels über das Internet abgewickelt werden. Heute liegen wir bei siebenundsiebzig Prozent!

Innovationen in der IT-Technologie sind Treiber für neue Geschäftsmodelle. Können Sie uns dazu einige Beispiel aus Ihrem Bereich nennen?

Prof. Dr. ten Hompel: Es gibt mittlerweile den intelligenten Behälter, den man als logistischen Dienstleister für die Produktion sehen kann. Darüber hinaus wird es ganz neue Geschäftsprozesse in der Zusammenarbeit der Unternehmen geben, bei denen die Cloud eine wichtige Rolle spielen wird, um die Lieferkette in hohem Maße zu automatisieren. Auch mit den intelligenten Transportfahrzeugen lassen sich ganz neue Geschäftsfelder im Lagerbereich eröffnen.

In dem Projekt Logistics Mall werden Standards entwickelt, um die Welten der Prozesse und der Programmierung zusammenzuführen. Sind hier schon erste Ergebnisse zu verzeichnen?

Prof. Dr. ten Hompel: Einige gute Beispiele – etwa im Warehouse Management – sind schon operativ. Wir sind überrascht über das große Interesse an der Software. Das betrifft auch große Unternehmen, die heute über einen grundlegenden Wandel ihrer Software nachdenken.

Sollte sich hierbei die Programmierung nach den Prozessen richten oder ist es besser, die Prozesse anzupassen?

Prof. Dr. ten Hompel: Wenn wir unsere Software in die Cloud bringen und auf einem gemeinsamen Repository arbeiten lassen, können die einzelnen Module sehr schnell untereinander kommunizieren. Es geht primär nicht darum, die Prozesse zu standardisieren, sondern die Software und deren Umgebung und Schnittstellen, um eine zukunftssichere und flexible Rechnerumgebung zu erhalten.

Welchen Beitrag kann das Cloud Computing leisten?

Prof. Dr. ten Hompel: Die Cloud-Umgebung kann sehr hilfreich sein, sofern einheitliche Standards vorhanden sind. Das Ziel sollte sein, Standard-Business-Objekte zu realisieren.

Große Software-Lieferanten versuchen hingegen momentan alles in ihrer eigenen Welt abzubilden.

Prof. Dr. ten Hompel: Das damit verbundene Customizing ist leider noch nicht überwunden. Ich bin jedoch davon überzeugt, dass insbesondere in

Das Fraunhofer-Institut für Materialfluss und Logistik IML

Das Fraunhofer IML gilt als die Adresse für alle Fragestellungen zu ganzheitlicher Logistik und arbeitet auf allen Feldern der inner- und außerbetrieblichen Logistik. Am Fraunhofer IML, gegründet 1981, arbeiten zurzeit 190 Wissenschaftler sowie 250 Doktoranden und vordiplomierte Studenten, unterstützt durch Kollegen in Werkstätten, Labors und Servicebereichen.
Nach Projekt- und Kundenbedarf zusammengestellte Teams schaffen branchenübergreifende und kundenspezifische Lösungen unter anderem im Bereich der Materialflusstechnik, des Warehouse Managements, der Geschäftsprozessmodellierung, der simulationsgestützten Unternehmens- und Systemplanung sowie in den Bereichen Verkehrssysteme, Ressourcenlogistik und E-Business. Das „Internet der Dinge“ wird Fraunhofer-weit vom Fraunhofer IML gemanagt. Im Bedarfsfall kann das IML auf 17 000 Mitarbeiter in 80 Einrichtungen der Fraunhofer-Gesellschaft zurückgreifen.
Das europaweit zurzeit größte Logistikforschungsprojekt ist der EffizienzCluster LogistikRuhr mit 120 Partnerunternehmen und elf Forschungseinrichtungen.
Neben Dortmund sind Frankfurt/Main, Hamburg, Prien am Chiemsee, Lissabon und Peking weitere Standorte.
Es ist geplant, dass die DOAG 2014 Logistik + SCM im Fraunhofer-Institut in Dortmund stattfindet.

der Logistik die Logistik- und die IT-Dienstleistungen zusammenwachsen können und müssen. Wir sprechen von hybriden Dienstleistungen, die mit Sicherheit auch wieder neue Geschäftsfelder eröffnen.

Zunächst als Hype-Thema gebrandmarkt, entwickelt sich Big Data in immer mehr Unternehmen zu einem der wichtigsten Investitionsfelder. Welche Bedeutung hat Big Data in der Logistik?
Prof. Dr. ten Hompel: Big Data spielt eine große Rolle, weil sich die Multi-Channel-Vertriebswege ändern. Dabei fallen extrem viele Daten, zum Beispiel aus den Sozialen Medien an.

In welche Richtung sollten sich IT-Unternehmen wie Oracle entwickeln, um auf die Herausforderungen des kommenden Jahrzehnts vorbereitet zu sein?
Prof. Dr. ten Hompel: Oracle wäre gut beraten, diese Entwicklung nicht in den USA zu machen, sondern hier in Deutschland. Die entscheidenden Innovationen in den Bereichen „Fertigung" und „Logistik" finden momentan hier statt.

Welchen Stellenwert hat dabei eine Anwendergruppe wie die DOAG?
Prof. Dr. ten Hompel: Sie sollten dringend darauf hinarbeiten, Oracle davon zu überzeugen, den weltweiten Zeitvorsprung, den wir hierzulande in der Logistik haben, mit deutschen Partnern gemeinsam zu nutzen.

Prof. Dr. Michael ten Hompel

Angesichts dieser großen Themen stellt sich die Frage, inwieweit kleinere und mittlere Unternehmen mit diesen Entwicklungen Schritt halten können. Sehen Sie hier Lösungsansätze?
Prof. Dr. ten Hompel: Ich hoffe, dass unser Mittelstand bei diesen Themen einsteigt. Es gibt verschiedene Möglichkeiten der Zusammenarbeit sowie entsprechende Förderprogramme. Wir arbeiten als Fraunhofer-Institut übrigens deutlich mehr mit kleineren und mittleren Unternehmen zusammen als mit großen Konzernen.

Das von Ihnen geleitete Fraunhofer-Institut hat 2011 mit einem großen Experiment die Effizienz von künstlich gesteuerten Fahrzeugen untersucht. Sind daraus auch Verbesserungsvorschläge für den Individualverkehr auf der Straße entstanden?
Prof. Dr. ten Hompel: Der Straßenverkehr ist nicht unser Thema. Wir beobachten dort zwar die Entwicklung, konzentrieren uns jedoch auf die Mobilität in der Logistik. Hinsichtlich des Verkehrs auf der Straße sind natürlich große Überlegungen notwendig, um die Veränderungen in der Logistik über ein Verkehrssystem abbilden zu können, das dafür im Grunde nicht konzipiert worden ist. Ich glaube aber, dass unsere künstlich gesteuerten Fahrzeuge die Halle bald verlassen können. Die Technik dafür beherrschen wir bereits.

Sie haben 2006 ein Gebrauchsmuster für einen Torwart-Roboter angemeldet. In welcher Klasse könnte dieser heute mithalten?
Prof. Dr. ten Hompel: Wenn wir unseren Robokeeper auf die höchste Stufe stellen, hält er alles. Es gibt nur links und rechts oben eine Ecke, die er nicht erreichen kann. Einige Fußballer trainieren mit dem Roboter, um genau diese Lücke exakt zu treffen. So verbessern sie die Präzision ihres Torschusses.

Zur Person:
Prof. Dr. Michael ten Hompel
Prof. Dr. Michael ten Hompel ist Inhaber des Lehrstuhls für Förder- und Lagerwesen an der Universität Dortmund und geschäftsführender Institutsleiter am Fraunhofer-Institut für Materialfluss und Logistik IML. Er studierte Elektrotechnik an der RWTH Aachen und promovierte an der Universität Witten/Herdecke. Neben seiner wissenschaftlichen Tätigkeit ist Prof. ten Hompel auch als Unternehmer tätig gewesen. So gründete er 1988 die GamBit GmbH und führte das Unternehmen, das sich vorrangig mit der Entwicklung und Realisierung von Warehouse-Management-Systemen beschäftigt, bis zu seinem Ausscheiden im Februar 2000 als geschäftsführender Gesellschafter.

Stichwort: Das Internet der Dinge
Die Grundidee ist einfach: Päckchen, Paletten und Behälter werden durch einen Chip gekennzeichnet, der neben Produktinformationen zusätzlich auch deren Transportziel speichert. Wie Datenströme im Internet finden Sendungen ihren Weg zum Ziel selbst. Kommt eine Sendung an eine Sortiermaschine, teilt sie den Bestimmungsort mit, wird entsprechend eingeordnet und zielgerichtet weiterbefördert. Diese Prozesse laufen schnell und autonom ohne eine zentrale Instanz ab. Mit anderen Worten: Selbst ist das Paket.

Monitoring mit optimalen Ergebnissen

Ralf Durben, ORACLE Deutschland B.V. & Co KG

Monitoring ist kein Selbstzweck, sondern dient in erster Linie der Kontrolle über Systeme, deren reibungsloser Betrieb von großer Bedeutung ist. Die Ermittlung der dazu notwendigen Informationen sollte dabei möglichst effizient und korrekt erfolgen.

Es gibt immer verschiedene Wege, ein Ergebnis zu erzielen, und das gilt natürlich auch zum Beispiel für das Monitoring von Oracle-Datenbanken. Wenn man aus Oracle-Sicht in die Vergangenheit blickt, hat alles mit einem Online-Monitoring in „SQL*DBA“ begonnen, begleitet von Skripten zur Analyse von internen Datenbank-Statistiken. Diese Form des Monitorings, also ein Tool, das einen Live-Blick in den Betrieb einer Datenbank bietet, ist an sich schön anzuschauen und hilfreich, wenn zum Zeitpunkt eines Problems der DBA zufällig auf den Online-Monitor blickt und dann auch noch die gegebene Situation schnell genug erfasst. In der Praxis zeigen sich aber schnell die Nachteile dieses Ansatzes:

- Vergangene Problem-Situationen sind nicht erfasst und gehen unter
- Bei mehreren Oracle-Datenbanken, und das ist wohl der Normalfall, müssen dauerhaft entsprechend viele Monitoring-Fenster geöffnet sein. Eine konzentrierte Beobachtung aller dieser Fenster ist in der Regel nicht möglich.

Aus diesen beiden Gründen sollte ein Monitoring-Tool die erfassten Daten oder Informationen für eine spätere Analyse speichern, was die meisten heutzutage auch anbieten. Bei der Ermittlung der Daten jedoch gibt es große Unterschiede. Es sind zwei grundlegende Verfahren möglich: externes und internes Monitoring.

Externes Monitoring

Ein externes Monitoring besteht darin, dass mittels einer Datenbank-Sitzung durch die Verwendung von SQL-Kommandos Daten aus der Datenbank erfasst werden. Mit dieser Methode lassen sich prinzipiell die Verfügbarkeit eines Systems überwachen sowie Performance-Daten erfassen. Typischerweise benutzen Monitoring-Tools Agenten oder Prozesse, die in regelmäßigen Abständen Skripte ausführen und die gesammelten Daten in einem zentralen Repository speichern.

Bis einschließlich Database 9*i* war dieses Verfahren die einzige von Oracle erlaubte Methode für ein Monitoring. Sie hat den Nachteil, dass dabei eine Datenbank-Sitzung erforderlich ist, die zusätzlich zu den laufenden Anwendungen Last in der Datenbank erzeugt. Das führt dazu, dass diese Form des Performance-Monitorings sehr sorgsam durchgeführt werden muss. Eine Erfassung aller signifikanten SQL-Kommandos, die in der Datenbank verarbeitet werden, ist darüber nicht ratsam.

Wie schon angedeutet, gab es noch mit Database 9*i* keine von Oracle erlaubte Alternativ-Methode, und so hat der Oracle Enterprise Manager bis zur Version 9*i* nur diese Form des Monitorings im Rahmen des „Datenbank Diagnostics Pack“ angeboten. Dabei werden die Skripte und SQL-Kommandos von dezentral auf den jeweiligen Servern laufenden Agenten ausgeführt.

Internes Monitoring

Das Oracle-Datenbank-System besteht aus passiven Dateien und einer aktiven

Instanz; in Oracle Database 12*c* können sogar mehrere Datenbanken von einer Instanz betrieben werden. Eine Instanz besteht aus Hauptspeicher und Prozessen, die ihrerseits auch für ein Performance-Monitoring genutzt werden können. Die dafür relevanten Daten sind hauptsächlich im Hauptspeicher-Bereich der Instanz vorhanden und können von einem Instanz-Prozess sicher und vollständig erfasst werden.

Seit Oracle Database 10*g* ergänzt diese Form des Monitorings das Datenbank Diagnostics Pack. Damit ist es möglich, in sehr kurzen Intervallen – zum Beispiel im Bereich weniger Sekunden – Informationen über Datenbank-Sitzungen und ihre Aktivitäten zu erfassen und für eine spätere Analyse zu speichern. Aufgrund des hohen Umfangs der durch diese Methode ermittelten Daten belässt Oracle diese Monitoring-Daten dezentral in den einzelnen Datenbanken und transportiert sie nicht in ein zentrales Repository.

Der große Vorteil des internen Monitorings besteht darin, dass in extrem kurzen Abständen eine Vielzahl von Daten erfasst werden können. Im Gegensatz zum externen Monitoring eignet es sich aber nicht für eine Live-Überwachung, da ein Ausfall des überwachten Systems auch das Monitoring stoppen würde. Daher ist immer eine Kombination aus beiden Ansätzen sinnvoll.

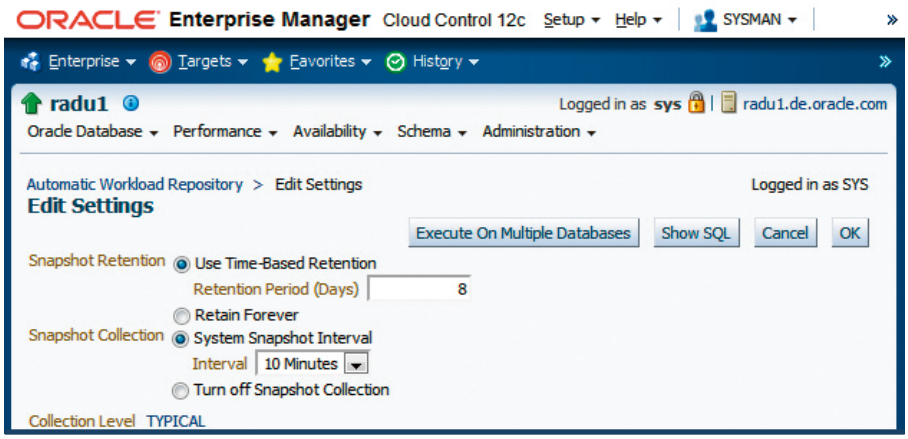


Abbildung 1: Auszug von Standard-Metriken für eine Oracle-Datenbank

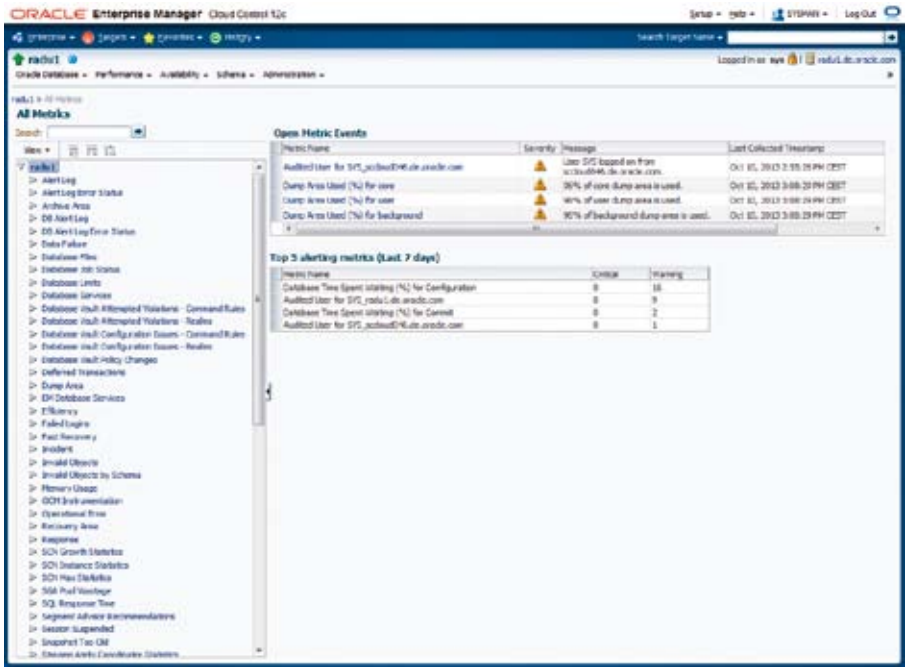


Abbildung 2: Verfügbarkeit eines Zielsystems im letzten Monat

Der Oracle-Weg

Oracle Enterprise Manager Cloud Control ist für Oracle das Werkzeug zur Verwaltung von Oracle-Produkten, also auch für die Oracle-Datenbank. Dabei sind externes und internes Monitoring in einem Gesamtkonzept kombiniert. Im Rahmen des externen Monitorings durch den Oracle-Enterprise-Manager-Agenten ist eine Vielzahl von Metriken nach Oracle Best Practice vordefiniert, inklusive Erfassungs-Intervalle und Bewertungs-Schwellenwerte (siehe Abbildung 1). Nachdem eine neue Datenbank in Cloud Control hinzugefügt wird, startet dieses Monitoring automatisch. Damit beginnt auch die Lebendüberwachung eines Zielsystems in Cloud Control automatisch, und dessen Verfügbarkeit ist für jeden Zeitraum, der in der Speicherdauer enthalten ist, schnell angezeigt (siehe Abbildung 2).

Benutzer des Enterprise Manager können diese Metriken auch an die eigenen Bedürfnisse anpassen. Darüber hinaus können sie eigene Metriken erstellen, die in der GUI „Metric Extensions“ genannt werden, um das externe Monitoring ihrer Oracle-Datenbanken beliebig zu erweitern. Die Einstellungen der genutzten Metriken lassen sich in Form von Monitoring-Templates definieren, um diese später einfacher auf neu zu überwachende Zielsysteme anzuwenden.

Warnungen und kritische Zustände von Metriken werden über vielfältige Wege kommuniziert, darunter Standards wie E-Mail oder SNMP. Über Betriebssystem-Kommandos lassen sich praktisch auch alle sonstigen Wege realisieren. Die Anbindung an externe Systeme, wie zum Beispiel Ticket-Systeme oder andere Leitstände, wird über Konnektoren realisiert.

Die durch dieses externe Monitoring erfassten Daten liegen zentral im EM-Repository, werden automatisch zu Stunden- beziehungsweise Tagesdaten aggregiert und unterliegen dort einer beliebig einstellbaren Speicherdauer. Die Speicherung findet in partitionierten Tabellen statt, über die maximale

Anzahl von Partitionen ist die Speicherdauer festgelegt. Per Default liegt diese bei sieben Tagen für Einzeldaten, etwa zweiunddreißig Tage für die Stundendaten und rund einem Jahr für die Tagesdaten. Eine neue Einstellung findet über die Prozedur „set_retention“ im Package „gc_interval_partition_mgr“ im Schema „SYSMAN“ des Repository statt.

Für alle Datenbanken ab der Version 10*g* nutzt Enterprise Manager auch das interne Monitoring, vorausgesetzt das Datenbank-Diagnostics-Pack ist eingeschaltet. Dabei liegen die erfassten Daten wie beschrieben in den jeweiligen Ziel-Datenbanken, was sich in der GUI von Cloud Control dadurch bemerkbar macht, dass für deren Anzeige oder Auswertung eine Datenbank-Verbindung besteht, man sich also eventuell an die Datenbank anmelden muss.

Durch die extrem performante Erfassung von Performance-Daten können sehr detaillierte Untersuchungen auch in die Vergangenheit vorgenommen werden. Dabei unterscheidet man bei der Erfassung zwischen zwei Fällen:

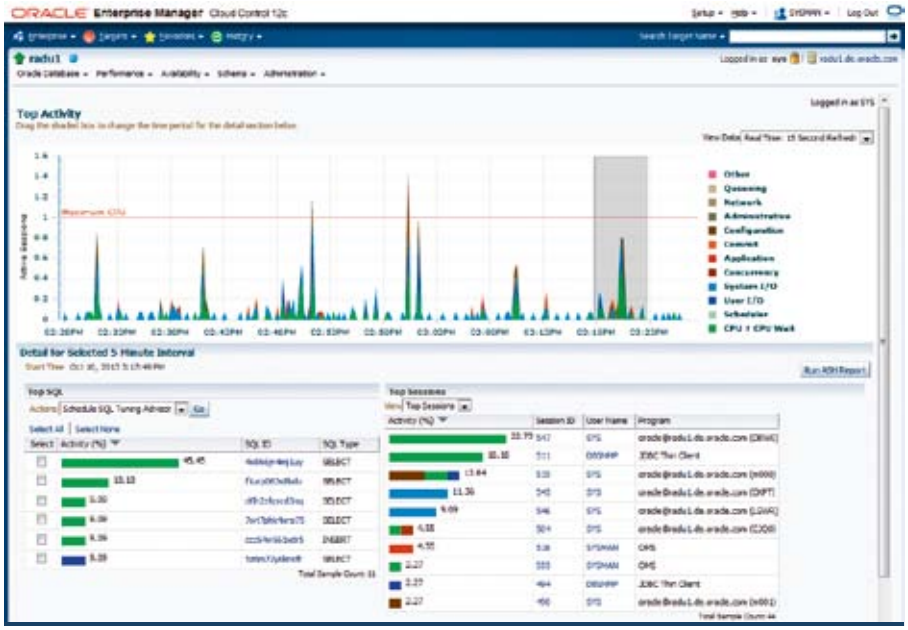


Abbildung 3: Vergleich zweier Zeiträume

- Allgemeine Datenbank-Statistiken
- Informationen zu einzelnen Datenbank-Sitzungen

Die allgemeinen Datenbank-Statistiken werden per Default im Intervall von dreißig Minuten erfasst. Dieses Intervall kann auf bis zu zehn Minuten heruntergesetzt werden. Die dabei gewonnenen Daten lassen sich im Prinzip mit den Ergebnissen des Statspack vergleichen, sind aber viel umfangreicher. Sie sind in einem Bereich namens „Automatic Workload Repository (AWR)“ im Tablespace „SYSAUX“ gespeichert.

Die Auswertung dieser Daten findet automatisch nach jeder Erfassung statt. Man kann sie aber auch jederzeit manuell starten, wobei auch beliebige Zeiträume miteinander verglichen werden können (siehe Abbildung 3).

Informationen zu einzelnen Datenbank-Sitzungen (per Default jeweils die Top-25-Sessions) werden im Sekundenbereich erfasst. Dabei sind für jede erfasste Session Ausführungsstatistiken, SQL-Kommandos und vieles mehr im Bereich namens „Active Session History (ASH)“ gespeichert (siehe Abbildung 4).

Beide Bereiche können direkt mit SQL- und PL/SQL-Befehlen angesprochen und genutzt werden. In Cloud Control jedoch verschwimmt die Grenze und die Daten stehen in ihrer Gesamtheit direkt zur Verfügung. Die Speicherdauer für die Ergebnisse des internen Monitorings beträgt per Default acht Tage und kann mit einem Aufruf angepasst werden (siehe Listing 1).

Eine Speicherdauer von zwanzig Tagen ergibt 20 * 1.440 Minuten = 28.800 Minuten. Das Erfassungs-Intervall betrifft nur den Bereich „WAR“ und liegt per Default bei dreißig Minuten. Beide Einstellungen sind auch in Cloud Control durchführbar, wobei in der GUI die Speicherdauer in Tagen angegeben wird – Cloud Control rechnet die Angabe dann in Minuten um (siehe Abbildung 5).

Abbildung 4: Die Aktivitäten der Top-Sessions in einer Datenbank ca. 5 Minuten in der Vergangenheit

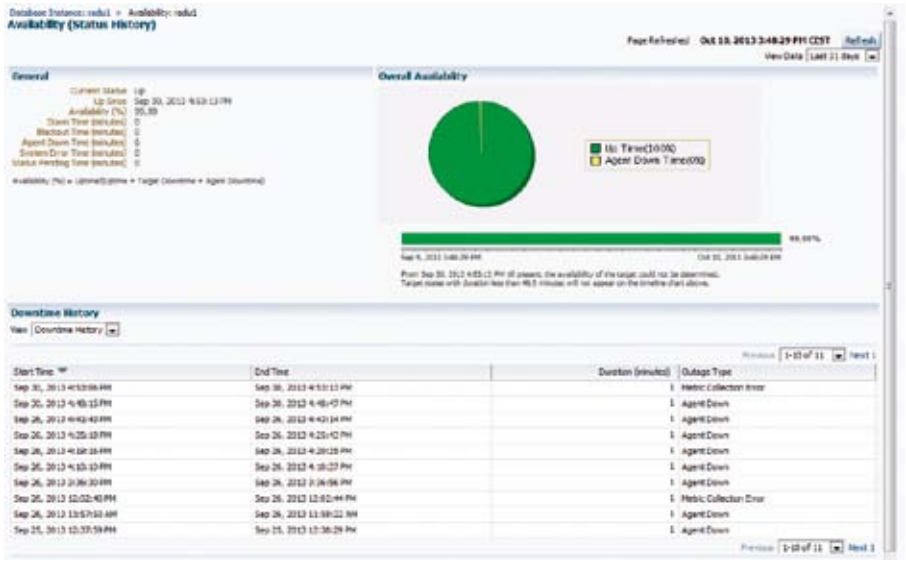


Abbildung 5: Speicherdauer für AWR in Cloud Control einstellen

```
begin
 DBMS_WORKLOAD_REPOSITORY.MODIFY_SNAPSHOT_SETTINGS(
      <Speicherdauer in Minuten>,
      <Erfassungsintervall für Datenbankstatistikdaten in
      Minuten>);
end;
```

Listing 1

Fazit

Das Diagnostics Pack und Oracle Enterprise Manager Cloud Control kombinieren externes und internes Monitoring zu einem Gesamtkonzept und optimieren das Ergebnis durch Ausnutzung der jeweiligen Vorteile.

Ralf Durben
ralf.durben@oracle.com

Monitoring von Oracle-Datenbank-Servern mit Nagios

Gerhard Laußer, ConSol* Software GmbH

Nagios ist das bekannteste und am weitesten verbreitete Open-Source-Monitoring-Tool. Längst dient es nicht mehr nur der Überwachung von Netzwerk-Infrastrukturen und Servern, sondern auch von Web-Applikationen, SAP, Virtualisierung, RZ-Infrastruktur und natürlich Datenbanken. Dank seiner flexiblen Erweiterbarkeit durch Plug-ins sind Nagios im Grunde keine Grenzen gesetzt. Jedes Gerät, jede Software, deren Zustand mithilfe dieser Scripts abgefragt werden kann, lässt sich so ins Monitoring einbinden. In diesem Artikel geht es natürlich um das Thema „Oracle-Datenbanken".

Nagios allein ist lediglich ein Scheduler, der mithilfe anderer Programme die Verfügbarkeit von Servern und Diensten ermittelt, über die Fehlerzustände Buch führt und gegebenenfalls Alarme verschickt. Besagte Programme können beliebige Skripte sein, die ermitteln, ob ein überwachtes System funktioniert oder ausgefallen ist (beziehungsweise im Begriff ist, auszufallen). Mittels definierter Exitcodes („0 = OK", „1 = WARNING", „2 = CRITICAL", „3 = UNKNOWN") wird diese Information an Nagios übergeben.

Zum Check-Resultat gehört auch ein erläuternder Text beziehungsweise eine Fehlermeldung, die in der Nagios-Web-GUI dann grün, gelb oder rot unterlegt ist. Neben der reinen Alarmierung können mit Nagios auch Messwerte, die bei jedem Check anfallen, gespeichert und als Langzeit-Graphen gezeichnet werden. „Dashboards" und „Reporting" sind weitere Themen, für die es Add-ons gibt.

Vor drei Jahren schuf eine Gruppe von Monitoring-Consultants die „Open Monitoring Distribution" (OMD). Das ist eine Sammlung von Nagios und den nützlichsten Tools dazu, zusammengefasst zu einem einzigen Installationspaket.

Nagios-Konfiguration und -Philosophie

In den Konfigurationsdateien von Nagios unterscheidet man mehrere Typen von Objekten. Zunächst gibt es das Host-Objekt, das eine IP-Adresse und einen Namen besitzt. Dies kann ein physikalischer oder virtueller Server sein oder aber auch eine virtuelle Cluster-Adresse. Dem Host zugeordnet sind Service-Objekte. Diese können beliebige Zustände abfragen, etwa ob die Hardware des Servers fehlerfrei funktioniert, wie viel Speicher zur Verfügung steht, binnen wie vieler Sekunden eine Webseite geladen wird und ob im Inhalt ein bestimmtes Schlüsselwort vorkommt, welche Temperatur im Inneren eines Storage-Schranks herrscht etc. Hier sind der Phantasie keine Grenzen gesetzt. Dies ist möglich, weil Nagios die Ermittlung der Zustände an die Plug-ins delegiert.

Bei OMD sind bereits zahlreiche Plug-ins mitgeliefert, unter anderem zur Überwachung von Basis-IT, also von Unix-/Windows-Servern, Festplatten oder Netzwerk-Geräten. Weitere Anforderungen können jedoch leicht selbst implementiert werden. Plug-ins sind nichts weiter als Scripts, die definierte Werte an Nagios zur Weiterverarbeitung übergeben.

Üblicherweise wird man nicht nur den Datenbank-Server überwachen, sondern auch das darunterliegende Betriebssystem. Die gebräuchlichen Checks umfassen den Füllstand der File-Systeme, die Auslastung der Netzwerk-Interfaces, Hauptspeicher, CPU und natürlich den Zustand der Hard-

ware (Temperaturen, Raid, Stromversorgung). Dieser Artikel geht darauf nicht näher ein, da der Schwerpunkt bei der Datenbank liegt. Das Plug-in der Wahl ist hier „check_oracle_health“, das OMD beiliegt.

Bei der Definition eines Service gibt man an, in welchen zeitlichen Abständen das dazugehörige Plug-in aufgerufen werden soll und, falls ein Fehler gemeldet wird, wie oft dieser Check wiederholt werden soll, bevor ein Alarm verschickt wird. Wohin dieser geht, legt Nagios mithilfe zweier weiterer Objekt-Arten fest: „Contacts“ und „Contactgroups“. Dahinter verbergen sich entweder reale Personen beziehungsweise deren Nutzer-Accounts, aber auch anonyme Empfänger von E-Mail oder SMS. Auf diese Art kann Nagios ein Bereitschaftshandy abbilden.

Die Contacts sind Services und Hosts zugeordnet. Damit erreicht man, dass beim Login (Single Sign-on mit AD ist möglich) eines Contact an der Web-Oberfläche dieser ausschließlich die Hosts und Services zu sehen bekommt, die ihm per Konfiguration zugewiesen sind. Außerdem ist damit auch festgelegt, wer bei Ausfall eines Service benachrichtigt wird.

Die Konfiguration von Nagios basiert auf Konfigurationsdateien. Mit der Zeit sind aber auch ein paar Tools entstanden, die das Editieren der Konfiguration im Web-Browser möglich machen. In OMD ist die Web-GUI „Thruk“ dabei, mit deren Config-View bequem Objekte angelegt und gepflegt werden können.

Vorbereitungen

Nach Installation des OMD-Pakets erzeugt der Administrator mit dem Kommando „omd create oramon“ eine sogenannte „Site“. Diese besteht aus einer Laufzeit-Umgebung und dem Benutzer „oramon“, der automatisch zusammen mit einer gleichnamigen Gruppe angelegt wird. Ab hier meldet man sich als User „oramon“ an, um die Site weiter zu konfigurieren. Das Site-Konzept von OMD erlaubt parallele Nagios-Laufzeit-Umgebungen auf einem Monitoring-Server. Das ist sehr praktisch, wenn man neue Versionen erst einmal testen will, bevor man die produktive Site migriert.

```
create user nagios identified by oradbmon;
grant create session to nagios;
grant select any dictionary to nagios;
grant select on V_$SYSSTAT to nagios;
grant select on V_$INSTANCE to nagios;
grant select on V_$LOG to nagios;
grant select on SYS.DBA_DATA_FILES to nagios;
grant select on SYS.DBA_FREE_SPACE to nagios;
```

Listing 1

```
$ check_oracle_health --mode connection-time \
    --connect NAPRAX --user nagios --password oradbmon
OK - 0.11 seconds to connect as NAGIOS |
connection_time=0.1068;1;5
```

Listing 2

Im Standardumfang von OMD ist das Plug-in „check_oracle_health“, das eine ganze Reihe von Prüfungen gegen einen Datenbank-Server ausführen kann, enthalten. Diese reichen vom einfachen Connect-Check über das Ermitteln von Tablespace-Füllständen bis hin zu selbstdefinierten SQL-Statements, die applikationsspezifischen Status liefern. Dazu ist natürlich Oracle-Client-Software auf dem Nagios-Server erforderlich, da die Kommunikation zwischen „check_oracle_health“ und der Oracle-Instanz üblicherweise über ein Netzwerk stattfindet.

Falls es kein einheitliches Installationspaket gibt, reicht auch der „Instant Client“ aus (wichtig ist das „sqlplus“-Paket). Diesen entpackt man einfach im Homeverzeichnis des Users „oramon“ und setzt anschließend die nötigen Environment-Variablen. Technisch wäre es kein Problem, den Instant Client bereits als Bestandteil von OMD mitzuliefern, aus lizenzrechtlichen Gründen ist das jedoch nicht möglich. Wer vorhat, viele Systeme mit OMD auszustatten oder häufig Software-Updates durchzuführen, dem ist das Erstellen eines eigenen OMD-Database-AddOns-RPM empfohlen, in das lizenzierte Clientsoftware gepackt wird. Die Komplettinstallation besteht dann aus zwei RPM-Paketen.

Für die Überwachung eines Datenbank-Servers, die über ein simples Anpingen hinausgeht, ist es natürlich erforderlich, sich dort anzumelden und aus den internen System-Tabellen Informationen auszulesen. Dazu legt man sich am besten einen eigenen User an, der die entsprechenden Rechte erhält (siehe Listing 1). Mit diesen Privilegien ist der Benutzer „nagios“ in der Lage, alle Features von „check_oracle_health“ zu nutzen.

Die Verfügbarkeit überwachen

Als Erstes will man natürlich wissen, ob eine Oracle-Instanz überhaupt läuft. Zu diesem Zweck ruft man „check_oracle_health“ mit dem Kommandozeilen-Parameter „--mode connection-time“ auf. Der Exit-Code richtet sich danach, ob ein Login überhaupt möglich war, und wenn ja, ob der ganze Vorgang länger als eine beziehungsweise fünf Sekunden gedauert hat. Beim Überschreiten dieser Schwellwerte zeigt Nagios einen Warning- beziehungsweise Critical-Zustand an. So einen Basis-Check könnte man auch mit dem Parameter „--mode tnsping“ durchführen, jedoch ist ein echter Login wesentlich aussagekräftiger. Was nützt es, wenn der Listener antwortet, die Instanz aber nicht läuft beziehungsweise sich niemand anmelden kann?

Es wurde bereits beschrieben, wie man einen geeigneten Monitoring-User in der Datenbank anlegt. Beim Aufruf von „check_oracle_health“

```
$ check_oracle_health --mode connection-time \
    --connect 10.73.9.102:1523/orcl12 \
    --user nagios --password oradbmon
$ check_oracle_health --mode connection-time \
    --connect nagios/oradbmon@10.73.9.102:1523/orcl12
```

Listing 3

```
check_oracle_health --mode connected-users
OK - 38 connected users | connected_users=38;50;100
```

Listing 4

```
$ check_oracle_health --mode tablespace-free --name USERS
OK - tbs USERS has 99.98% free space left |
'tbs_users_free_pct'=99.98%;5:;2:
'tbs_users_free'=32761MB;1638.40:;655.36:;0;32767.98
```

Listing 5

muss man jeweils Username, Passwort und Connect String angeben (siehe Listing 2). Hier wurde dem Parameter „—connect“ der TNS-Name mitgegeben. Alternativ kann auch die Schreibweise „Easy Connect“ verwendet werden, wobei Username und Passwort Bestandteil des Connect Strings sein können (siehe Listing 3).

Bereits dieser einfache Check kann mithilfe einer Aufzeichnung und der grafischen Darstellung der gemessenen Antwortzeit eine Aussage über das Verhalten eines Datenbank-Servers im Tagesverlauf machen (siehe Abbildung 1).

Ebenfalls von Interesse kann die Anzahl der eingeloggten User sein (siehe Listing 4). In den Beispielen wurde aus Gründen der Übersichtlichkeit die Parameter „connect“, „user“ und „password“ weggelassen.

Das Beispiel trifft sicher weniger auf Datenbanken zu, die Backends von Web-Applikationen sind. Laufen aber auf zahlreichen PC-Arbeitsplätzen Fat Clients, von denen jeder eine Verbindung zur Datenbank öffnet, ist diese Messung schon sinnvoller.

Tablespaces

Vollgelaufene Tablespaces sollten mit einem Monitoring-System der Vergangenheit angehören. Hier kann man sich frühzeitig alarmieren lassen, wenn der Platz langsam knapp wird. Mit „--mode tablespace-free“ ermittelt „check_oracle_health“, wie viel Prozent noch verfügbar sind (siehe Listing 5).

Sind in diesem Beispiel weniger als fünf beziehungsweise zwei Prozent (Default) des USERS-Tablespace frei, verschickt Nagios einen Alarm an den DBA. Weil bei den heute üblichen Platten und Storage-Systemen wenige Prozent auch große Datenmengen ausmachen, kann man alternativ mit absoluten Werten arbeiten. Die Kommandozeile „--warning 12: --critical 8: --units GB“ würde also ausdrücken: „Warning“, wenn weniger als zwölf, und „Critical“, wenn weniger als acht Gigabyte freier Speicherplatz im Tablespace verfügbar sind. Lässt man den Parameter „--name“ weg, bezieht sich der Check auf sämtliche Tablespaces. Alarmiert wird dann, wenn irgendeiner von ihnen die Schwellwerte unterschreitet.

Diese Vorgehensweise spart einem zwar Tipparbeit bei der Erstellung der Nagios-Konfiguration, wird jedoch nicht empfohlen. Wird der Check etwa wegen Tablespace1 rot, so kann man es leicht übersehen, wenn kurz darauf auch ein Tablespace 2 vollläuft. Ein Statuswechsel (so wie am Anfang von grün nach rot) findet nämlich dann nicht mehr statt.

Eine sinnvolle Ergänzung zur Tablespace-Überwachung ist das Monitoring der Filesysteme oder der Volumes im Storage. Die Alarme bei Unterschreiten des verfügbaren Speichers sind wichtig. Sie sagen dem DBA, dass es Zeit wird, aufzuräumen oder für neuen Platz zu sorgen.

Genauso nützlich ist allerdings auch die Aufzeichnung und Speicherung der gemessenen Werte durch Nagios. Mithilfe des in OMD enthaltenen „AddOns PNP4Nagios“ lässt sich de-

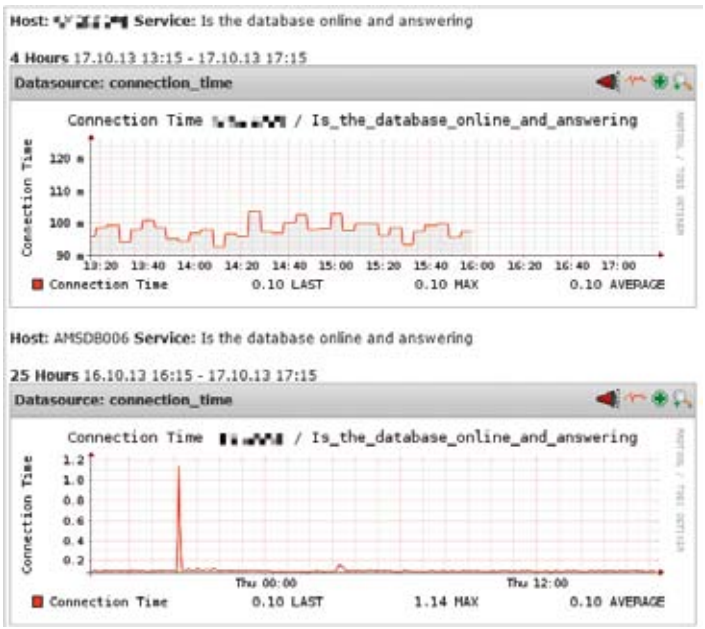


Abbildung 1: Dauer des Logins während der letzten 4 beziehungsweise 25 Stunden

```
$ check_oracle_health --mode switch-interval
OK - Last redo log file switch interval was 12 minutes. Next
interval presumably >418 minutes |
redo_log_file_switch_interval=756s;600:;60:
```

Listing 6

ren Verlauf grafisch anzeigen. So sieht man, wie sich der Platzbedarf langfristig entwickeln wird.

Performance-Monitoring

Zu jedem Monitoring gehört nicht nur die Überwachung auf produktionskritische Ausfälle, sondern auch ein Blick auf die Performance des Systems. In den letzten Versionen von Oracle hat sich ja in Sachen automatischer Optimierung von Speicher und Caches einiges getan.

Das Plug-in „check_oracle_health“ bietet in diesem Zusammenhang „--mode sga-data-buffer-hit-ratio“ an, allerdings dürfte der ermittelte Wert bei einem modernen Oracle stärkeren Schwankungen unterliegen. Er ist daher mit Vorsicht zu genießen. Eine sinnvolle Metrik ist hingegen die Länge der verstrichenen Zeit zwischen zwei Redo-Log-Switches. Folgen diese in kurzen Abständen aufeinander, deutet das auf hohe Schreiblast hin (siehe Listing 6).

Monitoring der Integrität

Der Modus „--mode corrupted-blocks“ warnt, wenn die Hardware-Infrastruktur Probleme bereitet. Innerhalb der Datenbank kann es aber auch zum Verfall kommen. Indexe und Views können beispielsweise ungültig werden, wenn sich die zugrunde liegenden Tabellen ändern. Um so etwas festzustellen, benutzt man „--mode invalid-objects“. Eine feinere Eingrenzung ist noch möglich, wenn der Parameter „--name2“ mit einem der Argumente „dba_objects“, „dba_indexes“, „dba_ind_partitions“ oder „dba_registry“ angehängt wird. Es wird dann nur auf fehlerhafte Objekte einer Kategorie geprüft.

Mit „--name <Ausdruck> --regexp“ kann man noch feiner aussieben. Es ist damit möglich, gezielt nach Objekten eines bestimmten Schemas zu suchen. Beispielsweise sorgt „--name ‚!(SYS|SYSTEM|DBSNMP|....)‘ –regexp“ dafür, dass System-Objekte ignoriert und nur solche überwacht werden, die zu Applikationen gehören.

Alertlog

Neben der Überwachung durch SQL-Befehle ist es auch wichtig, das Logfile von Oracle im Auge zu behalten. Das übernimmt das Plug-in „check_logfiles“. Diesem gibt man eine Liste von Fehler-Pattern mit, die den relevanten Meldungen entsprechen.

In regelmäßigen Abständen sorgt Nagios dafür, dass „check_logfiles“ überprüft, ob eine dieser Meldungen aufgetaucht ist. Stimmt eine Zeile mit einem der Fehler-Pattern überein, wird der DBA alarmiert. Dazu ein Auszug aus der Konfig-Datei für „check_logfiles“ (siehe Listing 7).

Die Trefferzeile aus dem Alertlog wird unverzüglich per E-Mail verschickt. In der Web-Oberfläche ist sie ebenfalls zu sehen, hier wird sie auch gleich rot (oder gelb, falls es nur eine Warnung ist) unterlegt (siehe Abbildung 2).

Auf der Webseite von „check_logfiles“ ist beschrieben, wie die Alert-

```
@searches = ({
  tag => 'oraalerts',
  logfile => '......../alert.log',
  criticalpatterns => [
      'ORA\-0*204[^\d]',        # error in reading control file
      'ORA\-0*206[^\d]',        # error in writing control file
      'ORA\-0*210[^\d]',        # cannot open control file
      'ORA\-0*257[^\d]',        # archiver is stuck
...
      'ORA\-19502[^\d]',        # write error on datafile
      'ORA\-27063[^\d]',        # number of bytes read/written is incorrect
      'ORA\-0*4031[^\d]',       # out of shared memory.
      'No space left on device',
      'Archival Error',
  ],
  warningpatterns => [
      'ORA\-0*3113[^\d]',        # end of file on communication channel
      'ORA\-0*6501[^\d]',        # PL/SQL internal error
      'Archival stopped, error occurred. Will continue retrying',
  ]
});
```

Listing 7

Datei in eine External Table eingebunden werden kann. In Umgebungen, in denen es unmöglich ist, als Nagios-Benutzer Zugriff ins Filesystem des Datenbank-Servers zu erhalten, kann „check_logfiles“ auch wie ein Client arbeiten und die Zeilen per SQL-Statements auslesen.

Applikations-Monitoring

Bisher wurde gezeigt, wie man die Grundfunktionen eines Oracle-Servers überwacht. Damit ist sichergestellt, dass er seinen Dienst erbringen kann. Eine Ebene höher sind Applikationen, die die Datenbank benutzen. Auch hier kann es zu Störungen kommen. Sofern sich diese durch SQL-Abfragen ermitteln lassen, hilft wieder „check_oracle_health“.

Mit dem Modus „sql“ werden beliebige Statements zur Ausführung gebracht. Die Bewertung, ob ein Fehler vorliegt, kann auf mehrere Arten erfolgen. Ist das Ergebnis des Statements eine Zahl, wird diese mit Warning- und Critical-Schwellwerten verglichen. Über- beziehungsweise Unterschreitung zieht einen Nagios-Alarm nach sich (siehe Listing 8).

Ist das Ergebnis ein String, sind die Parameter „--name2“ und „—regexp“ erforderlich. Das Argument des ersteren ist ein regulärer Ausdruck, der mit besagtem String verglichen wird. Bei Übereinstimmung gibt es einen Alarm.

Wenn nicht das Ergebnis eines bestimmten SQL-Statements relevant ist, sondern die Zeit, die zu seiner Ausführung benötigt wird, verwendet man „--mode sql-runtime“. Damit hat man ein mächtiges Werkzeug zur Hand, falls es häufig Beschwerden bezüglich der Antwortzeiten einer Applikation gibt. Man identifiziert die aufwändigsten oder am häufigsten aufgerufenen Datenbank-Anfragen und bindet diese ins Monitoring ein. So erhält man mithilfe von Nagios eine Aufzeichnung der Performance-Indikatoren über den ganzen Tag hinweg. Diese objektiven Zahlen sind eine gute Argumentationshilfe, wenn es „Heute ist es wieder besonders langsam“, heißt, was nicht selten nur aus einem subjektiven Eindruck heraus so empfunden wird.

Dies waren jetzt nur wenige Beispiele für Metriken, die zur Überwachung herangezogen werden können. Auf den Webseiten der Plug-ins finden sich weitere Anregungen. Sinnvolle Werte sind unter anderem „Library Cache Hitratio“, der Prozentsatz der Sort-Operationen, die im Hauptspeicher ausgeführt werden, oder „Soft Parse Ratio“. Mit Nagios, „check_oracle_health“ und „check_logfiles“ (die ja alle in OMD enthalten sind) lässt sich recht schnell ein umfassendes Monitoring für Oracle-Datenbanken aufbauen. Firmenspezifische Abfragen lassen sich ebenfalls leicht implementieren, wie man gesehen hat (siehe Abbildung 3).

Alarmierung

Wenn es um das Versenden von Alarm-Meldungen geht, ist Nagios genauso flexibel konfigurierbar wie bei den Plug-ins für die Checks. Man muss nur einstellen, wer bei welchem Fehler zu welchen Zeiten auf welchem Weg benachrichtigt werden soll. Das Warum kann beispielsweise sein: „Plug-in check_oracle_health hat festgestellt, dass der Datenbank-Server XY nicht antwortet, selbst nach drei Wiederholungen nicht.“ Wer benachrichtigt wird, ergibt sich aus der Zuordnung von Contacts und Contactgroups zu den betroffenen Host- oder Service-Objekten.

Service State Information

Current Status:	CRITICAL (for 0d 0h 2m 7s)
Status Information:	CRITICAL - (5 errors) - ORA-06512: at "SYS.DBMS_STATS", line 23461 ...
	tag alertlog_bba
	ORA-00600: internal error code, arguments: [ktsircinfo_num1], [1], [2], [82139], [], [], [], []
	ORA-06512: at "SYS.DBMS_STATS_INTERNAL", line 67
	ORA-06512: at "SYS.DBMS_STATS_INTERNAL", line 134
	ORA-06512: at "SYS.DBMS_STATS_INTERNAL", line 2468
	ORA-06512: at "SYS.DBMS_STATS", line 23461
Performance Data:	alertlog_lines=6 alertlog_warnings=0 alertlog_criticals=5 alertlog_unknowns=0
Current Attempt:	1/1 (HARD state)
Last Check Time:	02-20-2009 20:04:44
Check Type:	ACTIVE
Check Latency / Duration:	2.146 / 1.612 seconds
Next Scheduled Check:	02-20-2009 20:09:44
Last State Change:	02-20-2009 20:02:49
Last Notification:	N/A (notification 0)
Is This Service Flapping?	YES (26.12% state change)
In Scheduled Downtime?	NO
Last Update:	02-20-2009 20:04:49 (0d 0h 0m 7s ago)

Abbildung 2: Im Alertlog wurden Fehlermeldungen entdeckt

Host	Service	Status	Last Check	Duration	Attempt	Status Information
dbsrv1	app_oracle_common_NAPRAX_check_login	OK	11-14-2008 20:39:58	0d 4h 18m 40s	1/4	OK - 3.39 seconds to connect as NAGIOS
	app_oracle_common_NAPRAX_check_ping	OK	11-14-2008 20:40:30	0d 4h 25m 8s	1/4	OK - connection established to NAPRAX.
	app_oracle_logs_NAPRAX_check_alertlog	OK	11-14-2008 20:41:01	0d 0h 6m 36s	1/1	OK - no errors or warnings
	app_oracle_perf_NAPRAX_check_databuf_hitratio	OK	11-14-2008 20:37:33	0d 0h 15m 4s	1/4	OK - SGA data buffer hit ratio 99.97%
	app_oracle_perf_NAPRAX_check_dictcache_hitratio	OK	11-14-2008 20:42:04	0d 5h 37m 22s	1/4	OK - SGA dictionary cache hit ratio 100.00%
	app_oracle_perf_NAPRAX_check_inmemory_sorts	OK	11-14-2008 20:37:36	0d 4h 23m 2s	1/4	OK - PGA in-memory sort ratio 100.00%
	app_oracle_perf_NAPRAX_check_latches_hitratio	OK	11-14-2008 20:38:08	0d 22h 40m 33s	1/4	OK - SGA latches hit ratio 100.00%
	app_oracle_perf_NAPRAX_check_libcache_hitratio	OK	11-14-2008 20:38:39	0d 1h 3m 59s	1/4	OK - SGA library cache hit ratio 100.00%
	app_oracle_perf_NAPRAX_check_redo_io	OK	11-14-2008 20:39:11	0d 4h 19m 27s	1/4	OK - Redo log io is 0.000228 MB/sec
	app_oracle_perf_NAPRAX_check_softparse	OK	11-14-2008 20:39:42	0d 4h 18m 56s	1/4	OK - Soft parse ratio 100.00%
	app_oracle_perf_NAPRAX_check_switchinterval	OK	11-14-2008 20:40:14	0d 4h 23m 24s	1/4	OK - Last redo log file switch interval was 591 minutes
	app_oracle_tbs_NAPRAX_SYSAUX_check_usage	OK	11-14-2008 20:40:45	0d 4h 22m 53s	1/4	OK - tbs SYSAUX usage is 1.90%
	app_oracle_tbs_NAPRAX_SYSTEM_check_usage	OK	11-14-2008 20:41:17	0d 5h 37m 22s	1/4	OK - tbs SYSTEM usage is 2.12%
	app_oracle_tbs_NAPRAX_TEMP_check_usage	OK	11-14-2008 20:41:49	0d 5h 37m 21s	1/4	OK - tbs TEMP usage is 0.00%
	app_oracle_tbs_NAPRAX_TEST_TBS_check_usage	OK	11-14-2008 20:42:20	0d 5h 37m 22s	1/4	OK - tbs TEST_TBS usage is 19.50%
	app_oracle_tbs_NAPRAX_UNDOTBS1_check_usage	OK	11-14-2008 20:37:52	0d 4h 22m 46s	1/4	OK - tbs UNDOTBS1 usage is 0.03%
	app_oracle_tbs_NAPRAX_USERS_check_usage	OK	11-14-2008 20:38:23	0d 6h 32m 42s	1/4	OK - tbs USERS usage is 0.00%
	app_ticker_check_noticks	OK	11-14-2008 20:40:55	0d 0h 1m 42s	1/4	OK - noticks: 26s

Abbildung 3: Umfangreiches Monitoring eines Datenbank-Servers

```
$ check_oracle_health … --mode sql \
    --warning 8 -critical 15 \
    --name “select count(*) from app1.processes where status =
’failed‘ “ --name2 failedprocs
CRITICAL - failedprocs: 29 | failedprocs=29;8;15
```

Listing 8

Üblicherweise gibt es einen Contact namens „db-admins“, der allen datenbankrelevanten Services zugeordnet ist. Dessen Mailadresse ist eine Sammeladresse für das Datenbank-Team.

Ebenso konfigurierbar ist das Wann, also zu welchen Uhrzeiten überhaupt Alarme verschickt werden. Die angesprochene Flexibilität zeigt sich aber am deutlichsten beim Wie. Analog zu den Checks, die von beliebigen Skripten (den Plug-ins) ausgeführt werden, kommen auch auf der Alarmierungsseite Notification-Scripts zum Einsatz. Nagios kümmert sich lediglich um den Aufruf der Skripte mit passenden Kommandozeilen-Parametern. Was diese Scripts letztlich tun, bleibt dem Administrator überlassen.

Natürlich ist bei OMD ein Skript enthalten, das für den Versand von E-Mails sorgt. Damit dürften für die meisten Installationen bereits die Anforderungen abgedeckt sein. Wünscht man jedoch den Versand einer SMS oder das automatische Öffnen eines Tickets in einem entsprechenden Tool, kann dies durch geeignete Notification-Skripte geschehen. Im Internet gibt es zahlreiche solcher Programme, die man herunterladen und in sein Monitoring einbinden kann.

Um das Thema anschaulicher zu beschreiben, folgendes Szenario: Bei jeder Störung einer produktiven Datenbank sollen von 8 bis 18 Uhr die DBAs eine E-Mail bekommen. Ab 18 Uhr soll eine SMS an das Bereitschaftshandy geschickt werden. Bleibt ein Fehlerzustand länger als zwei Stunden unbearbeitet, soll auch der Teamleiter eine SMS bekommen (sogenannte „Notification Escalation“). Parallel dazu ist bei jedem Incident auch automatisch ein Ticket anzulegen. Störungen von Testsystemen sollen lediglich per E-Mail gemeldet werden. Bei Ende des Ausfalls sollen alle wieder eine „OK“-Mail erhalten und das zugehörige Ticket automatisch geschlossen werden.

Solche Abläufe sind bei Nagios ziemlich einfach zu konfigurieren. Man kann sogar noch einen Schritt weiter gehen. Als dritte Kategorie von Skripten, die aus Nagios heraus aufgerufen werden können, gibt es die Event-Handler. Damit kann man beispielsweise ein ausgefallenes System durchstarten lassen. Gelingt es auf diese Art, vor dem dritten Check wieder einen Normalzustand herzustellen, so wird kein Alarm ausgelöst. Den Bereitschafts-DBA wird es freuen. Dass etwas passiert ist, bleibt trotzdem nicht verborgen. In den Logs von Nagios wird ja jedes Ereignis protokolliert.

Reporting

Störungen und Ausfälle frühzeitig mitzubekommen, um größeren Schaden abwenden zu können, ist eine feine Sache. Die Vorgabe wird aber meistens lauten, dass es erst gar nicht zu solchen Situationen kommen soll. Am Ende des Monats will man daher wissen, ob die Systeme fehlerfrei durchgelaufen sind – und falls nicht, wie oft und wie lange nicht. Die Weboberfläche „Thruk“, Bestandteil von OMD, bietet zu diesem Zweck ein mächtiges Reporting-Tool an. Out-of-the-box lassen sich damit Verfügbarkeitsberichte erstellen sowie die Auskunft über die Erreichbarkeit einzelner Hosts, ganzer Host-Gruppen oder dedizierter Services geben (siehe Abbildung 3).

Ein Sonderfall ist SLA-Reporting. Hier gibt man beim erstmaligen Einrichten des Reports die vereinbarte Verfügbarkeit an und erhält dann im

Monatsbericht noch zusätzlich eine Liste der SLA-Verletzungen. Mit wenig Aufwand kann man Farben und Seitengestaltung so anpassen, dass das Aussehen der Reports dem firmentypischen Design entspricht.

Einmal angelegt kann man jederzeit einen aktuellen Report in der Thruk-GUI erzeugen und sich anzeigen lassen. Praktischer ist es allerdings, dies dem System zu überlassen. In Thruk muss man dazu nur einen Empfängerkreis und den Zeitpunkt des Versands angeben, dann werden die Reports automatisch beispielsweise zum Monatsende erstellt und per Mail als PDF verschickt.

Fazit

Nagios ist als lizenzfreies Monitoring-System so flexibel konfigurierbar, dass die meisten Anforderungen out-of-the-box umsetzbar sind. Fehlende Funktionalität kann mit wenig Aufwand dank der offenen Schnittstellen selbst implementiert werden. Für viele Sonderwünsche gibt es bereits existierende Add-ons, die man sich herunterladen kann. Das Budget, das ansonsten in Form von Lizenzgebühren verbraucht würde, investiert man besser in eigene Erweiterungen.

In Form von OMD steht ein Paket zur Verfügung, das den initialen Installationsaufwand zu einer Sache von Minuten werden lässt. Sicherlich bieten ein Enterprise Manager oder vergleichbare Kauftools dem DBA die Möglichkeit, seine Datenbanken auch bedienen zu können. Gleiches gilt für SAP, HP Insight Manager etc. Es empfiehlt sich jedoch, diese Programme ausschließlich als Management-Tools für die jeweiligen Spezialisten zu nutzen und die entsprechende Monitoring-Komponente an Nagios abzugeben oder zumindest zu koppeln. Auf diese Weise erhält man ein System (das dazugehörige Modewort lautet „Umbrella-Monitoring"), durch das sämtliche Alarmierungen laufen und das einen einheitlichen Blick auf alle IT-Objekte eines Unternehmens bietet.

Weiterführende Informationen

- http://www.nagios.org
- http://omdistro.org
- http://labs.consol.de/nagios/check_oracle_health
- http://labs.consol.de/nagios/check_logfiles
- http://www.thruk.org

Gerhard Laußer
gerhard.lausser@consol.de

Oracle-Datenbank-Security-Monitoring

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

Wann haben Sie die Sicherheit Ihrer Oracle-Datenbank das letzte Mal überprüft? Kennen Sie die Bedrohungen und die davon abgeleiteten Risiken? Vielleicht sagen Sie jetzt: „Meine Risiken sind unter Kontrolle. Ich kenne den notwendigen Schutzbedarf und habe diesen in meiner Datenbank bedacht." Für alle anderen fasst dieser Artikel ein paar Fakten darüber zusammen, was passiert, wenn Sie kein Security-Monitoring aktiviert haben. Es geht um das Wissen von Bedrohungen und Risiken und die damit verbundene und notwendige Transparenz und Kontrolle. Die Lösung dahinter beschreibe ich als Security-Monitoring.

Natürlich bietet Oracle verschiedene Möglichkeiten für ein Security-Monitoring out-of-the-box an. Hierunter fallen folgende Lösungen:

- *Oracle Database Audit*
  Standard-Funktionen in der Datenbank, die Aktivitäten innerhalb der Datenbank protokollieren. In der neuen Datenbank-Version 12*c* heißt dieses Audit „Unified Auditing".
- *Oracle Database Vault*
  Lösung, um zusätzliche Sicherheitszonen in der Datenbank einzuziehen, die Regeln in Form von Policies erzwingen und den Gebrauch und Missbrauch dieser Policies automatisch protokollieren.
- *Oracle Audit Vault & Database Firewall*
  Ein zentraler Protokoll-Server, der alle Aktivitäten, die in der Datenbank stattfinden, revisionssicher protokolliert und gegebenenfalls auch blockieren kann, ebenso wie Kommandos, bevor sie die Datenbank erreichen.
- *Oracle Compliance Dashboard*
  Eine Funktionalität innerhalb des Enterprise Manager Cloud Control 12*c*, die man mit dem „Lifecycle Management Pack" erwirbt. Damit können entsprechende Security-Policies überwacht und auf Einhaltung überprüft werden. In der Regel sind das Konfigurations-Einstellungen.

Dieser Artikel zeigt, was passiert, wenn man überhaupt kein Security-Monitoring betreibt. Dafür wurde eine kleine Auswahl sogenannter „DB Security Top Issues" zusammengestellt, die entstehen, wenn der Blick auf die Security fehlt. Der Autor hat in der Vergangenheit viele sogenannte „Datenbank-Security-Reviews" durchgeführt, also ein manuelles Security-Monitoring für eine ausgewählte Datenbank. Diese zeigen den Sicherheitszustand einer Datenbank auf und betrachten hierbei folgende Bereiche:

- Konfiguration der Datenbank entsprechend des Schutzbedarfs
- Monitoring- und Audit-Einstellungen innerhalb und außerhalb der Datenbank
- Einstellungen und Konzept-Implementierungen hinsichtlich Verfügbarkeit
- Die eingestellte Zugriffskontrolle, auch für Daten, die die Datenbank verlassen
- Maßnahmen für die Compliance-Einhaltung sowie Nachhaltigkeit

Aus vielen Untersuchungen nachfolgend eine Auswahl der „DB Security Top Issues".

Top Issue 1: Kaum Wissen und Verantwortungen definiert

Die Erfahrung aus vielen manuellen Security-Monitorings zeigt, dass selten das Wissen zu dem notwendigen Schutzbedarf vorhanden ist. Es werden weder Daten sicherheitstechnisch klassifiziert noch sind die teilweise zwingenden und unternehmensabhängigen Anforderungen auch aus den Gesetzen wie Bundesdatenschutzgesetz, Sozialgesetzbuch, SOX, PCI DSS etc. im Detail bekannt. Somit besteht selten ein wirkliches Wissen über den notwendigen Schutzbedarf bei den Personen, die für den Schutz sorgen müssen.

Maßnahme: Es sind organisatorische Maßnahmen zu treffen, die dafür sorgen, dass der Sicherheitszustand einer Datenbank in der Verantwortung einer definierten Business-Rolle liegt. Der Schutzbedarf für die klassifizierten Daten muss bekannt sein, Maßnahmen müssen umgesetzt werden.

Top Issue 2: Selten eigene Mindestsicherheit implementiert

Eine unternehmensweite Mindest-Sicherheit aller Datenbanken ist sehr selten implementiert, obwohl Oracle hier Vorgaben und Anregungen definiert, wie im „Oracle Database Security Guide 11*g*, Chapter 10, Keeping your database secure" beschrieben. Wenn von Mindestsicherheit die Rede ist, dann sind damit insbesondere die Maßnahmen gemeint, die jede Datenbank aktiviert haben muss. Man könnte auch sagen, dass ein IT-Grundschutz entsprechend des Bundesamts für Sicherheit in der Informationstechnik (BSI) einheitlich umgesetzt ist. Natürlich bietet eine Oracle-Datenbank viele Funktionen, die bei Aktivierung eine Sicherheit erhöhen können. Diese gilt es einzuschalten.

Maßnahme: Das Kapitel 10 im Database Security Guide (für die Datenbank 12*c* steht das Kapitel im Anhang) anwenden und darauf eigene Anpassungen vornehmen, sodass ein einheitlicher Unternehmensstandard für die Mindestsicherheit besteht. Es existiert außerdem in der My Oracle Support Knowledge Base ein Artikel: „10

Basic Steps to Make your DB secure from Attacks (ID 1545816.1)“. Dieser Unternehmensstandard kann dann mit entsprechenden Tools auf Einhaltung geprüft werden.

Top Issue 3: Schwache Zugriffskontrolle und Zwecktrennung

In den Bereichen „Zugriffskontrolle“ und „Zwecktrennung“ gibt es verschiedene Konzepte, die teilweise sehr veraltet oder zu komplex sind. Andere ergeben durchaus Sinn, werden jedoch durch andere Konzepte ausgehebelt. Im Bereich der Zugriffskonzepte sind die größten Fettnäpfchen:

- Nutzung von Standard-Rollen wie „CONNECT“ oder „RESOURCE“ für Nicht-Admin-User. Die „RESOURCE“-Rolle vergibt Rechte für die Anlage von Objekten und „UNLIMITED TABLESPACE QUOTA“ (wurde ab Version 12*c* aufgehoben). Wenn man Standard-Rollen an Datenbank-Benutzer vergibt, muss man die Berechtigungen in den Rollen kennen und entscheiden, ob diese wirklich notwendig sind. Oft wird die „RESOURCE“-Rolle vergeben, obwohl diese Benutzer gar keine eigenen Objekte besitzen. Wofür benötigt der Datenbank-Benutzer dann die „RESOURCE“-Rolle?
- Nutzung keiner Rollen, sondern ausschließliche und explizite Vergabe von Privilegien an den Benutzer. Das erhöht die Komplexität in der Zugriffskontrolle, da für weitere Benutzer viele „GRANTS“ vergeben werden müssen anstelle eines Rollen-„GRANT“.
- Sehr komplexe Zugriffskonzepte mit vielen Tausend „GRANTS“, doppelten Berechtigungen und redundanten Privilegien. Diese komplexen Zugriffskonzepte sind schwer zu kontrollieren. Eine Vereinfachung ist sofort sichtbar, wenn man das Berechtigungskonzept genauer betrachtet.
- Aushebelung der Zugriffskontrolle durch Vergabe von Privilegien an „PUBLIC“ sowie Erstellung von „PUBLIC Database“-Links auf die gleiche Datenbank mit Zugriff auf die Anwendung.
- Zu mächtige „GRANTS“ an Anwendungs-Owner mit „ANY“-Privilegien, die auch gleichzeitig als Anwendungs-User genutzt werden. Diese Benutzer werden von der Anwendung genutzt, die dann die Autorisierung vornimmt, aber auch von Endbenutzern, die mit Tools wie SQL-Developer auf den Daten operieren. Diese Endbenutzer können dann alle mächtigen Privilegien anwenden, da die Autorisierung in der eigentlichen Anwendung nicht mehr stattfindet. Zusammenfassend:

- Kein Least-Privilege-Konzept implementiert.
- Massive Nutzung von Datenbank-Links, die unsicher konfiguriert sind.
- Anlage einer Password-Datei mit vielen SYSDBAs; eine Remote-Administration der Datenbank wird aber nicht ausgeübt und eine Standby-Umgebung existiert nicht.
- Autorisierung übernimmt die Anwendung und die Datenbank liefert ihr mächtige Zugriffsrechte.

In der Zwecktrennung fordern die Gesetze eine Personalisierung der DBAs, sodass Admins tatsächlich eindeutig identifizierbar sein sollten. Stattdessen werden „SYS“ und „SYSTEM“ genutzt. Diese Standardbenutzer werden dann als sogenannte „Shared Accounts“ genutzt und das von vielen DBAs. Auch werden die Privilegien für das Account-Management vermischt, also die Erstellung von Usern und Rollen, sodass viele Personen, auch über Shared-Accounts, für das Account-Management zuständig sind. Die Datenbank 12*c* führt eine gewisse Zwecktrennung für DBAs ein, ein sogenanntes „DBA Segration of Duty (DBA SoD)“.

Maßnahme: Einheitliche Konzepte müssen durchgängig implementiert sein. Die Zugriffskontrolle sollte regelmäßig nach dem „least“-Privileg überprüft werden. Hierfür kann man mit der Datenbank-Version 12*c* die Funktionalität „Privilege Analysis“ verwenden und somit feststellen, welche Privilegien tatsächlich gebraucht werden. Bei der Zwecktrennung gilt, entsprechend dem „Top Issue 1“,Verantwortlichkeiten zu schaffen und eigene Konzepte in der Datenbank zu erzwingen. Man kann in der Datenbank auch eine DBA-Zwecktrennung implementieren. Zu hinterfragen gilt es Konzepte, die unnötig die Zugriffskontrolle verkomplizieren, wie etwa viele Schema-Kopien mit gleichen Tabellen-Strukturen – „Keep it simple“. Wichtig ist, die notwendige Transparenz zu schaffen, damit die Kontrolle nicht verloren geht. Auch eine regelmäßige Kontrolle und Attestierung der Zugriffskontrolle ist notwendig. Hierfür sind solche Prozesse geeignet, die aus dem Identity-Management-Umfeld kommen.

Top Issue 4: Erhöhte Komplexität und Anwendung falscher Konzepte

Komplexe Konzepte in der Datenbank können dazu führen, dass die Sicherheit unnötig leidet. Lieblings-Beispiel des Autors ist ein Konzept zur Erhöhung der Performance, wenn es um sehr viele Daten-Zeilen in einer Tabelle geht. Dafür werden veraltete Daten aus einer Tabelle „A“ in einem Schema „A“ in eine gleiche Tabelle „AA“ eines neuen Schemas „AA“ verschoben. Die Anwendung operiert auf Schema „A“, also auf den aktuellen Daten. Die Trennung der Daten soll dazu führen, dass die Abfragen auf Objekte im Schema „A“ schneller durchlaufen, da nicht mehr so viele Daten gelesen werden müssen. Dieses komplexe Konzept schafft aber Redundanzen in den Objekten (es existieren gleiche Objekte in unterschiedlichen Schemata) und natürlich Redundanzen in der Zugriffskontrolle, denn das gleiche Zugriffskonzept für Objekte im Schema „A“ ist nun auch auf das Schema „AA“ abzubilden.

An diesem Beispiel sollte man erkennen, dass die Zugriffskontrolle so komplex werden kann, dass man irgendwann die Übersicht verliert und damit auch die Kontrolle. Je mehr Schemata man aufbaut, desto komplexer wird das ganze Konstrukt. Abhilfe würde hier das Information-Lifecycle-Management-Konzept von Oracle helfen, eine transparente Lösung, die auf Datenbank-Objekte wie Tabellen abgebildet wird und das gleiche Ergebnis

hat, nämlich Performance, ohne die Komplexität zu erhöhen (siehe http://www.oracle.com/technetwork/database/enterprise-edition/index-090321.html).

Themen wie Datenschutz außerhalb der Datenbank sind auch selten im Fokus. So werden Exports und Backups von Daten durchgeführt, ohne den Datenschutz zu betrachten. Sobald die Daten die Datenbank verlassen, sind sie nicht mehr geschützt. Wenn doch, werden teilweise komplexe organisatorische Konzepte angewendet, die letztendlich aber nicht ausreichend für einen geeigneten Datenschutz sind. Obwohl man mit sehr einfachen Mitteln jeden Dump und jedes Backup mit Datenbankmitteln durch Verschlüsselung schützen kann.

Maßnahme: Die Sicherheitskonzepte der Datenbank kennen, die man transparent anwenden kann, ohne die Komplexität zu erhöhen und ohne die Anwendung in der Arbeit zu behindern. Diese Konzepte nützen auch oft der Sicherheit in der Datenbank.

Top Issue 5: Verfügbarkeit entsprechend Anforderungen?

In Bezug auf die Maßnahmen, die die Verfügbarkeit erhöhen, sind die Datenbank-Betreiber meist gut aufgestellt. Auffallend ist aber, dass Vermutungen angestellt werden. Es werden also zum einen Konzepte benannt, die so nicht umgesetzt sind, wie eine Retention Policy von zehn Tagen im Backup, oder es werden massive Anstrengungen implementiert, ohne genau die Anforderungen an die Verfügbarkeit zu kennen.

Hier sind wir wieder beim „Top Issue 1“: Man muss den Schutzbedarf, also auch die Verfügbarkeitsanforderungen, kennen und entsprechende Maßnahmen implementieren. Wenn Kunden für einen 24x7-Betrieb sehr gute Lösungen implementiert haben wie Dataguard, RAC und ein gutes „Backup&Recovery“-Konzept, dann meistens zum Schutz vor ungeplanten Ausfallzeiten. Ein Schutz vor geplanter Ausfallzeit wird, obwohl die notwendigen Lösungen dafür bereits aktiviert sind, nicht praktiziert. Hier fehlt erneut das Wissen um Konzepte wie dem Transient Logical Standby für Rolling Upgrades (Oracle-HA-Konzepte siehe http://www.oracle.com/technetwork/database/features/availability/twp-dataguard-11gr2-1-131981.pdf, Oracle Patch Assurance – Data Guard Standby-First Patch Apply, Rolling Upgrades (Doc ID 1265700.1), siehe http://www.oracle.com/technetwork/database/features/availability/maa-wp-11g-upgrades-made-easy-131972.pdf und http://www.oracle.com/technetwork/database/features/availability/maa-wp-11g-transientlogicalrollingu-1-131927.pdf).

Maßnahme: Genau die Anforderungen mit entsprechenden Maßnahmen erfüllen und das Wissen über die Anforderungen und über die implementierten Konzepte haben.

Top Issue 6: Schwaches oder gar kein Auditing implementiert

Hier geht es um die Überwachung der Datenbank mittels Audit. Die Gesetzeslage ist eindeutig: Ohne eine geeignete Protokollierung ist die Erfüllung von §9 des Bundesdatenschutzgesetzes nicht möglich. Trotzdem wird das Audit der Datenbank nicht immer aktiviert oder die Aktivierung nur halbherzig durchgeführt. Wichtige SYS-Objekte („sys.user$“ etc.), wichtige Anwendungsobjekte, wichtige und gefährliche Privilegien sowie „SYSDBA“-Aktivitäten sollten immer protokolliert werden. Ausreden, dass mit einer Audit-Protokollierung Unmengen an Daten entstehen, sind falsch. Man muss das Audit nur richtig konfigurieren. Hierbei ist es wichtig zu erkennen, wann gegen die eigenen Konzepte verstoßen wird, und nur dann diese Verstöße zu protokollieren. Dazu ein Beispiel: Bei Zehntausenden von Zugriffen pro Tag auf die Datenbank sollte man nicht „CREATE SESSION“ protokollieren, sondern „CREATE SESSION WHENEVER NOT SUCCESSFUL“, damit zumindest mögliche Brute-Force-Attacken erkannt werden.

Maßnahme: Ein sinnvolles Audit-Konzept aufsetzen. Es ist wichtig, Verstöße gegen eigene Policies (unerlaubte Zugriffe) zu erkennen und nicht alle Aktivitäten zu protokollieren.

Fazit

Wer den Sicherheitszustand nicht kennt, wird auch keine Maßnahmen ergreifen. Daher ist es wichtig, regelmäßig den Sicherheitszustand zu überprüfen. Wie man das macht, welche Erkenntnisse man aus den Einstellungen der Datenbank gewinnen kann und welche Best Practices existieren, beschreibt der Autor ausführlich in seinem neuen Buch „Oracle Security in der Praxis. Vollständige Sicherheitsüberprüfung Ihrer Oracle Datenbank“.

Die Sicherheit der Datenbank ist ernst zu nehmen. Dabei sollte man den wahren Zustand der Datenbank kennenlernen. Wissen über reale Zustände und Wissen über geeignete Konzepte schützt. Erst dann kann man entscheiden, welche Maßnahmen tatsächlich notwendig sind.

Carsten Mützlitz
carsten.muetzlitz@oracle.com

Vorschau

Schwerpunkt-Thema der nächsten Ausgabe ist

„Big Data“

Wir berichten über

- Datenquellen (Hadoop Distributed File System, NoSQL
- Analyse, Data Scientist
- M2M-Kommunikation
- Praktische Erfahrungen

Sie erscheint am
14. Februar 2014

2 Ausgaben gratis

Dynamische IT für Unternehmen

Anzeige
WIN-Verlag

Immer die Nase vorne

mit einem persönlichen Abonnement

www.digital-business-magazin.de/abo

WIN VERLAG

www.digitalbusiness-cloud.de

Überwachen Sie schon oder konfigurieren Sie noch?

Peter Bekiesch, Herrmann & Lenz Solutions GmbH

Monitoring ist als Basis-Disziplin im Rechenzentrums- und Datenbank-Betrieb unerlässlich. Dabei tauchen vielen Fragen auf: „Wie setze ich nun ein adäquates und maßgeschneidertes Monitoring auf und wie betreibe ich es?", „Kaufe ich ein Produkt, baue ich etwas selbst oder soll ich mich doch in der Open-Source-Welt umschauen?", „Überwache ich meine gesamte IT oder nur Teile davon?" oder „Wie viel Zeit sollte ich für das Projekt veranschlagen?" Der Artikel liefert entsprechende Antworten nach dem Motto „Organisation ist nicht alles, aber ohne Organisation ist alles nichts."

Bevor man sich auf die Suche nach einem geeigneten Werkzeug macht, ob nun gekauft oder nicht, müssen zunächst einige Hausaufgaben gemacht werden. Ziel ist es, die Rahmenbedingungen und Anforderungen im Laufe des Projekts nicht ständig neu zu definieren, sondern bereits zu Beginn wichtige Eckdaten zu setzen und Entscheidungen zu treffen.

Was möchte ich überwachen?

Im ersten Schritt erstellt man eine Liste von Maschinen und Komponenten, die ins Monitoring fließen sollen. An dieser Stelle macht man sich schon vorab die Mühe und unterteilt diese Komponenten nach Produktiv-, Entwicklungs- und Testsystem. Die Einteilung kann selbstverständlich bei jedem abweichen. Es sollte unbedingt vermieden werden, bereits zu Beginn in einem Organisationschaos zu versinken. Klare Strukturen machen auch später das Leben viel einfacher.

Nachdem nun die Liste der Komponenten festgelegt wurde, stellt sich die Frage, was man möchte – oder besser, was man auf diesen Komponenten überwachen muss. Bereits frühere Artikel und einschlägige Literatur listen eine breite Palette von Messpunkten auf, deren Überwachung sinnvoll erscheint. Bei dem Großteil ist das auch der Fall. Niemand stellt in Frage, dass etwa Plattenplatz- und Hauptspeicher-Verbrauch oder die allgemeine Netzwerk-Erreichbarkeit zu den zwingend notwendigen Messpunkten gehören. Doch auch hier ist die Konzentration auf das Wesentliche wichtig. Weniger ist manchmal mehr.

Wer soll wann und wie benachrichtigt werden?

Neben dem Überwachen ist die zweite wichtigste Aufgabe des Monitorings, die Probleme zu melden. Hier kommen nun wieder Menschen ins Spiel. Man definiert sein Team, das später die Monitoring-Plattform bedienen und/oder betreiben soll. In diese Zusammenstellung muss unbedingt einfließen, ob im Schichtbetrieb gearbeitet wird oder eine Rufbereitschaft gewährleistet sein muss. Betreibt man seine Produktiv-Systeme 7x24? Muss im Notfall jemand geweckt werden? Reichen Benachrichtigungen per E-Mail aus oder sind hier andere Mittel notwendig?

Aus diesen Fragestellungen ergibt sich ein Benachrichtigungsprofil, das später in die Konfiguration einfließen wird. Es sollte auch nicht vergessen und unterschätzt werden, dass Rufbereitschaft oder nächtliche Benachrichtigungen des Monitorings (etwa per SMS oder Anruf) arbeitsrechtliche Themen berühren.

Welche Informationen soll das Monitoring zusätzlich liefern?

Dies ist eine sehr häufig unterschätzte Frage. Das Monitoring ist nicht einfach nur dazu da, einige Maschinen „anzupingen" oder mal den verfügbaren Plattenplatz zu prüfen. Ein Monitoring-System kann sehr viel tiefer gehende Informationen über die IT liefern. Der obligatorische SLA-Report sowie eine Statistik über die Verfügbarkeit der Komponenten gehören zu den ersten und wichtigsten Auswertungen.

Ein Monitoring sollte aber noch viel mehr liefern können, wenn es die notwendigen Voraussetzungen erfüllt. Dann ist der steigende Verbrauch des Plattenplatzes ebenso (grafisch) auswertbar wie die Auslastung eines Prozessors oder die Paket-Verlustrate auf dem Netzwerk. Klar sollte sein: Überall dort, wo Messpunkte überwacht werden, fallen Daten an. Diese Daten sollten gespeichert und auswertbar sein.

Die Historie, der Blick in die Vergangenheit, ist für die Gegenwart und die Zukunft ein kostbarer Schatz an nützlichen Informationen. Diese intelligent ausgewertet und im besten Fall miteinander korreliert, ergeben Hinweise auf Schwachstellen und möglicherweise anstehende Probleme. Das Ziel eines Monitorings sollte es nicht nur sein, aktuelle Probleme zu erkennen, sondern auch, neue zu verhindern, bevor sie entstehen.

Ein Monitoring-System kaufen oder Open Source einsetzen?

Die letzte Frage in diesem Kontext befasst sich nun mit dem Werkzeug selbst. Es sollen hier nicht die Vor- und Nachteile kommerzieller Werkzeuge oder der auf dem Open-Source-Markt verfügbaren Werkzeuge aufgezählt werden. Man sollte vielmehr das Thema aus mehreren Blickwinkeln betrachten. Es gibt viele gute Gründe für kommerzielle Software, für Open Source und natürlich auch für selbst erstell-

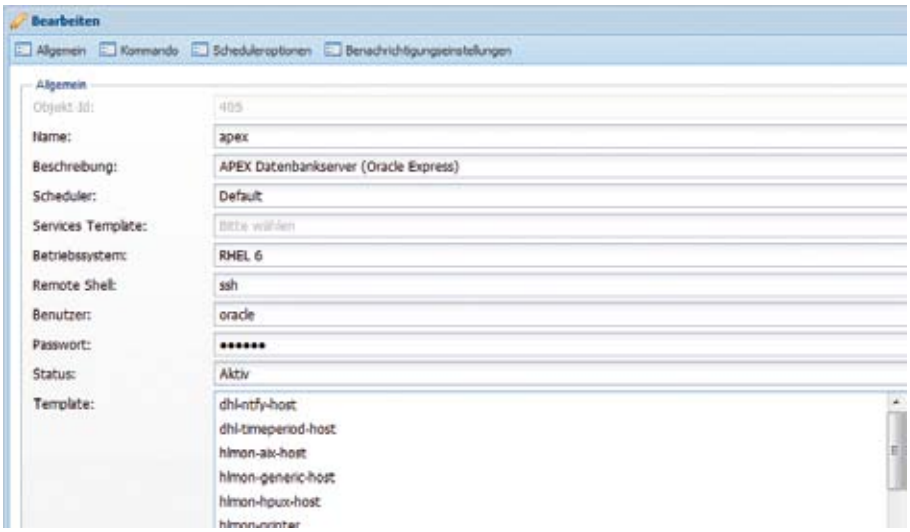


Abbildung 1: Das Aufsetzen des Systems

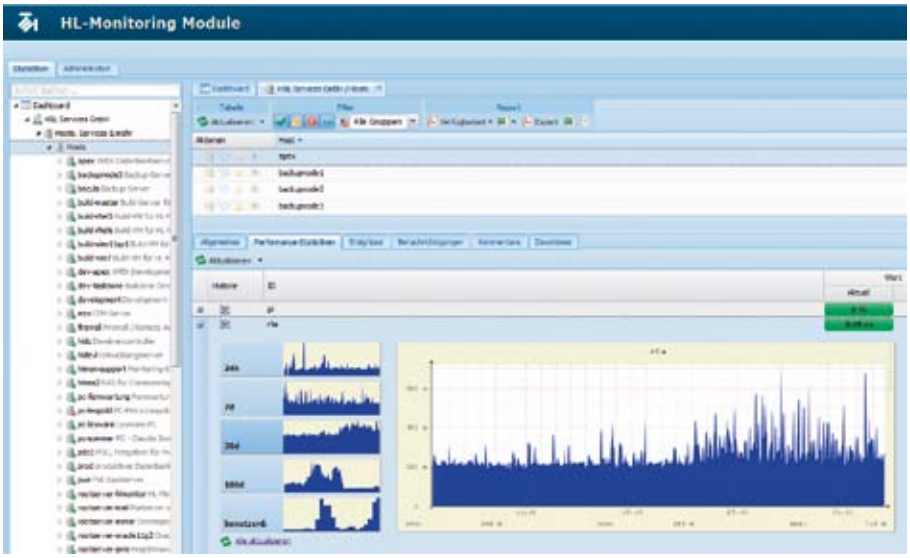


Abbildung 2: Die Messpunkte

te Komponenten. Muss man sich für die eine oder die andere Seite entscheiden? Nein! Der Autor verfährt nach dem Prinzip: „Von allem das Richtige".

Nagios/Icinga ist wohl das bekannteste Open-Source-Monitoring-Werkzeug auf dem Markt. Es hat inzwischen einen Status erreicht, der es beinahe zum Standard macht. Die simple Struktur von Hosts und Services ist leicht verständlich und einfach umzusetzen. Um das riesige Nagios/Icinga-Universum scharen sich inzwischen unzählige Plug-in-Hersteller, die Komponenten für Nagios/Icinga anbieten, um mit deren Hilfe nicht nur IT-Komponenten zu überwachen.

Was insgesamt in dieser Betrachtung fehlt, sind unter anderem folgende Fragen: „Wie viele Ressourcen benötigt man, um das Monitoring aufzusetzen und zu betreiben?", „Wie viel Aufwand muss man in die Konfiguration stecken?", Welche Plug-ins setzt man am besten ein?", Funktionieren die frei verfügbaren Plug-ins fehlerfrei?", „Wie viele Ressourcen sind notwendig, um das System auf einem aktuellen Stand zu halten?", „Was unternimmt man, wenn das Monitoring-System selbst ein Problem hat?" oder „Wer hilft einem?"

Die Grundannahme wäre jetzt, dass ein kommerzielles Produkt nun auf alle diese Fragen eine passende Antwort bietet. Dem ist aber meistens nicht so. Auch hier finden sich häufig versteckte Zeitfresser und Kosten. „End-to-End-Monitoring", in der Praxis, „Eierlegende Wollmilchsau" genannt, hört sich gut an und ist sicherlich ein sehr anspruchsvolles und interessantes Unterfangen. Braucht man das wirklich oder kann man sich gegebenenfalls dorthin entwickeln? Ein umfangreiches Rechte-System – wer darf was im Monitoring sehen, welche Maschine, welchen Messpunkt, welchen Report – braucht man das wirklich? Reicht nicht eine einfache Unterscheidung nach Schreib- und Lese-Rechten und bei Bedarf auch die Aufteilung in unterschiedliche Mandanten aus? Nicht selten befasst man sich allein mit diesem Thema mehrere Wochen, ehe auch nur eine Maschine überwacht wird. Der Autor verfolgt die Philosophie „Transparenz für alle". Warum soll der Datenbank-Administrator nicht wissen und sehen, dass es auf dem Server oder der Netzwerkleitung Störungen gibt? So ließe sich häufig die Standardaussage von Anwendern, „Die Datenbank läuft nicht", schon frühzeitig entkräften.

Dies sollte als Fragenkatalog am Anfang reichen, um die wichtigsten Eckpunkte zu definieren. Man sollte nur nicht dem Zwang verfallen, die 100-prozentige Lösung liefern zu wollen. Das schafft man nämlich nicht, denn dafür benötigt man unverhältnismäßig viele Ressourcen und zieht somit das Projekt unnötig in die Länge. Mit einer entsprechend akribisch erstellten Checkliste ausgestattet ist man gut gerüstet, um ein erfolgversprechendes Monitoring-Projekt zu starten.

Das Beispiel eines Hybrids unter den Monitoring-Systemen

„HL-Monitoring Module" wurde mit dem Ziel geschaffen, schnell und vor allem einfach eine Monitoring-Umgebung aufzusetzen. Selbstverständlich wird nicht an Funktionalität gespart. Lästiges Nach-Installieren von Komponenten oder Plug-ins gehört der Vergangenheit an. Der Kern erlaubt eine vollständige Server- und Netzwerk-Überwachung, zudem ist hier eine umfassende Oracle-, SQL-Server- und VMware-Überwachung in einem Werkzeug vereint. Hiermit lassen sich mehrere Ebenen auf einfache Art und Weise überwachen und die gewonnenen Messdaten sind sofort grafisch auswertbar.

Die Lösung ist kompatibel zu Nagios-/Icinga-Plug-ins, die nahtlos in-

Abbildung 4: Überwachung und Analyse

tegriert werden können. Auch eigenentwickelte Routinen lassen sich auf einfache Art einbinden. Somit müssen mühsam erstellte Programme nicht in den Papierkorb wandern.

Zu den besonderen Stärken zählt unter anderem die simple Administration, mit der neue Überwachungspunkte binnen Minuten eingebunden werden können. Dazu ein Beispiel: Um fünfundzwanzig Server und fünfundzwanzig Oracle-Datenbanken in die Überwachung aufzunehmen, ist ein Zeitbedarf von etwa zwei Stunden zu veranschlagen. Lästiges Schreiben von Konfigurationsdateien gehört der Vergangenheit an (siehe Abbildung 1).

Im vollständig webbasierten Cockpit können alle Mess-Ergebnisse direkt begutachtet und ausgewertet werden. Die Archivierung der erfassten Messdaten erfolgt automatisch. Hierbei sind keinerlei zusätzliche Installationen oder Aktionen seitens des Administrators notwendig. Neue Messpunkte werden automatisch gespeichert und ausgewertet (siehe Abbildung 2). Selbstverständlich umfasst die grafische Auswertung auch Daten, die im Oracle- oder SQL-Server-Umfeld gewonnen wurden. Die Korrelation der Daten ist ein einfaches, sehr mächtiges Mittel, um Schwachstellen oder Trends zu erkennen (siehe Abbildung 3).

Die Anzahl der virtualisierten Systeme auf Basis von VMware steigt stetig. Das Programm wird diesem Umstand gerecht und bietet eine umfassende Überwachung komplexer VMware-Umgebungen an. Die nahtlose Integration in die bestehende Monitoring-Landschaft schafft ungeahnte Möglichkeiten der Überwachung und Analyse (siehe Abbildung 4).

Fazit

Gleich zu Beginn sollte man wichtige Eckpunkte und Rahmenbedingungen festlegen, an denen man sich im Projektverlauf orientieren kann. Man sollte erst gar nicht versuchen, die „Eierlegende Wollmilchsau“ zu erschaffen, das haben schon viele versucht und sind gescheitert. Besser ist es, früh zu starten und früh in die Phase zu gelangen, in der schon wichtige Komponenten überwacht werden. Es müssen nicht gleich alle Komponenten sein, weniger ist mehr. Man kann später immer noch kontinuierlich aufstocken und somit das System besser kennenlernen.

Am besten schaltet man die Benachrichtigungen nacheinander ein, um nicht gleich zu Beginn mit einer E-Mail- oder SMS-Flut konfrontiert zu werden. Jedes System durchläuft eine Kalibrierungsphase, in der die Schwellwerte an die Realität angepasst, überflüssige Messpunkte entfernt und die Benachrichtigungsprofile den realen Gegebenheiten angeglichen werden. Monitoring ist ein Langzeitvorhaben, ein lebendes Projekt, ein System, das ständig verfeinert und optimiert wird.

Peter Bekiesch
peter.bekiesch@hl-services.de

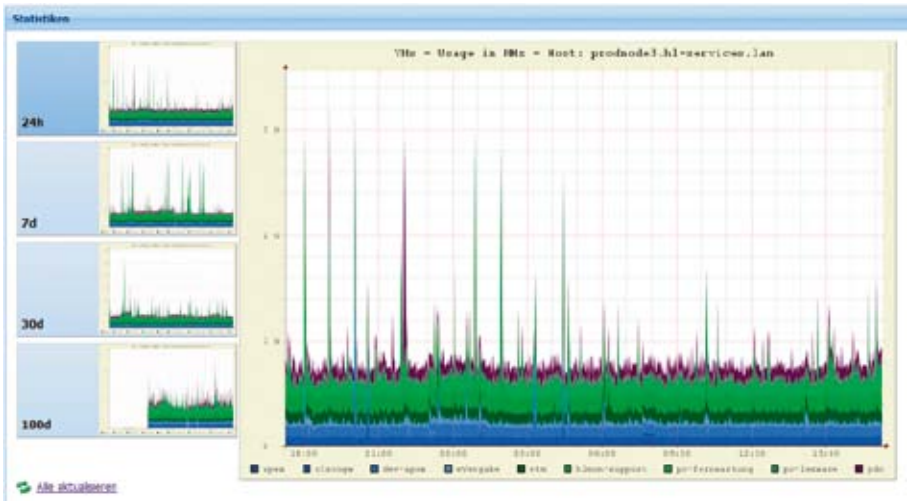


Abbildung 3: Die Korrelation der Daten

Last- und Performance-Monitoring von Business-Prozessen in der Datenbank

Kai Pillatzki, Stefan Triep und Andriy Terletskyy, Berenberg Bank

Um das Laufzeit-Verhalten von nachrichtengetriebenen Datenbank-Anwendungen sichtbar zu machen, werden detaillierte Informationen über das aktuelle Daten-Aufkommen sowie Verarbeitungs- und eventuell Wartezeiten verfügbar gemacht. Ein weiteres Ziel ist die Visualisierung dieser Daten und die Alarmierung im Falle einer Überschreitung bestimmter Schwellwerte. Dieser Artikel beschreibt das Last- und Performance-Monitoring von Business-Prozessen in der Datenbank mittels PL/SQL und Icinga.

Bei Berenberg wurde entschieden, dass die Business-Logik zur Datenverarbeitung in der Oracle-Datenbank liegen soll. Diese Logik wird mit der PL/SQL-Engine abgebildet. Für Einzelsatz-Verarbeitung werden Object Types (UDT), für Massendaten-Verarbeitung Packages sowie Oracle Advanced Queuing für die Jobsteuerung verwendet. Bei externen, nachrichtengetriebenen Systemen wie Börsen-Anbindungen werden die Messages erst in einer Queue gespeichert, von dort mit einem Scheduler-Job abgearbeitet und in einer Message-Tabelle gespeichert. Bei internen Systemen, die eine asynchrone oder parallele Verarbeitung erfordern, werden die Daten zuerst in einer Inbox-Message-Tabelle gespeichert und von dort über eine Eventqueue abgearbeitet (siehe Abbildung 1).

Diese Prozesse müssen aktiv überwacht werden, um zeitnah auf negative Verarbeitungs-Performance, Message-Staus oder Langzeit-Statistiken reagieren zu können. Das Monitoring soll keine negativen Einflüsse auf die Systemprozesse haben und einfach konfigurierbar sein. Die Daten-Visualisierung soll von der Messdaten-Ermittlung getrennt sein.

Architektur

Abbildung 2 zeigt eine Übersicht über die Monitoring-Architektur. Sie lässt sich in fünf grobe Bereiche unterteilen:

- Messpunkte (rot)
- Konfiguration (grün)
- Messdaten-Ermittlung (orange)
- Messdaten (blau)
- Visualisierung und Alarmierung (grau)

Messpunkte

Die Messpunkte beziehungsweise die zu überwachenden Objekte sind Queue- und Message-Tabellen. Eine wichtige Voraussetzung für eine einheitliche Messwert-Ermittlung sind Standardspalten an den Message-Tabellen. Die wichtigsten Spalten sind „APPLY_START“ (Verarbeitungsstart), „APPLY_END“ (Verarbeitungsende) und „ENQUEUE_ZST“ (Ankunftszeit) vom Typ „TIMESTAMP“ (siehe Abbildung 3). Anhand dieser Spalten lassen sich millisekundengenau die Verarbeitungs- oder Wartezeiten ermitteln. Die Spalten in den Message-Tabellen gehören zum Entwicklungsstandard.

Die Definition der Queue-Tabellen ist durch Oracle vorgegeben und bietet daher bereits alle notwendigen Spalten. Es werden aktuell mehr als 700 Messpunkte auf sieben Datenbank-Systemen überwacht. Jeder Messpunkt wird gleich behandelt und durchläuft folgendes Dreistufen-Prinzip:

1. *Einzelwert*
   Der Messpunkt enthält die Einzel

werte (siehe Abbildung 2, Message-Tabelle).
2. *Aggregation*
   Durch das Überwachungsintervall werden alle notwendigen Messdaten ermittelt (siehe Abbildung 2, „PA_MONITORING“).
3. *Aufbereitung*
   Durch eine Schnittstelle (siehe Abbildung 2, „V_MONITORING“) werden alle Messwerte für Icinga aufbereitet.

Aus Übersichtlichkeitsgründen sind folgende Messpunkttypen definiert:

- *Processed Messages*
  Die Anzahl der im Intervall verarbeiteten Messages. Es werden alle Messages ermittelt, die im Mess-Intervall liegen und einen „APPLY_START“-Zeitstempel besitzen.
- *Waiting Messages*
  Die Anzahl der im Intervall unverarbeiteten Messages, also „APPLY_START“ und „APPLY_END“, sind „null“.
- *Max. Processing Time*
  Die maximale Verarbeitungszeit aller Messages während des Intervalls in Millisekunden („APPLY_END“ – „APPLY_START“).
- *Max. Waiting Time*
  Die maximale (Warte-)Zeit einer unverarbeiteten Message während des Intervalls in Millisekunden („APPLY_START“ – „ENQUEUE_ZST“).
- *Arrived Time*
  Die maximale Dauer für den Message-Empfang aller Messages innerhalb des Intervalls („ENQUEUE_ZST“ – „SENDING_TIME“).

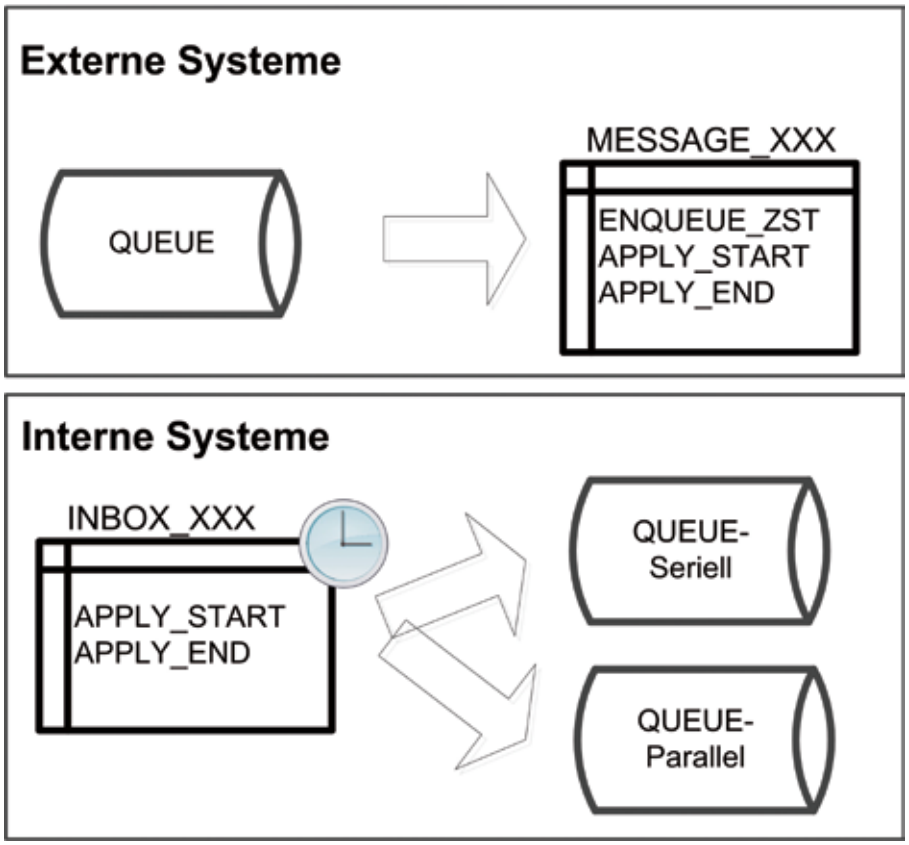


Abbildung 1: Anbindungsarten und deren Verarbeitung

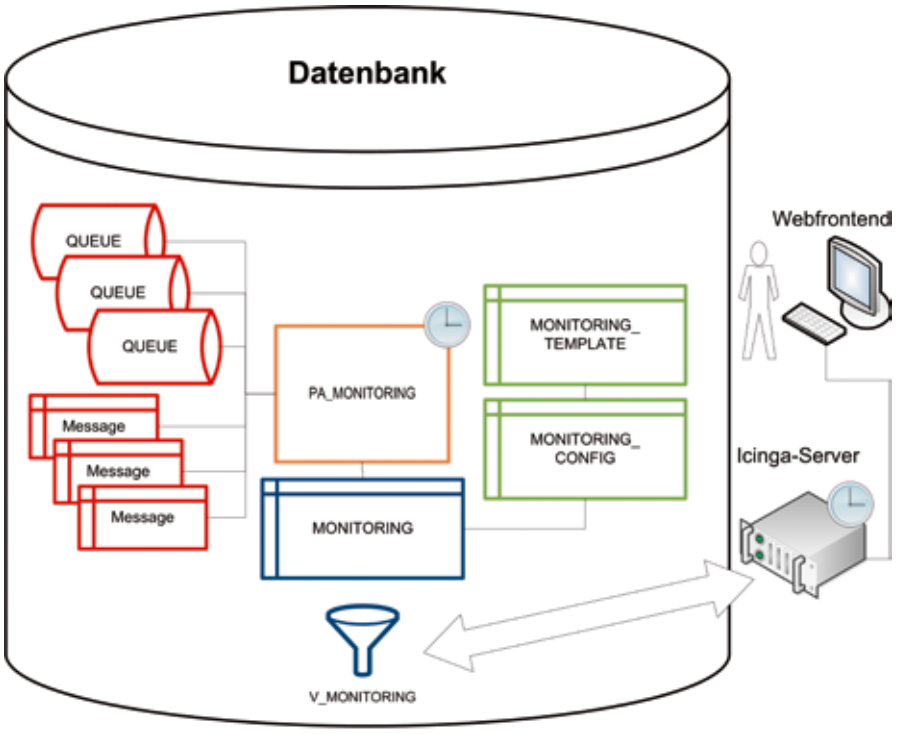


Abbildung 2: Übersicht über die Monitoring-Architektur

Konfiguration

Ein weiterer elementarer Bestandteil ist die Konfiguration der Messpunkte sowie deren Namen, Schwellwerte etc. Nachfolgend eine Auflistung der wesentlichen Konfigurationspunkte:

- *Messpunkt*
  Der einzelne Messpunkt ist über einen Eintrag in der Tabelle „MONITORING_CONFIG“ definiert.
- *Messpunkt-Typ*
  Die einzelnen Typen wurden bereits vorhin (siehe Messpunkte) erläu-

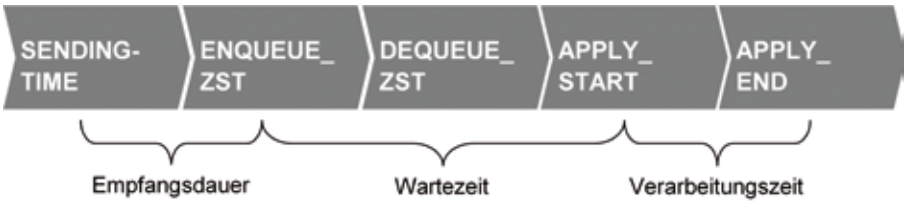


Abbildung 3: Wichtige Zeitpunkte bei der Message-Verarbeitung

tert. Jeder Messpunkt muss einem Typ zugeordnet sein.

- *Messpunkt-Bezeichnung*
  Jede Messpunkt-Bezeichnung ist eindeutig pro überwachte Datenbank. Die Bezeichnung wird im Alarmierungsfall ausgegeben und kann im Web-Frontend gesucht werden.
- *Schwellwerte*
  Schwellwerte dienen bei Überschreitung eines Messwertes zur Alarmierung. Jeder Messpunkt erhält individuelle Schwellwerte. Bei den Schwellwerten unterscheidet man wiederum:
  - *Warnung*
    Jede Überschreitung der Warnschwelle wird gemeldet.
  - *Kritisch*
    Kritische Schwellwerte sind per Definition größer als die Warnschwellwerte. Jede Überschreitung dieses Schwellwert-Typs sollte unbedingt zeitnah verfolgt und geklärt werden.
- *Template*
  Ein Template dient zur einheitlichen Darstellung von Messdaten. In diesem Fall haben sich die Autoren auf unterschiedliche Hintergrundfarben der Messpunktcharts je Datenbanksystem beschränkt. Icinga bietet jedoch eine große Auswahl an Darstellungsmöglichkeiten durch sogenannte „Special-Templates". Damit ist es möglich, mehrere Messpunkte in einem Chart zu kombinieren. Dazu später mehr im Abschnitt „Visualisierung".

Messdaten-Ermittlung

Zu jedem Messpunkt-Typ gibt es eine dynamische Abfrage (siehe „PA_MONITORING"), der der jeweilige Messpunkt mitgeteilt wird. So wird sichergestellt, dass jedem Messpunkt-Typ die gleiche Ermittlungslogik zugrunde liegt und die Messwerte untereinander vergleichbar sind.

Das Intervall zur Messdaten-Ermittlung muss nach den persönlichen Anforderungen bestimmt werden. Dabei ist zwischen einer hohen Taktung mit einer besseren Monitoring-Genauigkeit und der dadurch erzeugten Systemlast (sogenanntes „Grundrauschen") abzuwägen. In diesem Fall wurde entschieden, ein minütliches Intervall zu verwenden.

Das Intervall ist entscheidend für die Konfiguration im Zusammenspiel mit Icinga. Denn hier gilt das Alles-oder-nichts-Prinzip. Eine Mischung aus minütlichen und beispielsweise sekündlichen Intervallen je nach Messpunkt ist in der beschriebenen Konfiguration nicht möglich, da Icinga alle Messpunkte gleichzeitig abfragt. Bei Intervall-Mischung würde dies zu einer falschen oder fehlerhaften Darstellung der Messwerte führen.

Die Messdaten sind in der Tabelle „MONITORING" gespeichert und nach kurzer Zeit reorganisiert. Die Schnittstelle zu Icinga bildet die View „V_MONITORING". Icinga fragt minütlich die aktuellen Messdaten ab.

Visualisierung und Alarmierung

Zur Darstellung der Messwerte wurde Icinga ausgewählt. Icinga ist eine sogenannte „Fork" von Nagios. Diese freie Software wurde im Hause bereits von der Systemtechnik genutzt, um beispielsweise die Netzwerk-Auslastung oder andere Rechenzentrums-Informationen zu überwachen:

- *(Special-)Templates*
  Zur Darstellung von Messpunkten, auch Checks genannt, werden in der Regel Templates verwendet. Jeder Messpunkt ohne definiertes Template verwendet ein Default-Template. Eine besondere Ausprägung sind die sogenannten „Special-Templates". Hier können viele Parameter eines Charts vorgegeben werden wie Hintergrundfarbe, Diagrammtyp (Balken oder Linie), Farben der Balken oder Linien, Achsenbeschriftung und Titelbezeichnung. Außerdem bietet ein solches Template die Möglichkeit, mehrere Messpunkte in einem Chart darzustellen, beispielsweise Eingangs- und Ausgangs-Messages für alle Systeme.
- *Dashboard*
  Das Web-Frontend von Icinga bietet alle Informationen zu den Messpunkten. Jedoch fehlt eine komprimierte Gesamtansicht bestimmter Messpunkte über alle Datenbank-Systeme. Die zuvor beschriebenen Special-Templates können dazu genutzt werden, eine eigene Übersicht beziehungsweise ein Dashboard individuell zusammenzustellen. Die Dashboard-Anzeige wird als eine Fernseher- und Arbeitsplatzrechner-Ansicht mit Drill-Down-Funktionalität angeboten.

- *Alarm*
  Nachfolgend eine kurze Auflistung der möglichen Alarmierungswege:
  - *E-Mail:*
    Icinga übernimmt die Überwachung der Messpunkte und alarmiert standardmäßig per E-Mail.
  - *Nagios Status Monitor (NagStaMon)*
    Das Nagios-Desktop-Tool „NagStaMon" bietet eine E-Mail-Alarm-Alternative. Das Tool verbindet sich neben Nagios- auch mit Icinga-Servern und visualisiert eventuelle Schwellwert-Überschreitungen mithilfe eines Statusbalkens auf dem Desktop (siehe http://nagstamon.ifw-dresden.de).

Umgebungen

Es geht in erster Linie darum, die Produktions-Umgebung zu überwachen. Um negative Performance-Änderungen vor der Produktions-Übernahme erkennen zu können, sind auch eine frühzeitigere Überwachung der Pre-

Produktion und User-Acceptance-Test-Systeme (UAT) sinnvoll. Es wurde entschieden, auch in Integrations- und QS-Umgebung(en) das Monitoring zu ermöglichen.

Dabei ist jedoch zu beachten, dass sich beispielsweise die Messwerte aus der Produktion nicht direkt mit der QS-Umgebung vergleichen lassen, da die jeweiligen Umgebungen unterschiedliche Hardware-Konfigurationen verwenden. Es kann hier also nur die Performance der Integration mit der Integration-Umgebung und die der QS mit der QS-Umgebung verglichen werden.

Erfahrungen und Herausforderungen

Folgende Besonderheiten sind während der vergangenen Jahre im Laufe der Entwicklung der Monitoring-Landschaft aufgefallen:

- *Einheitliche Messwert-Ermittlung*
  Eine einheitliche Messwert-Ermittlung ist extrem wichtig. Es sollte dringend vermieden werden, pro Messpunkt oder Datenbank immer wieder neue Abfragen zur Messwert-Ermittlung zu schreiben.
- *Verwendung von Subviews bei Queue-Abfragen*
  Leider enthalten die „AQ$"-Views zu den Queues nicht alle die erwarteten Daten. Daher musste teilweise auf Data-Dictionary-Subviews (wie „*_H", „*_L", „*_S") zurückgegriffen werden.
- *Sommer-/Winterzeit bei Queues*
  Die Messwerte sind abhängig vom Queue-Erstellungszeitpunkt (Sommer- oder Winterzeit). Wurde eine Queue während der Sommerzeit angelegt, wird die Enqueue-Time um eine Stunde reduziert. Dies kann zu verwirrenden Messwert-Ergebnissen führen.
- *Enqueue nur sekundengenau*
  Der Enqueue-Zeitstempel bietet lediglich „DATE"-Genauigkeit. Wer mehr Genauigkeit will, also Millisekunden („TIMESTAMP"), muss selbst tätig werden. Als Workaround kommt der „JMS Payload"-Type zum Einsatz. Dieser besitzt ein zusätzliches Attribut mit Timestamp-Genauigkeit.
- *Messpunkt-Bezeichnung*
  Es ist darauf zu achten, dass die Messpunkt-Bezeichnung nicht zu lang wird. Icinga kennt zwar keine Längen-Begrenzung, aber es wird gegebenenfalls nicht die komplette Bezeichnung darstellt. Daher ist hier eine gewisse Namenskonvention (etwa „<Schema>.<Tabelle>.<Typ>") auch aus Übersichtlichkeitsgründen sehr hilfreich.
- *Automatische Messpunkt-Erstellung*
  Nutzt man eventuell mehrere Message-Typen in einer Queue und möchte pro Typ Messwerte erhalten, ist bei neuen Typen immer die Konfiguration zu erweitern. Einfacher ist es, man lässt sich beim ersten Auftreten eines neuen Typs automatisch einen neuen Konfigurationseintrag mit Standard-Schwellwerten erstellen.
- *Automatische De-/Reaktivierung von Messpunkten*
  Speziell bei der Einführung neuer Messpunkte kann es schwierig sein, realistische Schwellwerte zu definieren. Gelingt einem dies nicht, kann man nach der Einführung neuer Messpunkte möglicherweise durch viele überflüssige Alarm-E-Mails überrascht werden. Eine weitere Herausforderung besteht darin, dass die Messwert-Ermittlung trotz des Test-Monitorings auf den Test-Systemen langsamer in Produktion läuft. Hierfür wurde eine automatische, temporäre De- und Reaktivierung einzelner Messpunkte eingeführt. Läuft beispielsweise eine Messwert-Ermittlung länger als fünf Sekunden, wird dies protokolliert. Geschieht dies dreimal hintereinander, wird der Messpunkt für eine bestimmte Zeit deaktiviert. Der fehlerhafte Messpunkt wurde nun bereits dreimal gemeldet. Nach Ablauf des Zeitfensters reaktiviert sich der Messpunkt und versucht sich wieder an der Messwert-Ermittlung.

Fazit

Die beschriebene Umsetzung bietet eine kostengünstige Monitoring-Umgebung: Man benötigt einfache PL/SQL- und SQL-, jedoch tiefere Icinga-Kenntnisse. Wer keine Out-of-the-Box-Lösung erwartet und einen gewissen Startaufwand zum Aufbau und zur Konfiguration nicht scheut, erhält ein Monitoring-Paket, das sich an viele eigene Bedürfnisse anpassen lässt. Durch die Verwendung von Icinga anstelle von Nagios bekommt man gegebenenfalls schnelle Unterstützung von den Entwicklern, falls Erweiterungen dort gewünscht sind.

Diese Konfiguration ist nun schon seit einigen Jahren im Einsatz. Dank der eigenen Monitoring-Logik in PL/SQL können schnell und problemlos neue Messpunkte ergänzt und bei Bedarf optimiert werden. Störungen werden jetzt schneller erkannt und können deutlich zügiger behoben beziehungsweise kommuniziert werden.

Kai Pillatzki
(Datenbank-Entwickler)
kai.pillatzki@berenberg.de

Stefan Triep
(Icinga)
stefan.triep@berenberg.de

Andriy Terletskyy
(Datenbank-Architektur)
andriy.terletskyy@berenberg.de

Überwachen von Applikationslogik mittels Enterprise Manager 12*c*

Bernhard Wesely, Trivadis Delphi GmbH

Der Oracle Enterprise Manager überwacht standardmäßig sämtliche Infrastruktur-Komponenten einer Umgebung auf ihre Funktionsfähigkeit. Dies ist allerdings nur eine Seite der Medaille, denn auch die Applikationslogik selbst sollte überwacht werden, um Fehlern möglichst frühzeitig auf die Schliche zu kommen. Da hier Standard-Metriken nicht greifen, ist eine Möglichkeit gesucht, selbst Metriken zu erstellen.

Nach der Installation des Enterprise Manager (EM) verfügen wir bereits über eine beachtliche Anzahl an mitgelieferten Standard-Metriken. Damit lassen sich Oracle-Komponenten wie Server, Datenbanken und WebLogic Server, aber auch Systeme von Drittherstellern wie etwa der Microsoft SQL Server überwachen. Es ist jedoch ein Leichtes, sich Fälle auszumalen, in denen die Datenbank innerhalb ihrer normalen Parameter läuft, die Applikation ihre Aufgabe dennoch nicht ordnungsgemäß wahrnimmt. Um auch diese abdecken zu können, unterstützt der EM die sogenannten „Metric Extensions". Damit ist es möglich, eigene Metriken zu erstellen und damit zu überwachen, zu alarmieren und aufzuzeichnen.

User Defined Metrics

Von früheren EM-Versionen kennen wir bereits das Konzept der User Defined Metrics. Metric Extensions sind eine Erweiterung dieses Konzepts. Die wichtigsten Änderungen im Überblick:

- *Metric Extension Lifecycle*
  Wie die meisten Software-Projekte durchläuft eine Metric Extension nun einen Entwicklung/Test/Produktions-Zyklus
- *Erweiterte Protokoll- und Ziel-Unterstützung*
  Es sind jetzt mehr Protokolle und Ziel-Typen unterstützt
- *Integrierte Software-Verteilung*
  Sollte es nötig sein, für die Erfassung einer Metrik etwa ein Perl-Script auszurollen, so kann dies jetzt automatisch geschehen
- *Zentrale Definition*
  User Defined Metrics wurden pro Ziel definiert. Hat man das Ziel oder den Agenten entfernt, wurden die UDMs ebenfalls gelöscht. Metric Extensions hingegen sind zentral definiert und hängen nicht mehr an einem Ziel.

Entwickeln einer Metric Extension

Wie bereits erwähnt, durchlaufen Metric Extensions nun einen Entwicklungszyklus mit folgenden Stufen (siehe Abbildung 1):

- *Entwicklung*
  In dieser Stufe wird die Metrik in Version 1 erstellt. Anpassungen sind hier jederzeit möglich. Getestet wird die Metrik im Zuge des Erstellungs-Wizard. Ist die Entwicklung abgeschlossen, wird die Metrik als „Deployable Draft" gespeichert und geht nun in den Test-Zustand über.
- *Test*
  Dieser Deployable Draft kann wie jede andere Metrik auf beliebige Ziele ausgerollt werden, um das Erfassen der Daten sowie das Senden von Alarmen zu testen. Die Metrik lässt sich an dieser Stelle nicht mehr verändern; dazu ist eine neue Version erforderlich. Um die Metrik nun allgemein verwenden zu können, muss sie veröffentlicht werden und geht damit in den Published-Status über.
- *Produktion*
  Befindet sich die Metric Extension nun im Published-Status, kann sie über Monitoring-Templates automatisch auf den Zielsystemen ausgerollt werden. Wie auch im vorhergegangenen Test-Status lassen sich Metric Extension hier manuell auf einzelne Ziele ausrollen.

Abbildung 2 zeigt eine Metric Extension in allen drei Status.

Credential Sets

Um Metric Extensions auf dem jeweiligen Zielsystem ausführen zu

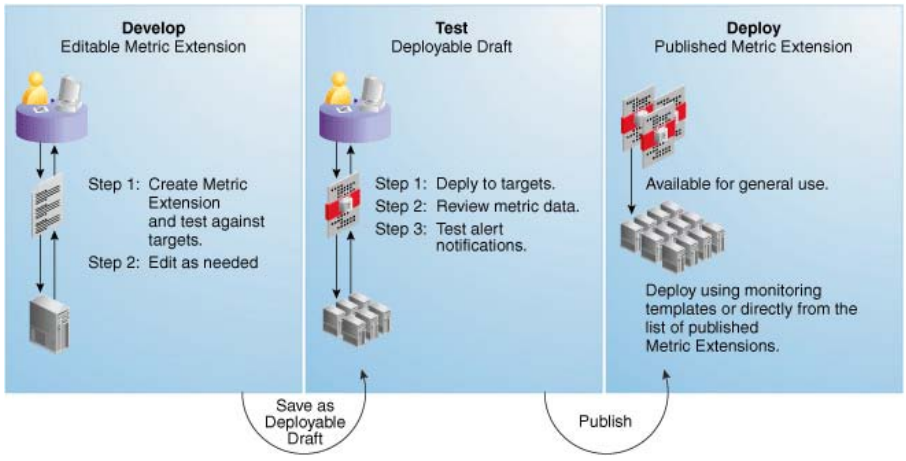


Abbildung 1: Metric Extension Lifecycle (Quelle: Oracle)

Name	Target Type	Display Name	Version	Description	Status	Deployed Targets
ME$email_queue	Database Instance	E-Mail Queue	1		Published	1
ME$email_queue	Database Instance	E-Mail Queue	2		Deployable Draft	0
ME$email_queue	Database Instance	E-Mail Queue	3		Editable	0

Abbildung 2: Metric Extension in unterschiedlichen Status

```
oracle@oms01:~/ [oms12c] emcli create_credential_set
     -set_name="Metric Extension Monitoring Credentials"
     -target_type=oracle_database
-supported_cred_types=DBCreds -monitoring

Credential set "Metric Extension Monitoring Credentials"
created successfully.
```

Listing 1: „emcli"-Kommando, um ein Monitoring Credential Set anzulegen

können, werden natürlich Zugangsdaten benötigt. Die Standard-Enterprise-Manager-Metriken verwenden hierzu die sogenannten „Default Monitoring Credentials". Für Datenbanken werden diese während der Veröffentlichung von Zielen im Credential Set „Monitoring Database Credentials" gespeichert. Typischerweise ist dies der Datenbank-Benutzer. Dieser hat jedoch nur eingeschränkte Rechte in der Datenbank und kann in den meisten Fällen nicht auf die Applikations-Tabellen zugreifen. Um nun einen eigenen Benutzer spezifizieren zu können, muss ein neues Credential Set angelegt werden. Dies geschieht über das Enterprise Manager Command Line Interface „emcli" (siehe Listing 1).

Um dem nun angelegten Credential Set eine Username/Passwort/Ziel-Kombination zuzuweisen, findet sich unter „Setup -> Security -> Monitoring Credentials" eine grafische Oberfläche. Hier wählt man die passende Kombination von Zielsystem und Credential Set und vergibt über den Wizard hinter dem Button „Set Credential" einen Benutzernamen und ein Passwort (siehe Abbildung 3).

Während der Erstellung der Metric Extension kann nun das zu verwendende Credential Set ausgewählt werden (siehe Abbildung 4). Achtung: Username und Passwort werden erst während der Ausführung gegen ein Zielsystem ausgewertet; es ist daher zwingend notwendig, Username und Passwort vor der Verteilung der Metrik für jedes System zu definieren.

Adapter

Adapter stellen die Kommunikationsmethode zwischen Metrik Extension und Zielsystem dar. Es sind folgende Adapter verfügbar:

- OS Command – Single Column
- OS Command – Multiple Values
- OS Command – Multiple Columns
- SQL
- SNMP
- Java Management Extensions (JMX)

Unterschiedliche Ziel-Typen unterstützen immer nur ihr passendes Subset an Adaptern. Es ergibt beispielsweise wenig Sinn, eine Datenbank mittels „JMX" oder einen Host mittels „SQL" kontaktieren zu wollen. Die Definition der möglichen Adapter pro Zieltyp ist bereits vorgegeben.

Ein Fallbeispiel

Angenommen, ein Web-Shop verschickt nach einer Bestellung automatisch Bestätigungs-E-Mails über die bestellte Ware. Gesteuert wird dieser Versand über eine Tabelle namens „WEBSHOP.EMAIL". Diese besteht aus dem Datum, an dem der Datensatz erzeugt wurde, dem Empfänger und dem Text der E-Mail.

Nach einer Bestellung wird ein Datensatz in dieser Tabelle erzeugt. Asynchron dazu läuft ein Job, der die Tabelle ausliest und E-Mails generiert. Für jede gesendete E-Mail wird die entsprechende Zeile anschließend gelöscht. Dieses einfach gehaltene Beispiel lässt sich natürlich auf jede beliebige Komplexität erweitern.

Bevor wir mit der Implementierung angefangen, muss definiert werden, was überwacht werden soll. Zwei Fragen bieten sich hier vordergründig an:

- *Ist der Versand der E-Mails schnell genug oder stauen sich Mails auf?*
  Die zu erfassende Metrik ist dann die Gesamtzahl der E-Mails in der Queue.
- *Wächst die Tabelle zu schnell an, obwohl das Limit von Punkt 1 noch nicht erreicht ist?*
  Die zu erfassende Metrik ist dann die Anzahl der in den letzten fünf Minuten hinzugekommenen Einträge.

Erstellen der Metric Extension

Über die Menüpunkte „Enterprise -> Monitoring -> Metrik Extensions" gelangt man zu der zentralen Administrationsseite für Metric Extensions. Hier können neue Metriken erstellt, editiert, ausgerollt und gelöscht werden. Über den Button „Create" lässt sich hier eine neue Metrik erstellen. Auf der ersten Seite geben wir allgemeine Informationen wie den Namen der Metric Extension oder den Ziel-Typ an. In diesem Fall ist das Ziel vom Typ „Database Instance".

Eine Metric Extension hat immer einen kurzen (intern verwendet) und einen langen Namen (Display-Name). Der Display-Name darf beispielsweise Leerzeichen enthalten und ist auch der Name, der eigentlich immer angezeigt wird. Passend zu unserem Ziel wählen wir noch den Adapter namens

„SQL“ aus. Im unteren Teil der Seite kann noch eingestellt werden, ob die Metrik aktiv ist beziehungsweise wie oft sie vom Zielsystem gesammelt werden soll. Diese Frequenz stellen wir auf fünf Minuten. Wollen wir die erhobenen Daten im Repository historisiert speichern, wählen wir unter „Use of Metric Data“ noch den Punkt „Alerting and Historical Trending“ an. Durch Klicken des „Next“-Buttons gelangen wir auf die nächste Seite.

Nun wird der gewählte Adapter konfiguriert. Im Falle des SQL-Adapters geht es hier um das auszuführende Statement. In unserem einfachen Beispiel reicht die simple SQL-Query „select count(*) from webshop.email“ aus, um die Metrik zu erfassen. Weitere Konfigurationen wie die Steuerung von PL/SQL oder die Verwendung von Bind-Variablen können weiter unten auf der Seite vorgenommen werden. Wir klicken den „Next“-Button.

Hier definieren wir die Spalten, die unser Statement zurückliefern wird. Aktuell ist dies nur eine, das „count(*)“. Über die Menüpunkte „Add -> new metric column“ erreichen wir ein Fenster, das uns einige Konfigurationsmöglichkeiten bietet. Wieder werden ein kurzer sowie ein langer Name verlangt. Zusätzlich können wir hier gleich Schwellwerte und angepasste Alarmmeldungen definieren. Für dieses Beispiel verwenden wir „anzahl_emails“ als Name, setzen den Comparison Operator auf „>“ (größer) und die Schwellwerte auf 5.000 („Warning“) und 10.000 („Critical“). Da sich dieser Alarm von selbst auflöst, sobald die Schwellwerte wieder unterschritten werden, lassen wir das Feld „Manually Clearable Alert“ auf „false“. Metriken, die keinen definierten „OK“-Zustand haben, wie Fehlermeldungen in Logfiles, würde man hier auf „true“ stellen. Mittels „OK“-Button kommen wir wieder zur vorhergehenden Ansicht zurück.

Der aufmerksame Beobachter fragt sich nun, wo die zweite Metrik (Rate der hinzugekommenen E-Mails pro fünf Minuten) abgeblieben ist. Schließlich erfassen wir diese Information ja nicht gesondert mit unserem „count(*)“. Hier hilft uns der Enterprise Manager, der diese Werte für uns errechnet. Wir definieren eine zweite Spalte auf Basis der ersten. Diese erzeugen wir über den Menüpunkt „Add -> Rate (per 5 min.) metric column on anzahl_emails“ (siehe Abbildung 5).

Der Name der Spalte ist hier schon vorgegeben, wir definieren nur den Display-Namen und den Schwellwert. Als Beispiel nehmen wir hier 100 („Warning“) und 200 („Critical“). Wir klicken auf „OK“. Abbildung 6 zeigt die fertige Spalten-Definition.

Auf der nächsten Seite geben wir die Credentials an, die der SQL-Adapter zum Einloggen in die Datenbank verwendet. Sollte der Benutzer alle erforderlichen Rechte besitzen, ist hier nichts zu verändern. Falls diese nicht ausreichen, muss ein Credential Set, wie im Kapitel „Credential Sets“ beschrieben, angelegt, konfiguriert und ausgewählt werden.

Im vorletzten Schritt kann die Metrik gegen ein oder mehrere Ziele getestet werden. Durch Klicken auf „Add“

Abbildung 3: GUI zur Verwaltung der Monitoring Credentials

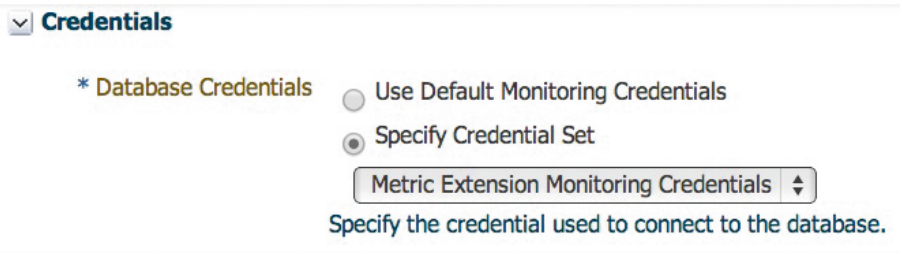


Abbildung 4: Auswahl des Credential Set während der Metric-Extension-Erstellung

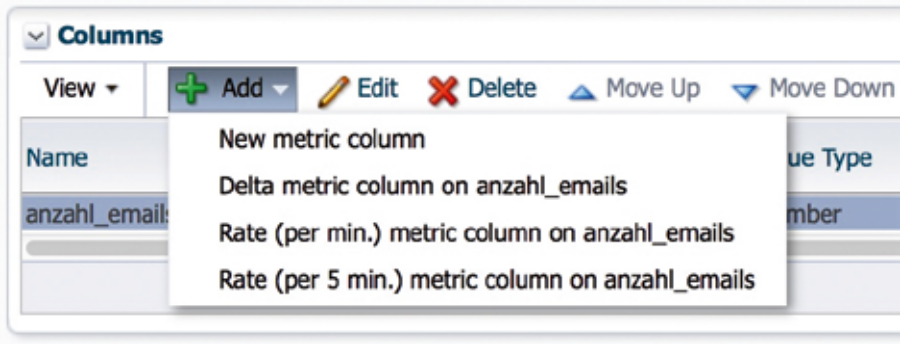


Abbildung 5: Hinzufügen der Rate-Spalte

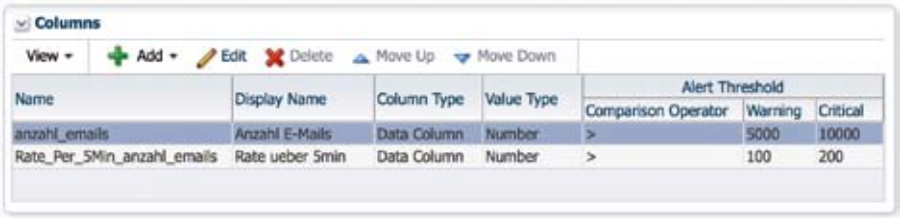


Abbildung 6: Spalten-Definition der Metric Extension

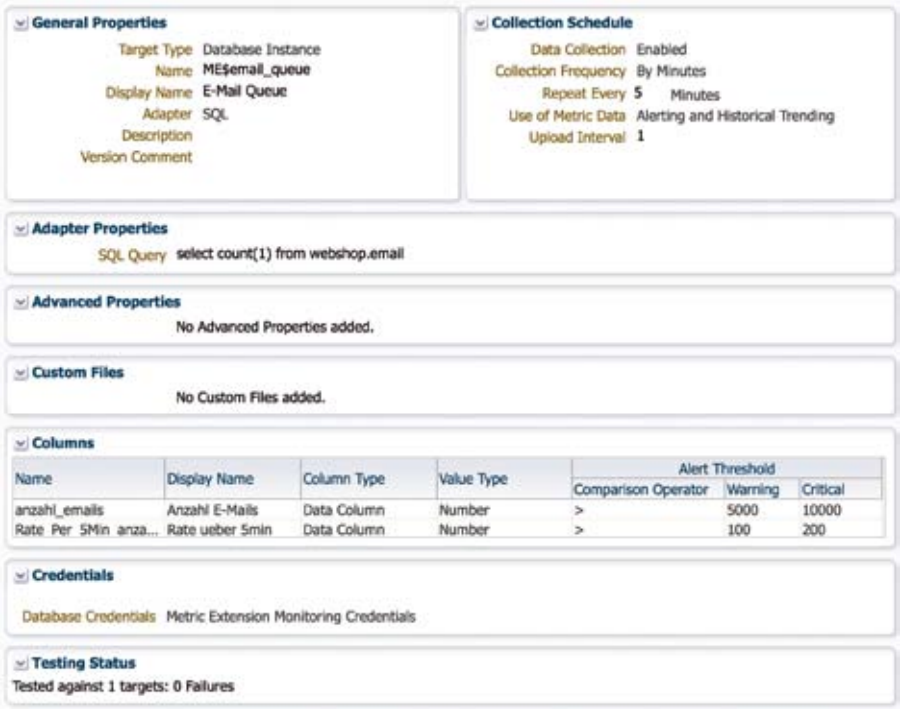


Abbildung 7: Zusammenfassung der Metric Extension

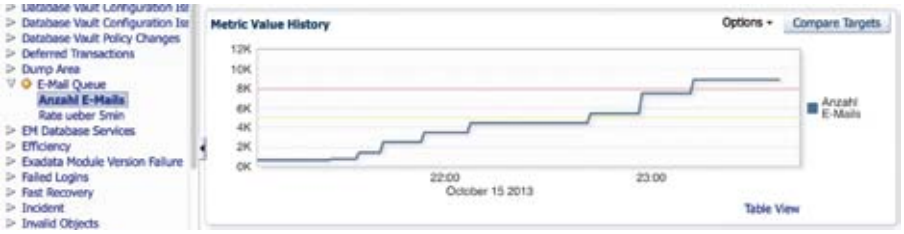


Abbildung 8: Graph der historisierten Metrikdaten

können Datenbanken hinzugefügt werden. Wir wählen hier unsere Webshop-Datenbank aus. Mittels „Run Test“ wird die Metrik auf dem Zielsystem ausgeführt. In der unteren Hälfte der Seite sehen wir nach kurzer Zeit das Resultat der Auswertung. Hier wird nur die selektierte Spalte dargestellt, die Rate wird ja über einen vergangenen Zeitraum berechnet und kann hier nicht angezeigt werden. Sollte alles passen, kommen wir über einen Klick auf „Next“ zur Zusammenfassung (siehe Abbildung 7); über einen weiteren Klick auf „Finish“ wird die Metric Extension angelegt.

Ausrollen der Metric Extension

Nach Erstellung der Metric Extension wollen wir sie natürlich am konkreten Objekt testen. Hierzu erstellen wir einen Deployable Draft. Wir wählen die Metrik aus und verwenden den Punkt „Actions -> Save as Deployable Draft“. Nun lässt sich die Metrik via „Actions -> Deploy to Targets“ ausrollen. Im nun folgenden Assistenten fügen wir durch „Add“ die Datenbank des Web-Shops hinzu. Nun wird die Metrik ausgerollt und entsprechend dem Collection Schedule gesammelt.

Events, die von Metric Extensions ausgelöst werden, unterscheiden sich für das Incident System nicht von Events der Oracle-eigenen Metriken. Genau wie die Standard-Metriken können Events über ein Ruleset, das auf „Targets“ zutrifft, in Kombination mit Rules des Typs „Metric Alert“ sowie „Metric Evaluation Error“ in das Incident System gebracht werden. „Metric Evaluation Error“ wird gern vergessen, ist aber gerade im Umfeld der Metric Extensions wichtig, um herauszufinden, ob der letzte Applikations-Rollout die Tabellenstruktur so verändert hat, dass die Metrik nicht mehr ausgewertet werden kann.

Anzeigen der Metrikwerte

Sofern wir bei der Erstellung der Metric Extension angegeben haben, dass wir die Daten auch für „Historical Trending“ speichern wollen, werden die Metrikdaten im Repository historisiert gespeichert. Diese Werte lassen sich in jeder Ansicht grafisch darstellen. Ob dies nun auf der Datenbank-Homepage über „Database -> Monitoring -> All Metrics“ erfolgt (siehe Abbildung 8), über Reports oder im „Charts“-Feature von Gruppen, ist ganz dem EM-Administrator überlassen. Über Reports lassen sich die Werte, Graphen und Alarme auch an Benutzer, die keinen Zugriff auf den Enterprise Manager haben, verteilen. Dies kann per Web-Zugriff auf eine spezielle Enterprise-Manager-Seite oder per E-Mail passieren.

Fazit

Es hat sich viel seit den User Defined Metrics der vergangenen EM-Versionen getan. Auch wenn die Kernfunktionalität kaum verändert wurde, gibt es rund herum nun viel mehr Möglichkeiten, eigene Anforderungen abzubilden. Mithilfe der EM-Standard-Metriken und auch der Metric Extensions ist es nun möglich, die eigene Infrastruktur lückenlos zu überwachen. Aber Achtung: Metric Extensions sind Teil des jeweiligen zielspezifischen Performance-Diagnose-Pack, wie zum Beispiel des Database-Diagnostics-Pack für Datenbanken, und damit lizenzpflichtig.

Bernhard Wesely
Bernhard.Wesely@trivadis.com

Datenbank-Monitoring mithilfe eigenständig entwickelter Tools

Jens Brill, eXirius GmbH

Die Grundlage einer optimal funktionierenden Datenbank ist eine kontinuierliche Überwachung. Oracle stellt hierfür das Diagnostik Pack bereit, bei dem es sich um eine Option der Enterprise Edition handelt. Doch was tun, wenn keine Enterprise Edition lizenziert wurde oder die Diagnostik-Pack-Option nicht erworben wurde?

Ein selbst entwickeltes Tool kann diese Aufgabe übernehmen. Eine solche Lösung ist das von eXirius IT entwickelte Tool „Database Live Monitor“ (DBLM). Das Ziel der Entwicklung eines Tools wie dem DBLM sollte es sein, eine erhöhte Verfügbarkeit zu erreichen, eine Performance-Überwachung zu gewährleisten und die Arbeitszeit der Administratoren zu optimieren. Dabei sollte das Tool die Datenbank möglichst wenig belasten.

Die Notwendigkeit einer Selbstentwicklung ergibt sich aus der täglichen Erfahrung und Arbeit der Administratoren mit Oracle-Datenbanken. Ein großer Teil dieser Arbeit besteht aus Routine-Tätigkeiten wie der Kontrolle des Backups und der Log-Dateien oder die Überwachung der Füllstände der Tablespaces. Ein eigenständig entwickeltes Tool kann einen großen Teil dieser Aufgaben übernehmen und warnen, sobald Handlungsbedarf besteht. Ziel ist es, Probleme nicht nur zu lösen, wenn sie auftreten, sondern sie proaktiv zu verhindern.

Bei der Entscheidung für eine Programmiersprache sollten verschiedene Gesichtspunkte eine Rolle spielen. Wo möchte ich das Monitoring-Tool einsetzen? Wie umfangreich sollen die Überwachungs-Funktionen sein? Welches Know-how für die Programmierung ist im Unternehmen bereits vorhanden?

Natürlich stellt sich die Frage, warum hier ausgerechnet eine als altmodisch und kryptisch geltende Programmiersprache wie Perl zum Einsatz kommt. Neben der Ausgereiftheit der bereits im Jahr 1985 in einer ersten Version implementierten Sprache spielen die freie Lizenz, die weitgehende Plattform-Unabhängigkeit und der riesige Umfang an Erweiterungs-Modulen eine wichtige Rolle. Spezielle Anforderungen müssen somit oft nicht aufwändig programmiert werden, sondern es kann einfach eine entsprechende Bibliothek („Modul“) benutzt werden.

Perl kompiliert ebenso wie Java den als „ASCII/UTF-8“-Text vorliegenden Quellcode zunächst in einen Bytecode. Dieser wird jedoch im Unterschied zu Java von der Perl-Virtuellen-Maschine ausgeführt. Dies ist einerseits sehr performant, besitzt aber andererseits die Flexibilität einer interpretierten Skript-Sprache, was unter anderem bedeutet, dass Zeichenketten zur Laufzeit als Quellcode nachkompiliert und ausgeführt werden können. Da es sich bei einem Monitoring-Tool um ein betriebskritisches Überwachungssystem handelt, sollte hier auf eine bewährte Technik zurückgegriffen werden.

Arbeitsweise und Aufbau

Der Aufbau des DBLM ist so flexibel wie möglich gestaltet. Es ist in allen Oracle-Datenbank-Versionen und auf allen Betriebssystemen einsetzbar. Die Überwachung einer Single-Instanz ist ebenso möglich wie die Überwachung von Standby-Datenbanken und Real Application Clustern (RAC). Die Anzahl der Datenbanken spielt dabei keine Rolle.

Die programmierten Überwachungs-Parameter sind in zwei Bereiche unterteilt: diejenigen, die den Betrieb der Datenbank gewährleisten, und diejenigen, die Abfragen zur Performance durchführen. Alle Parameter sind modular programmiert und können unabhängig voneinander aktiviert und deaktiviert werden. Falls notwendig, lassen sich aufgrund des modularen Aufbaus neue Parameter definieren und jederzeit einbinden. Grenzwerte werden individuell an die zu überwachenden Datenbanken angepasst.

Es sind zwei Kategorien von Parametern definiert. So gibt es Kollektoren, die einen bestimmten Zustand anzeigen. Nur die Zustände „OK“ oder „NICHT OK“ sind möglich. Der Kollektor zeigt also an, ob eine Datenbank geöffnet ist oder nicht. Die Indikatoren ermitteln einen Systemzustand, der mit vorher definierten Schwellenwerten verglichen wird. Hierbei handelt es sich etwa um Füllstände von Tablespaces.

Nicht immer ist die Datenbank für einen Stillstand verantwortlich. Daher überwacht DBLM auch wichtige Funktionen des Servers, um Fehlern an dieser Stelle vorzubeugen.

Benachrichtigung und Problemlösung

Die wichtigste Aufgabe eines Monitoring-Tools ist es, die gewonnenen Erkenntnisse dem Datenbank-Administrator in geeigneter Form bereitzustellen. DBLM versendet in seinen Grundeinstellungen die Informationen per E-Mail und gibt Handlungsanweisungen zur Problemlösung. Andere Methoden der Benachrichtigung sind integriert, etwa die später angesprochene Integration in andere Monitoring-Systeme.

Für die Ergebnisse der Abfragen ist es unumgänglich, eine Form zu wählen, in der Probleme sofort erkannt werden. Das bedeutet, wichtige von unwichtigen Informationen zu trennen und herauszufiltern. Bekommt der Datenbank-Administrator hundert E-Mails am Tag mit für ihn unwichti-

gen Informationen, tritt ein Gewöhnungseffekt ein. Eine wichtige E-Mail, bei der dringender Handlungsbedarf besteht, wird dann womöglich übersehen und als Folge kommt es zu einem Ausfall der Datenbank, der hätte verhindert werden können.

Viele Unternehmen setzen bereits fertige Überwachungs-Software ein. Oft bieten diese jedoch nicht die Möglichkeit zur Überwachung von Oracle-Datenbanken beziehungsweise die gelieferten Informationen entsprechen nicht dem geforderten Umfang. Daher ist es sinnvoll, ein selbst entwickeltes Tool in eine solche Überwachungs-Software zu integrieren.

Häufig kommt es vor, dass Oracle-Administratoren auch andere Überwachungsaufgaben übernehmen. Die Einbindung des selbst entwickelten Tools in ihrer Monitoring-Systeme stellt sicher, dass alle Informationen an einer zentralen Stelle zusammenlaufen.

Strategische Vorgaben an die IT verbieten oftmals den Betrieb weiterer Software-Komponenten auf für andere Zwecke produktiv genutzten Servern. Eine Black-Box-Lösung bietet die Alternative, das Monitoring-Tool Policy-konform zu installieren. Für DBML existiert hierfür eine Appliance-Variante, bei der alle für den Betrieb nötigen Komponenten vorinstalliert sind, sodass der Aufwand für die Inbetriebnahme in der Regel deutlich reduziert ist und sich auf die Anpassung der Konfiguration beschränkt.

Häufig steht im Unternehmen ohnehin eine Virtualisierungsplattform wie VMware, Citrix und/oder Xen zur Verfügung, auf der man dann eine DBLM-vServer-Appliance betreiben kann. Alternativ wäre sogar ein Betrieb auf alter Hardware mithilfe einer Live-CD oder eines USB-Sticks denkbar. Neben dem geringeren Aufwand für die Installation hat diese Lösung auch im laufenden Betrieb Vorteile. Bei Updates muss nur die Konfiguration übernommen werden, der dann modernere Unterbau wird einfach ausgetauscht.

Fazit

Die Entwicklung eines eigenen Monitoring Tools ist mit einem Aufwand verbunden, der nicht unterschätzt werden sollte. Je nach Anzahl der zu überwachenden Datenbanken und gewünschtem Umfang des Überwachungstools kann sich der Entwicklungsaufwand jedoch lohnen, da dadurch viele Prozesse automatisiert werden. Es ergibt sich eine enorme Zeitersparnis für die Oracle-Administratoren; das Tool hilft, Ausfallzeiten zu minimieren, und kann auf Performance-Probleme hinweisen.

Jens Brill
jens.brill@exirius.de

Oracle Database 12c: Das erste Patch-Set-Update bringt Neuerungen

Der übliche Rhythmus für das Erscheinen von Security Patch Updates (SPUs) und Patch Set Updates (PSUs) betrifft mit dem am 15. Oktober 2013 erschienen Update erstmals die neue Datenbank-Version 12c. Die Administratoren werden von einer Neuerung überrascht, denn die vor einigen Monaten erst von Critical Patch Updates (CPUs) in Security Patch Updates umbenannten Pakete werden abgeschafft — für 12c und zukünftige Versionen wird es ausschließlich Patch Set Updates geben. Das unterstützt einerseits die bei vielen Installation gängige Praxis, eben diese PSUs mehr oder weniger regelmäßig einzuspielen, beraubt Administratoren von sicherheitsrelevanten Systemen aber der Möglichkeit, exakt die geforderten (Sicherheits-)Patches einzuspielen. PSUs enthalten neben Sicherheits-Patches auch weitere Patches, die laut Oracle folgenden Bedingungen genügen:

- Die Patches werden extrem gut von Oracle kontrolliert und getestet
- Die Auslieferung als Patch-Bundle reduziert Konflikte bei der Patch-Installation
- Da API- sowie Optimizer-Änderungen etc. ausgeschlossen sind, sind nur minimale Funktionstests für die Inbetriebnahme erforderlich

Da diese Eigenschaften für die PSUs der Vergangenheit durchaus zutreffen, sind diese bei den DBAs sehr beliebt. Diejenigen, die in der Vergangenheit trotzdem lieber CPUs/SPUs eingespielt haben, sollten sich also an die neue Vorgehensweise gewöhnen. Für Oracle 11g wird es bis zum Ende der Laufzeit weiterhin SPUs geben.

Eine der zentralen Fragen bei den SPUs und PSUs ist grundsätzlich, ob diese unbedingt eingespielt werden sollen. Oft lässt sich das durch einen Blick in die Risiko-Matrix klären: Wenn bei den eingesetzten Produkten keine Schwachstellen mit hohem Risiko vorhanden sind, kann man gegebenenfalls darauf verzichten.

Diesmal ist mit CVE-2013-3826 eine Schwachstelle vorhanden, die es ermöglicht ohne Passwort über Netzwerk auf Datenbank-Inhalte lesend zuzugreifen. Betroffen sind alle aktuell unterstützten Oracle-Versionen. Daher ist ausdrücklich empfohlen, das aktuelle Update – in welcher Form auch immer – einzuspielen.

In einer Anmerkung zu dieser Schwachstelle verweist Oracle auf die Tatsache, dass die Verschlüsselung von Oracle-Net-Verbindungen nicht mehr Bestandteil der Advanced-Security-Option ist, sondern für alle Editionen kostenlos zur Verfügung steht. Dies ist wohl ein erneuter Hinweis darauf, dass Oracle den Anwendern die Verschlüsselung von Verbindungen an die Datenbank stärkstens empfiehlt.

Dierk Lenz
http://blog.hl-services.de

Die Top-10-Monitoring-SQL-Befehle

Marco Patzwahl, MuniQSoft GmbH

Viele Kunden haben mehr als hundert Datenbanken zu betreuen. Da kommt man ohne automatisierte Überwachungs-Skripte nicht sehr weit. Deswegen soll diese kleine Sammlung an SQL-Befehlen helfen, das Monitoring der Datenbanken zu vereinfachen. Der Fokus liegt dabei auf der Datenbank-Version 12.1, jedoch sind Anmerkungen bei den Befehlen, wie man diese auch in älteren Datenbank-Versionen verwenden kann. Die Skripte sind zum Download verfügbar.

1. Alert-Log überwachen

Ab Version 11.2.0.2 kann die Alert-Datei (XML-Variante) mittels „SELECT" ausgewertet werden (siehe Listing 1).

Die Überwachung findet alle fünf Minuten statt. Wenn die Abfrage sehr lange läuft, ist die XML-Datei wahrscheinlich schon sehr groß geworden. Man kann sie entweder manuell löschen (nachdem sie eventuell vorher gesichert wurde). Alternativ kann auch Oracle die Datei löschen, wenn man Instance- und Datenbank-Name entsprechend ersetzt (siehe Listing 2).

2. Listener.log überwachen

Ab Version 11.2.0.2 lässt sich die „listener.log"-Datei (XML-Variante) mittels „SELECT" auswerten (siehe Listing 3).

Die Überwachung erfolgt alle fünf Minuten. Dieser Ansatz funktioniert nur, wenn der Listener seine „listener.log" und „log.xml" in die „diag"-Struktur schreibt. Zuständig dafür ist der „listener.ora"-Parameter: „DIAG_ADR_ENABLED_<listener_name>= ON|OFF". Wenn man ein anderes Verzeichnis angibt, landen die Daten nicht mehr in der internen „diag"-Struktur: „ADR_BASE_< listener_name > = <pfad >".

3. Backups überwachen

Man sollte täglich prüfen, ob die letzten Backups gelaufen sind. Wenn nicht, ist das Backup zu wiederholen (siehe Listing 4). Die Überwachung geschieht einmal am Tag.

4. Welche Datenbank-Objekte invalide sind

Man bekommt für jedes Schema maximal eine Zeile nur mit der Anzahl der ungültigen Objekte angezeigt. Wenn keine Zeile zurückkommt, ist auch kein Objekt defekt (siehe Listing 5). Das Zeit-Intervall für die Überwachung ist einmal am Tag.

5. Tabellen/Dateien mit Block-Korruption

Bei dieser Abfrage sollte dreimal „0" zurückkommen (siehe Listing 6). Ansonsten sollte man die Views jeweils

```
SELECT
to_char(originating_timestamp,'DD.MM.YYYY Hh24:MI:SS') as log_date, message_text
FROM v$diag_alert_ext
WHERE trim(component_id) = 'rdbms'
AND (message_text like '%ORA-%'
OR  message_text like '%TNS-%'
OR  message_text like '%Checkpoint not%')
AND originating_timestamp > sysdate-interval '1' hour;
```

Listing 1

```
dos/unix> adrci exec= „set home diag/rdbms/o12c/o12c; purge -age 0 -type alert"
```

Listing 2

```
SELECT
to_char(originating_timestamp,'DD.MM.YYYY Hh24:MI:SS') as log_date,message_text
FROM v$diag_alert_ext
WHERE trim(component_id) ='tnslsnr'
AND  message_text like '%TNS-%'
AND originating_timestamp>sysdate-interval '1' hour;
```

Listing 3

```
SELECT start_time, to_char(end_time,'HH24:MI:SS') end_time, elapsed_seconds as elap_sec,input_
type, status,output_device_type as out_device,
round(input_bytes/1024/1024/1024) INPUT_GB, round(output_bytes/1024/1024/1024) OUTPUT_GB
FROM v$rman_backup_job_details r
ORDER BY 1;
```

Listing 4

```
SELECT o.*,i.ind_i as "Ind (I)"  FROM (
SELECT o.owner,
count( CASE WHEN o.object_type= 'TABLE'  AND o.status<>'VALID'  THEN 'x' END ) as "TAB (I)",
count( CASE WHEN o.object_type= 'VIEW'   AND o.status<>'VALID'  THEN 'x' END ) as "Views (I)",
count( CASE WHEN o.object_type= 'CLUSTER'  AND o.status<>'VALID'  THEN 'x' END ) as "Clus
(I)",
count( CASE WHEN o.object_type= 'TYPE'  AND o.status<>'VALID'  THEN 'x' END ) as "Type (I)",
count( CASE WHEN o.object_type= 'SYNONYM' AND o.status<>'VALID'  THEN 'x' END ) as "Syn (I)",
count( CASE WHEN o.object_type= 'PACKAGE BODY'  AND o.status<>'VALID'  THEN 'x' END ) as
"PackB (I)",
count( CASE WHEN o.object_type= 'PROCEDURE'     AND o.status<>'VALID'  THEN 'x' END ) as "Proc
(I)",
count( CASE WHEN o.object_type= 'TRIGGER'       AND o.status<>'VALID'  THEN 'x' END ) as "Trig
(I)",
count( CASE WHEN o.object_type= 'FUNCTION'      AND o.status<>'VALID'  THEN 'x' END ) as "Func
(I)"
FROM all_objects o
GROUP BY o.owner) o,
(SELECT owner,
count( CASE WHEN status NOT IN ('VALID','N/A') THEN 'x' END ) as ind_i
FROM all_indexes i GROUP BY owner) I
WHERE o.owner=i.owner
AND "TAB (I)" + "Views (I)" +"Clus (I)" + "Type (I)" + "Syn (I)" + "PackB (I)" +  "Proc (I)"
+ "Trig (I)" + „Func (I)" + i.ind_i>0;
```

Listing 5

```
SELECT
(SELECT count(*) FROM gv$backup_corruption) as BAK_CORRUPT,
(SELECT count(*) FROM gv$copy_corruption) AS COPY_CORRUPT,
(SELECT count(*) FROM gv$database_block_corruption) AS DB_BLOCK_CORRUPT
FROM dual;
```

Listing 6

```
SELECT instance_name, host_name, startup_time, status, logins
FROM gv$instance
ORDER BY 1;
```

Listing 7

```
SELECT con_id, name, open_mode
FROM v$pdbs
ORDER BY 1;
```

Listing 8

```
SELECT
round((space_limit)/1024/1024/1024,2) “Max Space (GB)” ,round((space_limit-space_
used)/1024/1024/1024,2) “Free Space (GB)”
FROM v$recovery_file_dest;
```

Listing 9

```
SELECT username, account_status, lock_date, expiry_date
FROM dba_users
WHERE username NOT IN (SELECT distinct schema_name FROM v$sysaux_occupants)
AND username NOT IN (‘APPQOSSYS’,‘FLOWS_FILESv,‘XS$NULL’,‘ORACLE_OCM’,
‘OUTLN’);
```

Listing 10

```
SELECT username, account_status, lock_date, expiry_date
FROM dba_users
WHERE (oracle_maintained = ‘N’
OR username IN (‘SYS’,‘SYSTEM’))
AND account_status <> ‘OPEN’;
```

Listing 11

```
SELECT min(first_time) as start_time, sum(“<1min”), sum(“1min<x<5min”), sum(“5min<x<10min”),su
m(“10min<x<60min”),sum(“>60min”) FROM (SELECT
 first_time,
 case when (round((first_time-lag(first_time,1)over ( order by sequence#))*24*60))<=1 then 1
end as “<1min”,
 case when (round((first_time-lag(first_time,1)over ( order by sequence#))*24*60))>1 AND
    (round((first_time-lag(first_time,1)over ( order by sequence#))*24*60))<=5 then 1 end as
“1min<x<5min”,
 case when (round((first_time-lag(first_time,1)over ( order by sequence#))*24*60))> 5 AND
    (round((first_time-lag(first_time,1)over ( order by sequence#))*24*60))<=10 then 1 end as
“5min<x<10min”,
 case when (round((first_time-lag(first_time,1)over ( order by sequence#))*24*60))> 10 AND
    (round((first_time-lag(first_time,1)over ( order by sequence#))*24*60))<=60 then 1 end as
“10min<x<60min”,
 case when (round((first_time-lag(first_time,1)over ( order by sequence#))*24*60))>60 then 1
end as “>60min”
FROM v$loghist
WHERE first_time > (sysdate - 1));
```

Listing 12

```
SELECT tum.tablespace_name,round(tum.used_space*t.block_size/1024/1024/1024,2) used_GB,
round(tum.tablespace_size*t.block_size/1024/1024/1024,2) max_possible_size_gb,round(used_per-
cent,2) as used_percent
FROM dba_tablespace_usage_metrics tum, dba_tablespaces t
WHERE tum.tablespace_name = t.tablespace_name;
```

Listing 13

nach den Details durchforsten. Die Überwachung erfolgt einmal am Tag.

6. Zustand der Datenbank allgemein
Listing 7 zeigt, ob die Instanz hochgefahren wurde und Anmeldungen der Benutzer möglich sind (normale Datenbank/RAC). Für die Pluggable Database (ab 12.1) gilt Listing 8. Alle fünf Minuten findet diese Überwachung statt.

7. Freier Speicherplatz in der Fast Recovery Area
Falls die Fast Recovery Area für die Instanz zum Einsatz kommt, sollte man regelmäßig prüfen, ob dort noch genug freier Speicherplatz zur Verfügung steht (siehe Listing 9). Falls nicht, sind nur mit RMAN alte Dateien zu löschen oder alternativ die Fast Recovery Area zu vergrößern. Dies wird alle fünf Minuten überwacht.

8. Gesperrte und abgelaufene Accounts
Listing 10 zeigt (bis Version 12.1) alle Accounts, die derzeit gesperrt sind. Damit nicht die Oracle-eigenen, bereits vorgesperrten Accounts mit angezeigt werden, sollte man diese herausfiltern. Ab Version 12.1 lässt sich die Ausschluss-Liste abhängig von der Datenbank-Version erweitern (siehe Listing 11). Die Überwachung erfolgt jede Stunde.

9. Redo-Log-Switches
Dieser Select zeigt für die letzten vierundzwanzig Stunden an, wie oft Log-Switches unter einer Minute, zwischen einer und fünf Minuten, zwischen fünf und zehn Minuten, zwischen zehn und sechzig Minuten und über sechzig Minuten aufgetreten sind (siehe Listing 12). Ideal wäre ein Bereich zwischen zehn und sechzig Minuten. Wenn die Datenbank darunter liegt, kann man sich überlegen, die Größe der Redo-Log-Datei zu verändern. Das Zeit-Intervall für die Überwachung ist jede Stunde.

10. Wachstumsmöglichkeiten der Tablespaces
Der SELECT zeigt den aktuellen Füllpegel des Tablespace und die maximale Größe, die er laut seiner Einstellung erreichen könnte (siehe Listing 13). Ob die Platten diesen Platz wirklich zur Verfügung stellen könnten, wird jedoch nicht erkannt. Die Überwachung erfolgt alle fünf Minuten.

Hinweis: Eine digitale Version dieses Artikels lässt sich unter http://www.muniqsoft.de/tipps/publikationen.htm herunterladen.

Marco Patzwahl
m.patzwahl@muniqsoft.de

„Lizenz-Compliance wurde bisher in vielen Unternehmen nicht mit der Priorität behandelt, die notwendig wäre …“

Es ist nicht immer leicht, die Lizenzbedingungen von Oracle zu verstehen. Deshalb hat sich der Arbeitskreis Lizenzierung unter der Leitung von Michael Paege, stellv. Vorstandsvorsitzender und Leiter Competence Center Lizenzierung der DOAG, dem Thema angenommen. Entstanden ist ein kompakter, digitaler Lizenzguide, in dem die wichtigsten Aspekte der Lizenzierung von Oracle-Software zusammengefasst sind. Diesen können DOAG-Mitglieder unter www.doag.org/go/lizenzguide kostenfrei bestellen.
Für Michael Paege ist das Hauptthema nach wie vor die Lizenzierung bei Virtualisierung mit VMware. Vor drei Jahren hat die DOAG ein Presse-Roundtable zu diesem Thema veranstaltet. Danach fanden auch Gespräche zwischen der DOAG und dem Chief Customer Relation Officer des Oracle Headquarters statt. Leider ist es nicht gelungen, in dem gewünschten Maß bei Oracle nachhaltige Verbesserungen zu erreichen.
Im Hinblick auf den Themenkomplex „Virtualisierung“ sowie die Metrik „Named User Plus“ in Verbindung mit Datentransfers wird die Lizenz-Compliance leider in vielen Unternehmen nicht mit der Priorität behandelt, die notwendig wäre. Bei Audits werden dann viele Firmen auf dem falschen Fuß erwischt. Viele Unternehmen – vor allem, wenn sie sich ursprünglich für die Metrik Named-User-Plus (NUP) entschieden hatten – sind heutzutage nicht mehr korrekt lizenziert. Mit der Zeit haben sie Architektur-Änderungen vorgenommen, neue Lösungen eingeführt, und auch angefangen, Daten mit anderen Systemen und der Außenwelt auszutauschen. Die Idee hinter dem Lizenzguide ist es, Transparenz zu schaffen. Die Informationen zur Lizenzierung musste man sich bisher mühsam aus den jeweiligen Verträgen und auf den Oracle-Webseiten zusammensuchen. Jetzt gibt es zum ersten Mal ein kompaktes, übersichtliches Dokument, in dem die wichtigsten Informationen zu finden sind.

Mit Ops Center fest im Blick: Hardware- und OS-Monitoring für Oracle Server und Solaris

Elke Freymann, ORACLE Deutschland B.V. & Co. KG

Das beste Überwachungstool für Hard- oder Software sollte der Hersteller selber liefern, denn schließlich sitzt er ja an der Quelle der notwendigen Informationen. Natürlich hängt damit auch die Messlatte für die zu erfüllenden Ansprüche recht hoch; wenn es aber um die Überwachung von Oracle-Sun-Server-Hardware und das Betriebssystem Solaris geht, muss Oracle Enterprise Manager Ops Center 12c sich mit seinen Fähigkeiten keinesfalls verstecken.

Für die Hardware ist es fast schon zwingend notwendig: Wenn Oracle die inneren Werte des Servers nicht überwachen kann, wer dann? Aber auch im Bereich „Solaris-Monitoring" bietet Ops Center Funktionen, die andere Tools so nicht zur Verfügung stellen. Ganz unschlagbar: ein Oracle-Server, auf dem Solaris läuft. Da kann Ops Center noch weiteres Zusatzwissen in die Monitoring-Aufgabe mit einbringen.

Wer hier wen überwacht

Enterprise Manager Cloud Control und Enterprise Manager Ops Center sind zwei eigenständige Produkte mit Management-Aufgaben im weitesten Sinne. Ops Center konzentriert sich dabei auf das Infrastruktur-Management (Hardware- und Betriebssystem-Schichten) sowie Virtualization Management mit Fokus auf Solaris-Zonen und Oracle VM Server for SPARC – historisch bedingt als Logical Domains bekannt. Jedes dieser beiden Produkte der Enterprise Manager Suite hat auch seine eigene Architektur. Abbildung 1 zeigt diese für Ops Center.

Bei der Installation von Ops Center wird zunächst der sogenannte „Enterprise Controller" samt seinem Data Repository eingerichtet. Dieses Data Repository, eine Oracle-Datenbank, wird entweder lokal auf dem Management-Server mit installiert oder „remote" auf einem anderen Server eingerichtet. Als nächste, serverseitig notwendige Software-Komponente wird mindestens ein sogenannter „Proxy Controller" installiert. Und genau dieser ist derjenige, der die eigentliche Arbeit übernimmt, also auch zum Beispiel die Monitoring-Daten von den überwachten Objekten einsammelt und an den Enterprise Controller weiterleitet.

Der Enterprise Controller füllt damit das Data Repository, bringt die Daten in der grafischen Benutzeroberfläche zur Anzeige, löst Alarmierungen aus und sorgt bei entsprechenden Konfigurations-Einstellungen und Internet-Zugang dafür, dass bei gemeldeten Problemen auch Service Requests in My Oracle Support eröffnet werden.

Der Proxy Controller hat, je nachdem, was für einen Objekt-Typ er überwacht, unterschiedliche Kommunikations-Mechanismen zur Verfügung. Mit einem Hardware-Objekt

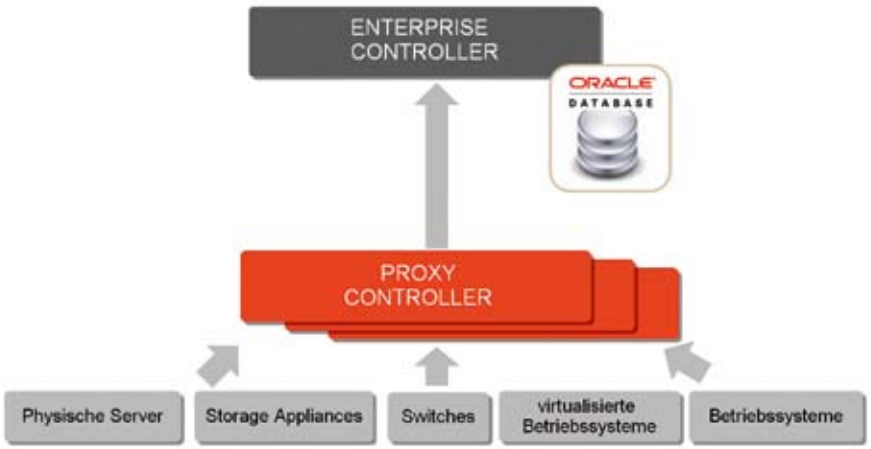


Abbildung 1: Architektur-Überblick Enterprise Manager Ops Center

– genauer dem Service-Prozessor eines Servers, einer ZFS Storage Appliance oder eines Ethernet- beziehungsweise InfiniBand-Switch – werden genau diejenigen Protokolle, die die Hardware nach außen hin anbietet, angesprochen. Typischerweise sind das IPMI und „ssh". Hardware, die dabei verstanden wird, muss vom Hersteller Oracle oder ehemals Sun Microsystems stammen.

Soll ein Software-Objekt überwacht werden, im Falle von Ops Center eine Betriebssystem-Instanz, egal ob „bare metal" oder virtualisiert installiert, so hat Ops Center die Wahl: Sind nur reine Überwachsaufgaben wahrzunehmen, so kann ein „agentless monitoring" erfolgen. Der Proxy Controller sammelt dann per „ssh login" die relevanten Daten in zyklischen Abständen ein.

Sollen aber auch aktive Management-Aufgaben wie Virtualization-Management oder Patching-Aufgaben inklusive der Erstellung von Reports über Paketstände auf dem überwachten Betriebssystem erledigt werden, so wird ein installierter Ops-Center-Agent benötigt. Dieser hält dann die Verbindung mit dem Proxy und empfängt zum Beispiel auch auszuführende Jobs über diesen Kanal.

Überwachung der Oracle-Hardware

Für die reine Hardware-Überwachung wendet sich der Proxy Controller direkt an den Service-Prozessor der Maschine und bietet ein komplettes Monitoring mit dem gleichen Abdeckungsgrad, den auch der Service-Prozessor nativ selbst liefert. Das Ganze erfolgt jedoch zentralisiert: Von der graphischen Benutzeroberfläche aus hat man alle überwachten Server im Blick – und nicht nur einen individuellen (siehe Abbildung 2).

Da es sich um Hardware aus dem eigenen Haus handelt, ist Ops Center auch in der Lage, Hardware-nahe Aufgaben zu erledigen:

- Firmware-Updates für den Service-Prozessor
- Firmware-Updates für verbaute Server-Komponenten
- Klassisches Lights-Out-Management, also Power off/Power on für den eigentlichen Server und Zugriff auf die serielle Konsole des Betriebssystems

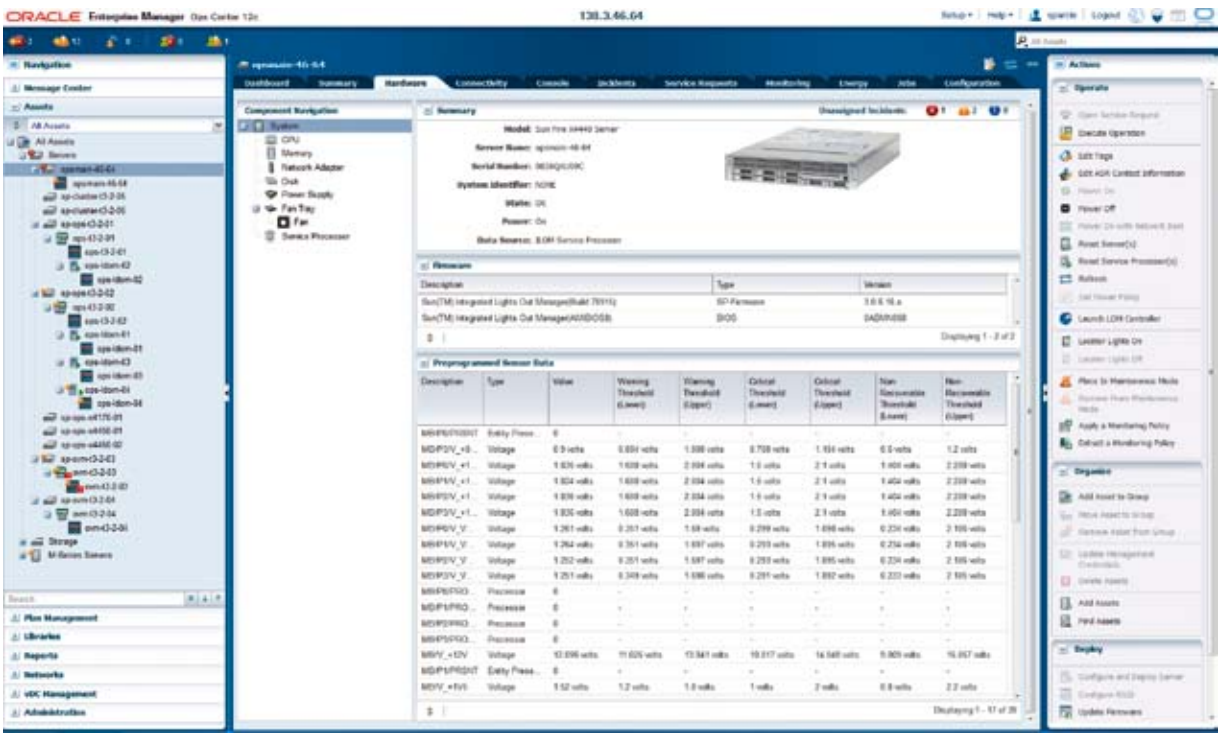


Abbildung 2: Hardware-Monitoring

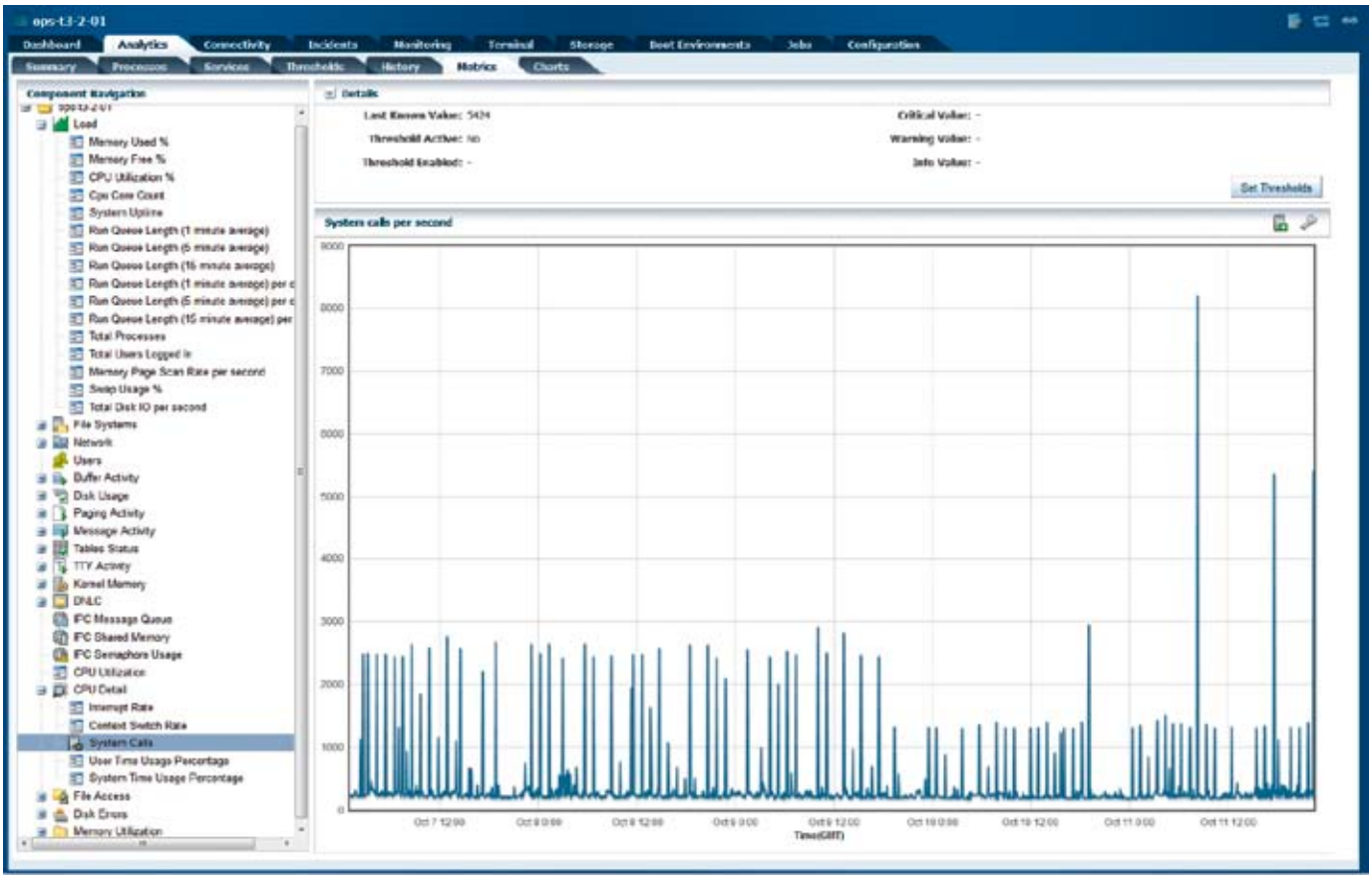


Abbildung 3: Anzahl System Calls als Beispiel für einen Metrikwert

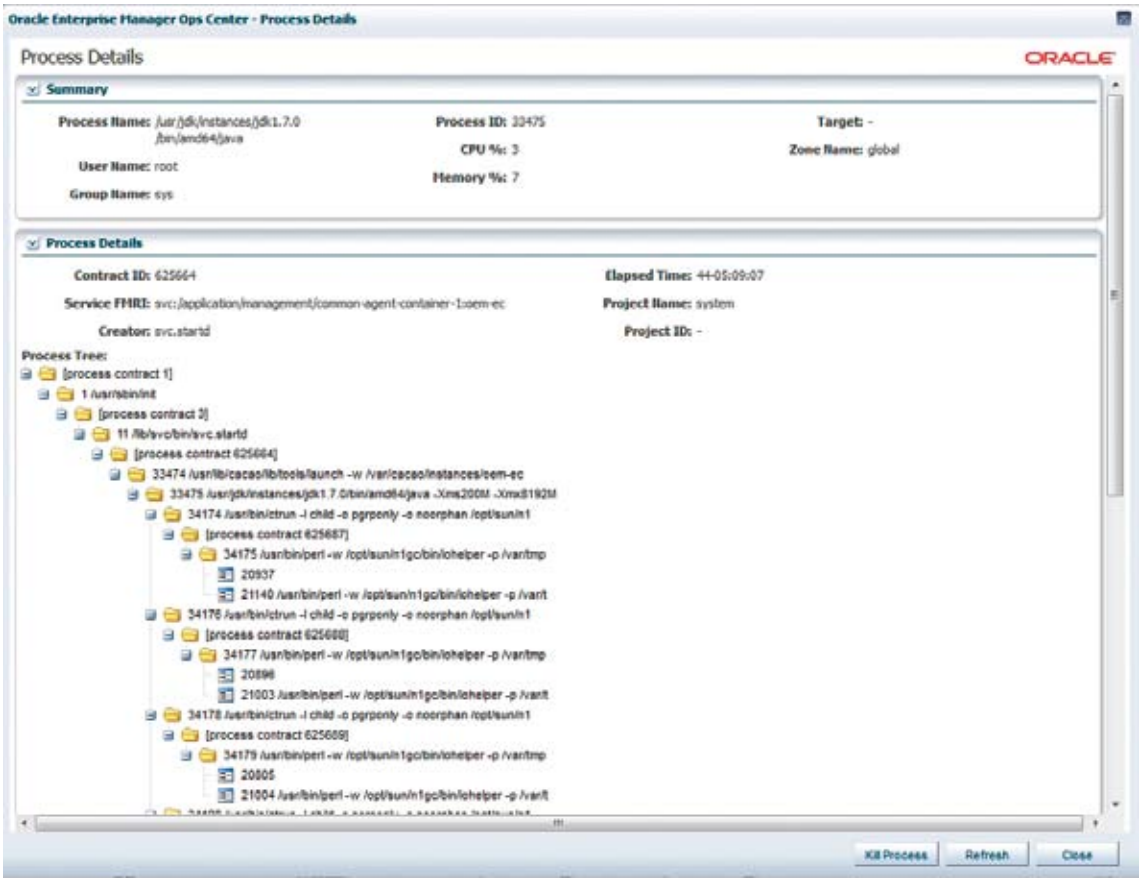


Abbildung 4: Detail-Ansicht für einen Ops-Center-Prozess

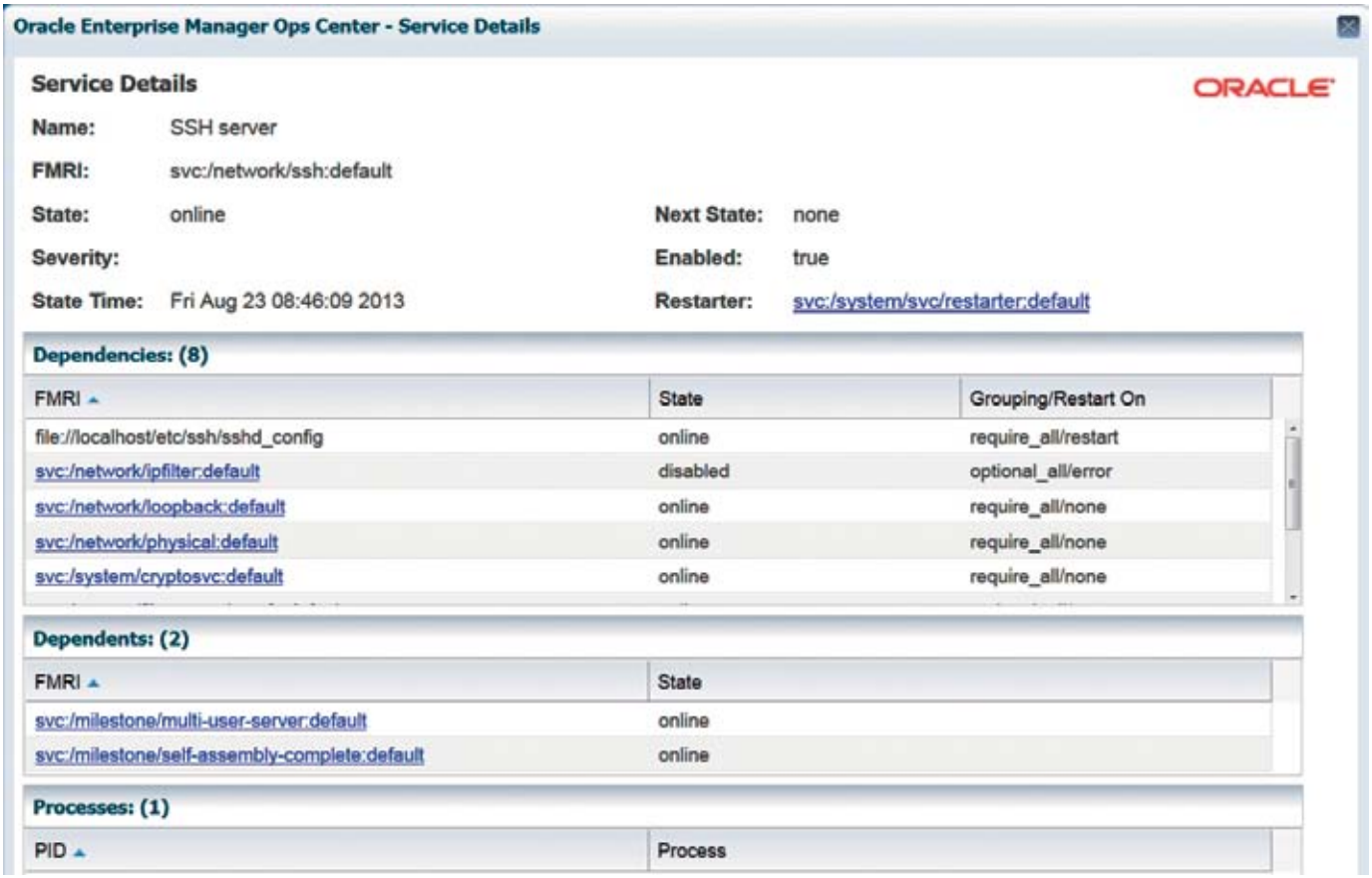


Abbildung 5: SSH-Server-Service in Detail-Ansicht

Überwachung von Solaris

Zunächst bietet Ops Center natürlich die ganz klassische Überwachungsmöglichkeit für Kennzahlen, sogenannte Metriken, des Betriebssystems. Diese Metriken decken die folgenden Funktionsbereiche ab:

- Load
- File Systems
- Networks
- Users
- Buffer Activity
- Disk Usage
- Paging Activity
- Message Activity
- Tables Status
- TTY Activity
- Kernel Memory
- DNLC
- IPC Message Queue
- IPC Shared Memory
- IPC Semaphore Usage
- CPU Detail
- File Access
- Disk Errors
- Memory Utilization

Neben den reinen Metrikwerten ist auch immer eine Ansicht aller laufenden Prozesse interessant. Ops Center stellt diese mit den Parametern „Bezeichnung“, „Benutzer“, „Status“, „CPU- und Memory-Verbrauch“ in einer Liste dar, die sortiert und durchsucht werden kann (siehe Abbildung 3).

Ist ein spezieller Prozess von Interesse, so können seine Details inspiziert werden: der „Process Tree“, „Thread-Informationen“, „Handles“, „Aufruf-Parameter“ und Details zur Memory-Belegung sind abrufbar (siehe Abbildung 4).

Für einen schnellen Überblick werden Top-Consumer nach dem Bereichen CPU-, Memory-, Netzwerk- und I/O-Auslastung auch noch in einer separaten Übersicht zusammengestellt. Um darüber hinaus dem Anspruch gerecht zu werden, Spezifika von Solaris zu kennen, ist in Ops Center zum Beispiel die Überwachung der Solaris-Services implementiert.

Alle Services werden in einer ähnlichen Liste wie die Prozesse dargestellt. Parameter, die ausgewertet werden, sind „Service Name“, „Identifier“, „Startzeitpunkt“ und ganz wichtig der aktuelle Status des Service.

In Ops Center bereits von Haus aus eingebaute Monitoring-Regeln überwachen diesen Status und lösen eine Alarmierung aus, wenn ein Service nicht mehr laufen sollte.

Wer sich für die Details eines Service interessiert und sich zum Beispiel per Klick auf einen Link das Logfile des Service anzeigen lassen will, wird in der Detail-Ansicht fündig (siehe Abbildung 5).

Außerdem arbeitet Ops Center mit der „Fault Management Architecture“ von Solaris zusammen: So werden zusammengehörige Fehlermeldungen auf Hardware- und Betriebssystem-Ebene zu geeigneten Alarmierungen zusammengefasst und liefern die bestmögliche Überwachung für die Einheit aus Server und Betriebssystem.

Kommen Server-Virtualisierungs-Technologien wie Logical Domains oder Solaris-Zonen zum Einsatz, ver-

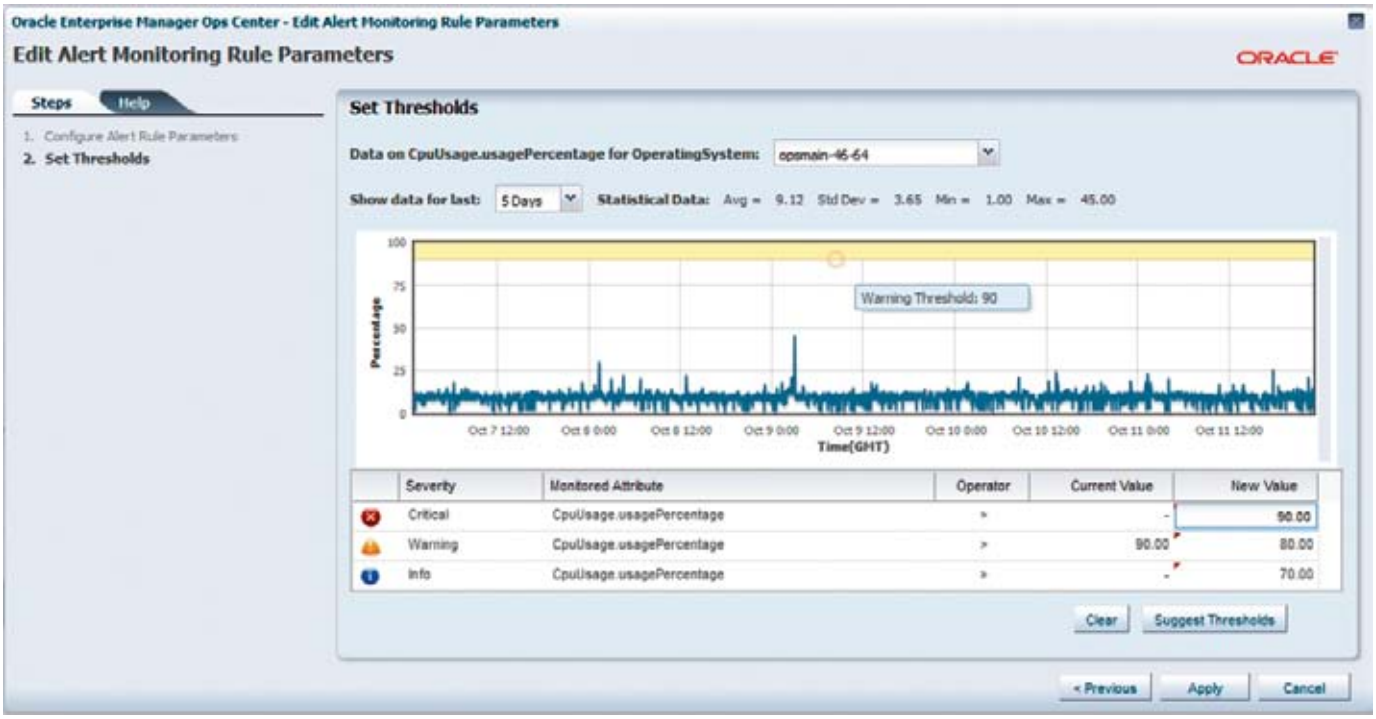


Abbildung 6: Smart Thresholding zur Überwachungsregel „CPU-Auslastung“

schafft Ops Center einen klaren Überblick über die Zuordnung der virtualisierten Instanzen zu den physischen Servern und stellt Funktionen zum „Virtualization Management“ bereit. Ansichten, die den Ressourcen-Verbrauch der virtualisierten Instanzen im Vergleich ausweisen, bieten weitere Übersichtsmöglichkeiten.

Historische Daten zu überwachten Betriebssystem-Parametern und Gesamtkurven für die CPU-, Memory-, Dateisystem- und Netzwerk-Auslastung sowie die komplette System Load werden gesammelt, angezeigt und können exportiert werden.

Noch etwas kann Ops Center gut im Blick behalten: Auf welchem der Server fehlen zum Beispiel noch empfohlene Patches für Solaris 10? Auf welchem Server sollte noch ein Firmware-Upgrade für den Service-Prozessor durchgeführt werden, denn dieser Server weicht von der Standardvorgabe ab, die für diesen Typ getroffen wurde?

Solche Überwachungsaufgaben erledigt Ops Center, wenn man entsprechende Abfrage-Reports definiert und nach einem festgelegten Zeitplan zyklisch ablaufen lässt.

Monitoring-Rules und Policies

Out-of-the-box bringt Ops Center eine ganze Reihe vordefinierter Überwachungsregeln mit, die sogenannten „Alert Monitoring Rules“. Bestandteile einer solchen Monitoring-Policy sind zusammengehörige Überwachungsregeln. Zusammengehörig heißt in diesem Fall: Dieser Regelsatz passt zum Beispiel für die Hardware-Überwachung eines M-Klasse-Servers oder für die Überwachung einer Instanz von Solaris. Mit einer Monitoring-Policy ist immer auch die Information verknüpft, auf welchen Typ der überwachten Objekte dieser Regelsatz passt.

Die vordefinierten Regeln kann man, wenn es um den Bereich der Software-Überwachung geht, modifizieren und so auf die individuellen Bedürfnisse anpassen. Regeln, die Hardware im Auge behalten, lassen sich nicht ändern. Aber in beiden Bereichen – Hard- und Software – lassen sich die Monitoring-Policies um weitere, selbst definierte Regeln erweitern, oder Regeln können auch aus einer Policy entfernt werden.

Die angepassten Policies werden dann unter eigenem Namen abgelegt und können als anzuwendender Default für Objekte passenden Typs deklariert werden, die man neu ins Ops Center einhängt. Man kann natürlich die modifizierten Policies auch auf die Server oder Server-Gruppen ausrollen, die bereits im Ops Center bekannt sind.

Definiert man eine eigene Regel, so gibt man ganz klassisch den zu überwachenden Parameter und die Schwellwerte verschiedener Abstufung an („critical“, „warning“, „info“), bei deren Überschreitung eine entsprechende Alarmierung erfolgen soll. Für die Festlegung der Schwellwerte ist das sogenannte „smart thresholding“ von Ops Center hilfreich: Sofern historische Werte für den betreffenden Überwachungsparameter vorhanden sind, werden diese dargestellt und man bekommt einen Anhaltspunkt dafür, wie man den Schwellwert definieren will (siehe Abbildung 6).

Außerdem gibt man bei der Definition einer Alert-Monitoring-Rule an, wie lange der entsprechende Schwellwert überschritten sein muss, damit ein Alarm ausgelöst wird und welche „immediate action“ – also welches Skript – ausgeführt werden soll, wenn der Alarm auftritt.

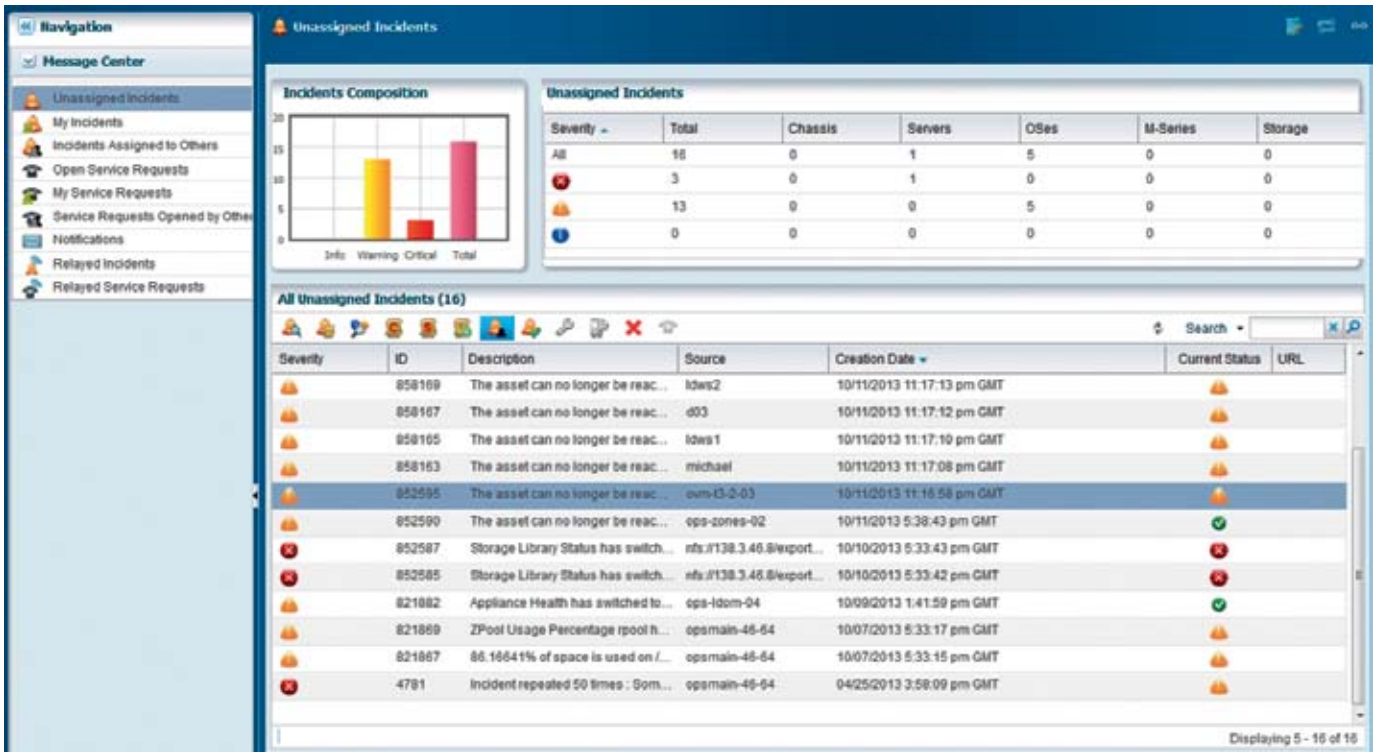


Abbildung 7: Incidents im Message-Center

Incidents und das Message Center

Kommt es zu einer Alarmierung, wird die „immediate action" ausgeführt. Benutzer, die ein entsprechendes Notification Profile zugewiesen bekommen haben, erhalten eine Benachrichtigung per E-Mail. Außerdem wird das entdeckte Problem auch als Incident mit einem Status und Angaben zum zugeordneten Bearbeiter verwaltet.

Offene Incidents kommen sowohl auf Ebene der betroffenen Einheit (zum Beispiel Hardware-Eintrag für einen Server oder Betriebssystem-Schicht) als auch in einem zentralen Message-Center, das systemweit alle entsprechenden Informationen sammelt, zur Anzeige und Bearbeitung (siehe Abbildung 7).

Fazit

Ops Center ist das vom Hersteller empfohlene Tool, um Oracle-Server und das Betriebssystem Solaris optimal zu überwachen. Anfallende Lizenzkosten stellen auch keine Hürde für den Einsatz dar: Die Nutzung und die Wartung (Service-Anspruch) von Ops Center sind kostenfrei im Oracle Premier Support enthalten.

Weiterführende Informationen

- Zentrale Quelle für Ops Center 12c Informationen im OTN: http://www.oracle.com/technetwork/oem/ops-center/index.html
- Dokumentation mit How-to-Guides: http://docs.oracle.com/cd/E27363_01/index.htm
- Ops-Center-Everywhere-Programm: http://www.oracle.com/us/corporate/features/opscenter-everywhere-program-1567667.html
- Download über OTN: http://www.oracle.com/technetwork/oem/ops-center/oem-ops-center-188778.html

Elke Freymann
elke.freymann@oracle.com

Neu: MySQL Cluster 7.3 GA

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

Seit Juni 2013 steht MySQL Cluster 7.3 als Produktions-Release zur Verfügung. Zu den großen Neuheiten zählen eine „node.js"-NoSQL-Schnittstelle, eine grafische Benutzeroberfläche für den Auto-Installer sowie die Unterstützung von Fremdschlüsseln. Damit können Anwender mit den schnellen Innovationszyklen bei Web-, Cloud-, sozialen und mobilen Diensten weiter Schritt halten.

MySQL Cluster ist eine skalierbare, ACID-konforme, transaktionale Echtzeitdatenbank, die 99,999 Prozent Verfügbarkeit mit den niedrigen Gesamtbetriebskosten einer Open-Source-Lösung verbindet. Sie basiert auf einer verteilten Multi-Master-Architektur ohne singuläre Fehlerquellen und wird horizontal auf Standard-Hardware skaliert, um lese- und schreibintensive Arbeitslasten mit Auto-Sharding zu verarbeiten.

Ursprünglich als eingebettete Telekommunikations-Datenbank für netzinterne Anwendungen entworfen, die Carrier-Grade-Verfügbarkeit und Echtzeitleistung erfordern, wurde MySQL Cluster schnell um neue Funktionen erweitert, die es unter anderem für lokale oder in der Cloud bereitgestellte Web-, Mobil- und Unternehmens-Anwendungen einsetzbar machen. Die neuen Funktionen von MySQL Cluster betreffen folgende Bereiche:

- Automatische Installation von MySQL Cluster
- Fremdschlüssel
- JavaScript-Treiber für „Node.js"
- Leistungsverbesserungen
- MySQL 5.6: Erweiterungen der Replikation
- MySQL Cluster Manager: Zentralisierte Sicherung und Wiederherstellung

Automatische Installation von MySQL Cluster

Hohe Priorität hat in diesem Release die wesentlich einfachere und schnellere Bereitstellung eines Clusters, der genau auf die Anwendung und Umgebung des Nutzers zugeschnitten ist – dieser soll sich darauf konzentrieren, die Vorteile von MySQL Cluster in der Anwendung zu nutzen, statt darüber nachdenken zu müssen, wie die Datenbank installiert, konfiguriert und gestartet werden kann. Die gesamte Steuerung der automatischen Installation erfolgt über eine einfach zu bedienende Web-Anwendung und führt durch folgende Schritte:

- Eingabe der voraussichtlichen Art der Arbeitslast (beispielsweise „Echtzeitanwendung mit vielen Schreibvorgängen") zusammen mit der Liste der Hosts, auf denen der Cluster ausgeführt werden soll.
- Das Installationsprogramm stellt automatisch Verbindungen mit den einzelnen Hosts her, um die verfügbaren Ressourcen zu ermitteln (Betriebssystem, Arbeitsspeicher und CPU-Kerne).
- Dann wird eine Topologie vorgeschlagen (also welche Knoten/Prozesse den Cluster bilden und auf welchen Hosts sie ausgeführt werden sollten). Der Benutzer kann diese Topologie akzeptieren oder verändern.
- Basierend auf allen bereitgestellten und ermittelten Informationen werden Konfigurations-Einstellungen vorgeschlagen – auch diese kann der Benutzer akzeptieren oder verändern.
- Zum Schluss kann das Installationsprogramm optional den Cluster bereitstellen und starten sowie den Status der Knoten anzeigen, während der Cluster in Betrieb genommen wird.

Fremdschlüssel

MySQL Cluster 7.3 unterstützt nativ die Nutzung von Fremdschlüsseln. Das Entwurfsziel war, die Implementierung von Fremdschlüsseln in InnoDB möglichst genau nachzustellen. So können Benutzer leichter zu MySQL Cluster übergehen, wenn dies sinnvoll ist, und ihre vorhandenen Kenntnisse der MySQL-Entwicklung bei der Erstellung neuer Anwendungen weiter nutzen. Auf einige Ausnahmen sei hier hingewiesen:

- InnoDB unterstützt keine „No-Action"-Einschränkungen; MySQL Cluster unterstützt diese.
- InnoDB kann das Erzwingen von Fremdschlüssel-Einschränkungen mit dem Parameter „FOREIGN_KEY_CHECKS" aussetzen; MySQL Cluster ignoriert diesen Parameter zurzeit.
- Man kann keine Fremdschlüssel zwischen zwei Tabellen einrichten, wenn die eine mit MySQL Cluster gespeichert ist und die andere mit InnoDB.
- Primärschlüssel für MySQL-Cluster-Tabellen lassen sich nicht über Fremdschlüssel-Einschränkungen ändern.
- In MySQL Cluster sind Fremdschlüssel-Einschränkungen für partitionierte Tabellen zulässig.

JavaScript-Treiber für „Node.js"

„Node.js" ist eine Plattform, mit der schnelle, skalierbare Netzwerk-Anwendungen (normalerweise Web-Anwendungen) mit JavaScript entwickelt werden können. „Node.js" ist dafür entworfen, dass ein einzelner Thread Millionen von Client-Verbindungen in Echtzeit bedienen kann. Dies wird durch eine asynchrone, ereignisgesteuerte Architektur erreicht. Es entspricht der Funktionalität von MySQL Cluster, sodass die beiden Lösungen hervorragend zusammenpassen.

Der NoSQL-Treiber von MySQL Cluster für „Node.js" wurde als Modul für die V8-Engine implementiert. Er bietet eine native, asynchrone JavaScript-Schnittstelle für „Node.js", über die Abfragen direkt an MySQL Cluster gesendet und Ergebnismengen empfangen werden können, ohne Transformationen in SQL. Zusätzlich kann man auch festlegen, dass der Treiber SQL verwendet, sodass dasselbe API für InnoDB-Tabellen verwendet werden kann.

Mit dem JavaScript-Treiber von MySQL Cluster für „Node.js" können Architekten JavaScript vom Client auf dem Server wiederverwenden, bis hin zu einer verteilten, fehlertoleranten, transaktionalen Datenbank. Dies unterstützt hoch skalierbare Echtzeitdienste:

- Verarbeitung von Streaming-Daten aus digitalen Werbungs- und Benutzerverfolgungs-Systemen
- Spiele und soziale Netzwerkseiten – als Grundlage der Back-End-Infrastruktur von Diensten für mobile Geräte

JavaScript ist mit „Node.js" Teil eines wachsenden Portfolios von NoSQL-APIs für MySQL-Cluster, zu dem bereits „Memcached", „Java", „JPA" und „HTTP/REST" gehören. Natürlich können Entwickler weiterhin SQL nutzen, um komplexe Abfragen auszuführen und das umfangreiche Umfeld von Konnektoren, Frameworks, Hilfswerkzeugen und Kompetenzen zu nutzen.

Leistungsverbesserungen

MySQL Cluster 7.3 baut auf den enormen Verbesserungen der Leistung und Skalierbarkeit in MySQL Cluster 7.2 auf. In diesem Release wurde zwei Bereichen besondere Beachtung geschenkt. Der erste ist die Verbesserung des Durchsatzes jeder Anwendungsverbindung zur Datenbank, mit dem noch höhere Skalierbarkeit ermöglicht wird. Der zweite Bereich ist die Verarbeitung von SQL-Abfragen auf dem MySQL-Server, mit der Abfragen beschleunigt und die Last der Cluster-Knoten verringert werden können, was wiederum den Durchsatz erhöht.

Die Leistungsfähigkeit von MySQL Cluster wird besonders deutlich, wenn so viele parallele Vorgänge angeboten werden wie möglich. Um dies zu erreichen, sollte Parallelität auf jeder Ebene konfiguriert werden: Es sollten mehrere Anwendungs-Threads Arbeit an den MySQL-Server (oder das sonstige API) senden und es sollte mehrere MySQL-Server sowie mehrere Verbindungen zwischen dem MySQL-Server (oder Knoten des sonstigen API) und den Datenknoten geben.

Jede Verbindung zu den Datenknoten verbraucht eine der 256 verfügbaren Knoten-IDs. In einigen Szenarien kann dies eine Grenze für die Skalierbarkeit des Clusters darstellen. In MySQL Cluster 7.3 ist der Durchsatz über jede dieser Verbindungen wesentlich höher. Dies bedeutet, dass weniger Verbindungen (und somit weniger Knoten-IDs) erforderlich sind, um dieselbe Arbeitslast zu bewältigen. Dies wiederum bedeutet, dass dem Cluster mehr API-Knoten und Datenknoten hinzugefügt werden können, um die Kapazität und Leistung noch weiter zu skalieren. Vergleichstests haben gezeigt, dass der Durchsatz pro Verbindung bis zu sieben Mal höher ist.

MySQL 5.6: Erweiterungen der Replikation

Die native Integration mit MySQL 5.6 ermöglicht es, InnoDB- und MySQL-Cluster-Speicher-Engines innerhalb einer MySQL-5.6-basierten Anwendung zu kombinieren – und die in MySQL 5.6 enthaltenen Neuheiten bezüglich der MySQL-Replikation zu nutzen. MySQL Cluster verwendet die MySQL-Replikation für die Implementierung der geografischen Replikation. Nutzer von MySQL Cluster 7.3 profitieren nun von:

- Prüfsummen für Replikations-Ereignisse
- Verringerter Größe des Binär-Logs
- Der Möglichkeit, Replikationen zeitlich verzögert vorzunehmen

MySQL Cluster Manager: Zentralisierte Sicherung und Wiederherstellung

MySQL Cluster unterstützt Online-Sicherungen (und die anschließende Wiederherstellung dieser Daten). MySQL Cluster Manager 1.2 vereinfacht den Prozess. Die Datenbank kann jetzt mit einem einzelnen Befehl gesichert werden, wobei wiederum die Daten auf jedem Datenknoten im Cluster gesichert werden: „mcm> backup clus-

```
mcm> list backups mycluster;

+----------+--------+--------+----------------------+
| BackupId | NodeId | Host   | Timestamp            |
+----------+--------+--------+----------------------+
| 1        | 1      | grün   | 2012-07-31T06:41:36Z |
| 1        | 2      | braun  | 2012-07-31T06:41:36Z |
| 1        | 3      | grün   | 2012-07-31T06:41:36Z |
| 1        | 4      | braun  | 2012-07-31T06:41:36Z |
| 1        | 5      | lila   | 2012-07-31T06:41:36Z |
| 1        | 6      | rot    | 2012-07-31T06:41:36Z |
| 1        | 7      | lila   | 2012-07-31T06:41:36Z |
| 1        | 8      | rot    | 2012-07-31T06:41:36Z |
+----------+--------+--------+----------------------+
```

Listing 1

ter mycluster;“. Der Befehl „list.“ kann identifizieren, welche Sicherungen im Cluster verfügbar sind (siehe Listing 1).

Man kann dann auswählen, welche dieser Sicherungen man wiederherstellen möchte, indem man beim Aufrufen des Wiederherstellungsbefehls die zugeordnete Sicherungs-ID angibt: „mcm> restore cluster -I 1 mycluster;“. Falls man die vorhandenen Inhalte der Datenbank entfernen muss, bevor man die Wiederherstellung durchführt, kommt die in MCM 1.2 eingeführte Option „—initial“ für den Befehl „start cluster“ zum Einsatz, um die Daten aus allen MySQL-Cluster-Tabellen zu entfernen.

Jürgen Giesel
Juergen.Giesel@oracle.com

Mario Beck
Mario.Beck@oracle.com

Oracle macht den Enterprise Manager 12*c* fit für die Multitenant-Option

Mit dem Oracle Enterprise Manager Cloud Control 12c Release 3 Plug-in Update 1 (12.1.0.3) öffnet Oracle den Weg für das volle „Lifecycle Management“ von Database-as-a-Service (DBaaS) mit der neuen Multitenant-Option der neuen Datenbank 12c.

Hinsichtlich der Datenbank-Bereitstellung bietet das Release umfassende Auswahlmöglichkeiten in Sachen „Provisioning“, „Monitoring“, „Verwaltung“, „Backup“ und „Wiederherstellung“ von Datenbanken in der Cloud.

So sind gesamte Datenbanken, Instanz-Klone, Schemata oder sofort einsatzbereite DBaaS schnell erstellt. Der „DBaaS rapid start Kit“ enthält zudem eine Lösung, die es ermöglicht, Exadata als „out of the Box“-Plattform einzurichten. Sie erlaubt es Administratoren weiterhin, ihr DBaaS-Setup mit Skripten und Third-Party-Lösungen zu integrieren.

Für die Nutzung von „Pluggable Databases“ ist eine Reporting-Funktionalität vorgesehen, die unter anderem eine Messung der Workloads sowie eine Kosten-Analyse ermöglicht. Darüber hinaus können Administratoren auch die Leistungsfähigkeit und den Lebenszyklus von Container Databases administrieren und monitoren.

Ein Service-Katalog, in dem genehmigte Konfigurationen von „Pluggable Databases“ festgehalten werden, soll Unternehmen bei der Standardisierung helfen. Zudem sorgt ein rollen- und richtlinienbasiertes Management für die Einhaltung der Compliance und Governance-Anforderungen.

Das neue Release stellt auch „Testing as a Service“ zur Verfügung. Somit können zusätzlich zu den Lasttests auch Funktions- und Regressionstests über den Enterprise Manager in der Private or Public Cloud durchgeführt werden.

Auch beim Patching bringt das Release Neues mit sich: Mit einem Out-of-Place-Patching lassen sich Datenbanken in einer RAC-Umgebung ohne Ausfallzeiten patchen.

Zusätzlich zu der bisherigen Unterstützung von Oracle ZFS Storage Applicances und Netapp sind nun auch generische ZFS-Dateisysteme nutzbar. So funktioniert die Lösung sowohl auf NAS- als auch auf SAN-Storage gleichmäßig. Darüber hinaus ist das Klonen von Datenbanken jeder Größe unter Verwendung von Snap Clone innerhalb von Minuten erledigt.

Auch für das Daten Lifecyle Management stellt das neue Release einen Advisor zur Verfügung, der „Heat Map“-Informationen über Tablespaces und Segmenten von häufig abgefragten oder geänderten Daten visualisiert und damit eine optimalere Nutzung von Ressourcen ermöglicht.

Partitioning in der Datenbank 12*c*: Was ist neu?

Jan Ott, Trivadis AG

Die neuen Features sollen die tägliche Wartung der Datenbank vereinfachen, die Verfügbarkeit erhöhen und die Performance verbessern – soweit die Ankündigungen. Dabei fallen Worte wie „Asynchronous", „Online", „Partial Indexes" und „Single Operation". Der Artikel zeigt, was es damit auf sich hat, was die neuen Partitioning-Features dazu beitragen und ob die Ziele im praktischen Einsatz erreicht werden.

Eine Herausforderung bei der täglichen Arbeit ist der Umgang mit Partitionen. Die Datenmengen werden größer und daher auf mehr Partitionen verteilt. Partitionen einfach und effizient verwalten zu können, ist von zentraler Bedeutung. Zum Beispiel lassen sich ältere Daten aus mehreren Partitionen zusammenlegen und gleichzeitig komprimieren oder Partitionen, die zu groß wurden, auf mehrere verteilen. Neu in der Datenbank 12*c* ist nun die Möglichkeit, mit einer Operation mehrere Partitionen zu erstellen (ADD), zu löschen (DROP/TRUNCATE), zusammenzuführen (MERGE) und aufzuteilen (SPLIT).

ADD – Multiple Partition

Mehrere Partitionen lassen sich am Ende einer bestehenden Partition oder neuer Listen mit einer Operation einfügen. Dieses Feature ist für die folgenden Partitionierungsarten verfügbar:

- Range/Composite
- Liste; nur, wenn keine Default-Partition existiert

Listing 1 zeigt ein Beispiel zum Einfügen mehrerer Jahre, pro Jahr eine Partition. Die Partitionen sind durch Komma getrennt aufzulisten.

MERGE – Multiple Partition

Dieses Feature zum Zusammenführen mehrerer Partitionen zu einer mit nur einer Operation ist für folgenden Partitionierungsarten verfügbar:

- Range; die Bereiche müssen aneinander grenzen

```
ALTER TABLE t_partition_range
ADD
  PARTITION p_2007
    VALUES LESS THAN (TO_DATE('01.01.2008','dd.mm.yyyy')),
  PARTITION p_2008
    VALUES LESS THAN (TO_DATE('01.01.2009','dd.mm.yyyy')),
  PARTITION p_2009
    VALUES LESS THAN (TO_DATE('01.01.2010','dd.mm.yyyy'))
```

Listing 1

```
ALTER TABLE t_partition_list
MERGE PARTITIONS p_fr, p_it, p_de INTO PARTITION p_weu
```

Listing 2

```
ALTER TABLE t_partition_list
MERGE PARTITIONS p_b TO p_d INTO PARTITION p_b_to_d
```

Listing 3

```
ALTER TABLE t_partition_range
SPLIT PARTITION p_2000 INTO (
  PARTITION p_2000_q1
    VALUES LESS THAN (TO_DATE('01.04.2000','dd.mm.yyyy')),
  PARTITION p_2000_q2
    VALUES LESS THAN (TO_DATE('01.07.2000','dd.mm.yyyy')),
  PARTITION p_2000_q3
    VALUES LESS THAN (TO_DATE('01.09.2000','dd.mm.yyyy')),
  PARTITION p_2000_q4
)
```

Listing 4

- List; die Listen werden zusammengeführt (UNION). Wenn eine der Partitionen die Default-Partition ist, wird sie erweitert
- Partition und Subpartition

In unserem Beispiel sind die einzelnen Partitionen durch Komma getrennt aufgelistet (siehe Listing 2). Ein Bereich ist durch zwei Partitionsnamen abgegrenzt (siehe Listing 3).

```
ALTER TABLE t_partition_range
DROP PARTITIONS p_2003, p_2004, p_2005
```

Listing 5

```
ALTER TABLE t_partition_range
TRUNCATE PARTITIONS p_2003, p_2004, p_2005
```

Listing 6

```
ALTER TABLE t_partition_range
DROP PARTITION p_2003 ,p_2004 UPDATE INDEXES;

Danach ist der globale Index VALID, hat jedoch Waisen (ORPHA-
NED_ENTRIES).
SELECT index_name, status, orphaned_entries
  FROM user_indexes
 WHERE index_name = 'PK_PARTITION_RANGE';

INDEX_NAME          STATUS   ORPHANED_ENTRIES
------------------- -------- -----------------
PK_PARTITION_RANGE  VALID    YES
```

Listing 7

```
BEGIN
  dbms_part.cleanup_gidx
    (schema_name_in => USER
    ,table_name_in => 'T_PARTITION_RANGE'
    );
END;
/

PL/SQL procedure successfully completed.

SELECT index_name, status, orphaned_entries
  FROM user_indexes
 WHERE index_name = 'PK_PARTITION_RANGE';

INDEX_NAME          STATUS   ORPHANED_ENTRIES
------------------- -------- -----------------
PK_PARTITION_RANGE  VALID    NO
```

Listing 8

SPLIT – Multiple Partition

Dieses Feature zum Verteilen der Daten einer Partition auf mehrere Partitionen in einer Operation ist für die folgenden Partitionierungsarten verfügbar:

- Range/List
- Partition/Subpartition

Listing 4 zeigt dazu ein Beispiel.

DROP – Multiple Partition

Dieses Feature zum Entfernen mehrerer Partitionen in einer Operation ist für folgende Partitionierungsarten verfügbar:

- Range/List
- Partition/Subpartition

Ein Beispiel dazu ist in Listing 5 zu sehen.

TRUNCATE – Multiple Partition

Dieses Feature zum Löschen mehrerer Partitionen in einer Operation ist für folgende Partitionierungsarten verfügbar:

- Range/List
- Partition/Subpartition

Listing 6 zeigt dazu ein Beispiel.

Alle Ziele erreicht?

Das Ziel der vereinfachten Wartung wurde erreicht. Es ist mit der neuen Syntax einfacher, aus einer Partition mehrere zu erstellen (SPLIT). Auch die Performance ist besser bei einem SPLIT, da die Ausgangspartition nur einmal gelesen wird. Zudem erfolgt alles in einem Schritt. Die Verfügbarkeit wird leider nur bedingt erreicht. Das System ist zwar schneller wieder verfügbar, bei hoher Last ist jedoch immer noch ein Wartungsfenster erforderlich.

Im Design steht man immer wieder vor der Frage „Können wir globale Indizes auf partitionierten Tabellen einsetzen?“ Ein „TRUNCATE“ oder „DROP“ einer Partition setzt den globalen Index auf „unusable“. Danach ist der Zugriff über diesen Index nicht mehr möglich; es wird ein „full table scan“ verwendet. Dabei kommt das System zum Stillstand. In der Version 11*g* gab es dafür die beiden Möglichkeiten, entweder keine globalen Indizes zu verwenden oder „TRUNCATE/DROP“ durch „DELETE“ zu ersetzen (was die Last enorm erhöht) und bei „DROP“ die Partition im Wartungsfenster zu entfernen.

Oracle geht diese Herausforderung jetzt an. Es bleibt ein globaler Index verfügbar („usable“). Bei diesem wird markiert, dass er Waisen („orphan“) besitzt. Diese verwaisten Einträge werden zu einem späteren Zeitpunkt aufgeräumt. Die Transaktion ist ein DDL-Statement und dauert daher, wie von „TRUNCATE/DROP“ gewohnt, nur sehr kurz. Es gelten folgende Verfügbarkeit und Einschränkungen:

- Range/List
- Partitionen/Subpartitionen
- Nur „Heap“-Tabellen, keine „Index only“-Tabellen (IOT)
- Keine Objekt-Types und Domain-Indizes

```
ALTER TABLE t_partition_range
MOVE PARTITION p_2003 ONLINE UPDATE INDEXES

dbms_part.cleanup_online_op(
  schema_name    => USER,
  table_name     => 'T_PARTITION_RANGE',
  partition_name => 'P_2003'
);
```

Listing 9

```
CREATE TABLE t_partition_range(
   id      INTEGER NOT NULL CONSTRAINT pk_partition_range
                                    PRIMARY KEY USING IN-
DEX
  ,name    VARCHAR2(100)
  ,time_id DATE
)
-- setzen für die Tabelle => default für die Partitionen
INDEXING OFF
PARTITION BY RANGE (time_id) (
  -- nicht spezifiziert => übernimmt das Setting der Tabelle
  PARTITION p_2000
    VALUES LESS THAN (TO_DATE('01.01.2001','dd.mm.yyyy'))
  -- kann gesetzt werden, nicht nötig, das gleich wie Tabelle
 ,PARTITION p_2001
    VALUES LESS THAN (TO_DATE('01.01.2002','dd.mm.yyyy'))
    INDEXING OFF
  -- für die Partition ist das Indexieren eingeschaltet
 ,PARTITION p_2002
    VALUES LESS THAN (TO_DATE('01.01.2003','dd.mm.yyyy'))
  INDEXING ON)
```

Listing 10

```
ALTER TABLE t_partition_range
ADD PARTITION p_2003
  VALUES LESS THAN (TO_DATE('01.01.2004','dd.mm.yyyy'))
  INDEXING ON

SELECT table_name, partition_name, indexing
  FROM user_tab_partitions
 WHERE table_name = 'T_PARTITION_RANGE'
 ORDER BY partition_name;

TABLE_NAME          PARTITION_NAME  INDEXING
------------------- --------------- ----------
T_PARTITION_RANGE           P_2000        OFF
T_PARTITION_RANGE           P_2001        OFF
T_PARTITION_RANGE           P_2002        ON
T_PARTITION_RANGE           P_2003        ON
```

Listing 11

Bei „DROP“ oder „TRUNCATE“ muss neu „UPDATE INDEXES“ verwendet werden. Der Index wird wie bisher erstellt. Die Maintenance des Index übernimmt Oracle automatisch durch den Job „SYS.PMO_DEFERED_GIDX_MAINT_JOB“. Dieser läuft per Default um zwei Uhr in der Nacht. Der Zeitpunkt lässt sich anpassen. Zudem kann die Wartung mit dem „DBMS_PART“-Paket jederzeit injiziert werden. Listing 7 zeigt dazu ein Beispiel (bitte „UPDATE INDEXES“ beachten). In Listing 8 wird der Index aufgeräumt und nicht dem Wartungsjob überlassen.

Das Ziel „Asynchronous Global Index Maintenance” ist auf der ganzen Linie erreicht. Globale Indizes können nun vermehrt verwendet werden. Die Performance liegt im Subsekunden-Bereich, da das DDL-Statement eine Data-Dictionary-Änderung ist. Der Index wird zu einem späteren Zeitpunkt aufgeräumt. Dies kann in einem Wartungsfenster erledigt oder dem automatischen Job „PMO_DEFERED_GIDX_MAINT_JOB“ überlassen werden. Somit ist auch die Verfügbarkeit erhöht, da der Index immer online ist.

Online Move Partition

Die Oracle-Datenbank muss verfügbar sein, die Wartungsfenster werden immer kürzer. Die Möglichkeit, Partitionen während des Betriebs zu reorganisieren, um Platz freizugeben, sie in einen neuen Tablespace zu verschieben, zu komprimieren und/oder Parameter zu verändern, ist daher sehr willkommen. Wenn es zu Problemen kommt, stellt Oracle das Paket „DBMS_PART“ zur Verfügung. Listing 9 zeigt ein Beispiel. Zu beachten ist das Schlüsselwort „ONLINE“. Mit „UPDATE INDEXES“ wird erreicht, dass die Indizes verfügbar bleiben.

Auch hier ist das Ziel erreicht. Die Daten bleiben während der Reorganisation verfügbar. Ein Wartungsfenster ist nicht erforderlich – es besteht nur eine kleine Einschränkung. Auf der Partition wird ein exklusives „Lock“ benötigt. Dies kann bei Systemen mit hoher Last ein Ding der Unmöglichkeit sein, zum Beispiel, um Partitionen mit Vergangenheitsdaten zu kompri-

```
CREATE INDEX idx_parr_name_partial ON t_partition_range
(name)
GLOBAL INDEXING PARTIAL
```

Listing 12

```
SELECT index_name, status, num_rows
  FROM user_indexes
 WHERE index_name IN (‚PK_PARTITION_RANGE‘
                     ,‘IDX_NAME_GLOBAL_PARTIAL‘);
```

Listing 13

```
INDEX_NAME                STATUS   NUM_ROWS
------------------------- -------- ----------
PK_PARTITION_RANGE        VALID    48
IDX_NAME_GLOBAL_PARTIAL   VALID    24
```

Listing 14

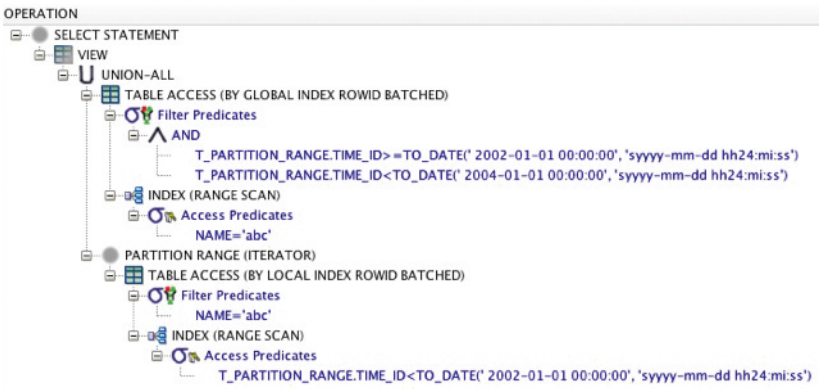


Abbildung 1: Ein Explain Plan im SQL Developer, Partial Index 1

```
CREATE TABLE t_partition_parent (
  pk INTEGER NOT NULL CONSTRAINT pk PRIMARY KEY USING INDEX,
  i  INTEGER
)
PARTITION BY RANGE (pk) INTERVAL (10)
(PARTITION p_1 VALUES LESS THAN (10))
```

Listing 15

mieren oder in einen anderen Tablespace auslagern, da auf die Daten weniger zugegriffen wird. Es gilt, ein Zeitfenster mit kleiner Last zu finden, zum Beispiel am Wochenende oder in der Nacht, um die Daten dann online zu reorganisieren. Da das System weiterläuft, können vereinzelte Benutzer arbeiten und werden nicht von einem Wartungsfenster behindert.

Partial Indexes for Partitioned Tables

Es wird immer wieder die Frage gestellt, wie viele Indizes man verwenden sollte. Ein Grund dafür ist der Platzbedarf. Bei den neuen partiellen Indizes kann der Index pro Partition ein- oder ausgeschaltet werden. Die Partition des ausgeschalten Index ist leer, braucht somit keinen Platz und muss nicht nachgeführt werden. Es gelten folgende Verfügbarkeit und Einschränkungen:

- Globale Indizes
- Lokale Indizes

Listing 10 zeigt dazu ein Beispiel. Im ersten Schritt sind die Tabelle und die Partitionen entsprechend zu definieren. Bei neuen Partitionen kann man das Indexierungsverhalten angeben, dabei wird der Index eingeschaltet (siehe Listing 11). Nun ist der Index partiell zu definieren, in Listing 12 als globaler Index.

Die Anweisung in Listing 13 zeigt, was im partiellen Index steht. Die Tabelle enthält 48 Zeilen, 12 pro Partition. Der normale Index „PK_PARTITION_RANGE“ hat 48 Zeilen. Doch im partiellen Index sind nur 24 (siehe Listing 14).

Interessant ist, wie Oracle die Daten selektiert (siehe Abbildung 1): „SELECT * FROM t_partition_range WHERE name = ‚abc‘;“. Der Optimizer löst das vorherige Select mit einem „UNION-ALL“ auf. Er verwendet den globalen partiellen Index für die Partitionen mit dem eingeschalteten Index. Für den Rest der Daten macht er einen Scan auf die Partitionen, die nicht im Index sind.

Partielle Indizes sind hilfreich für Tabellen, in denen zusätzliche Indizes für bestimmte Bereiche einer Tabelle gebraucht werden. Es ist nun möglich, diese individueller dem Abfrage-Muster anzupassen. Leider gibt es nur ein „ON/OFF“-Muster für alle partiellen Indizes einer Tabelle.

Interval-Reference-Partitioning

Interval-Partitioning ist seit der Version 11*g* eine Erleichterung. Es kann ein Intervall angegeben werden und Oracle fügt die benötigten Partitionen automatisch an. Das Intervall lässt sich nun auch auf eine Kind-Tabelle übertragen. Im ersten Schritt erstellt man eine „INTERVAL“-Vater-Tabelle (siehe Listing 15). Der zweite Schritt ist das

```
CREATE TABLE t_partition_child(
  fk INTEGER NOT NULL,
  i  INTEGER,
  CONSTRAINT part_child_fk FOREIGN KEY (fk)
                           REFERENCES t_partition_parent(pk)
)
PARTITION BY REFERENCE(part_child_fk)
```

Listing 16

Erstellen einer Kind-Tabelle (siehe Listing 16). Die Referenz auf die Vater-Tabelle war in der Version 11*g* nicht möglich.

Interval-Reference-Partitioning ist sicher eine nützliche Ergänzung und kann die Wartung vereinfachen, da man beispielsweise Vater-Kind in einer Operation austauschen kann („exchange"). Neue Partitionen werden automatisch angelegt und synchron zwischen Vater- und Kind-Partitionen gehalten. Oracle führt darüber hinaus mit der Version 12*c* noch drei weitere Features ein:

- „TRUNCATE TABLE" nun mit der „CASCADE"-Option
- XML DB und Domain Index unterstützen Hash-Partitionierung
- Statistiken unterstützen inkrementelle Statistiken für die „Exchange"-Partition

Fazit

Die Partitionierung wurde von Oracle in die richtige Richtung entwickelt. Die Wartung ist vereinfacht, da globale Indizes auf partitionierte Tabellen erstellt und die effizienten Befehle „DROP" und „TRUNCATE" verwendet werden können. Ein weiterer Vorteil ist das Interval-Reference-Partitioning, wobei die Datenbank die Erstellung der benötigten Partitionen der Vater- und Kind-Tabelle automatisch übernimmt.

Die Performance ist durch die „SPLIT/MERGE"-Partition erhöht, so muss zum Beispiel beim Split die Partition nicht mehrmals gelesen werden. Auch die Verfügbarkeit ist gestiegen, „Online Move"-Partitionen lassen sich nun ohne Wartungsfenster verschieben.

Eine Sache stört allerdings immer noch: Es muss der richtige Zeitpunkt abgewartet werden. Bei hoher Last erhält die Datenbank das „exklusive Lock" nicht und der Prozess bricht ab. Bei hoher Last auf der Partition braucht es also weiterhin ein Wartungsfenster.

Immerhin sind einige Ziele erreicht. Wer mit Partitionen arbeitet, wird die kleinen, aber feinen Verbesserungen zu schätzen wissen.

Jan Ott
jan.ott@trivadis.com

Oracle Data Integrator 12c mit neuer Benutzeroberfläche und Golden Gate mit Mandantenfähigkeit-Unterstützung

Mit dem Major Update seines Data-Integrations-Portfolios bringt Oracle die Version 12c von Data Integrator (ODI) und Golden Gate heraus. Die Entwicklung vom Warehouse Builder indes wurde eingestellt.

Oracle hat dem Data Integrator eine neue Benutzeroberfläche verpasst. Mit ihrem deklarativen, Flow-basierten Ansatz sorgt sie für eine einfache Bedienung. Zudem kann die Mapping-Logik wiederverwendet werden, was für kürzere Entwicklungszeiten sorgt. In Sachen Performance hat Oracle an der Parallelität der Daten-Integrationsprozesse gearbeitet.

Oracle hat auch die Interoperabilität zwischen dem Data Integrator und dem Oracle Warehouse Builder (OWB) – dessen Entwicklung eingestellt wurde – verbessert. Dies soll OWB-Kunden eine einfachere Migration nach ODI ermöglichen. Die enge Integration mit dem Enterprise Manager 12c sorgt für ein besseres, zentrales Monitoring. Darüber hinaus ist auch ein engeres Zusammenspiel zwischen ODI und Golden Gate vorgesehen. Dies ermöglicht ein effizienteres Laden und Transformieren von Real-Time-Daten.

Bei Oracle Golden Gate hat Oracle vor allem an der Optimierung für Oracle Database 12c gearbeitet. Dabei unterstützt das Produkt die mandantenfähige Architektur der neuen Datenbank und die Cloud-basierte Echtzeit-Replikation.

Für eine bessere Performance und Skalierbarkeit liefert Oracle extra für Oracle Golden Gate eine einfache Streaming-API.

Im Bereich Security hat das Development-Team die Integration mit Oracle Credential Store und Oracle Wallet weiterentwickelt, die den Anwndern das Speichern und Zurückholen von verschlüsselten Benutzernamen und Passwörtern ermöglicht.

In puncto Hochverfügbarkeit hilft die Integration mit dem Data Guard und Fast-Start-Fail-Over (FSFO) für mehr Ausfallsicherheit.

Darüber hinaus bringt Golden Gate 12c eine bessere Unterstützung von Echtzeit-Replikation von Daten in heterogenen Umgebungen mit MySQL, Microsoft SQL Server, Sybase nutzen IBM DB2 mit sich.

ADF, Forms und .NET – alles vereint in einer Mobile-Scanner-App bei Volkswagen

Madi Serban, Bahar Us, Rastislav Misecka und Mathias Waedt, PITSS sowie David Christmann, Volkswagen AG

Das anspruchsvolle Projekt wurde im Jahr 2013 durchgeführt. Ziel war die Neuentwicklung einer alten Forms-2.0-Anwendung für Handscanner-Mobil-Geräte. Die neue Anwendung sollte auf unterstützten Plattformen laufen, toll aussehen, eine hervorragende Leistung anbieten und rechtzeitig entwickelt werden (zwei Monate Entwicklungszeit).

Dieser Artikel beschreibt die Überlegungen, die zur im Titel genannten Technologie-Auswahl geführt haben, Herausforderungen (wie Session Management, Tastatur-Bedienung, Sicherheit etc.), Architektur, das Zusammenspiel der Komponenten und die Lösungen, die es uns erlaubten, das Rennen gegen die Zeit zu gewinnen.

Alte Forms-Anwendungen für mobile Geräte

Oracle Forms ist in vielen Großunternehmen immer noch präsent. Man betreibt damit seit Jahrzehnten geschäftskritische Prozesse. Forms-Anwendungen funktionieren zuverlässig, mit exzellenter Leistungsfähigkeit und benötigen nur ab und zu Wartung oder Weiter-Entwicklung. Die meisten Forms-Anwendungen haben auch mindestens ein paar Masken, die für mobile Endgeräte konzipiert wurden. Diese sind in vielen Fällen von enormer Bedeutung, vielleicht sogar die meistbenutzten Masken, zum Beispiel für Handheld-Scanner in der Handels- oder Automobil-Industrie. Deshalb ist es sehr wichtig, dass diese mobilen Masken ständig funktionsfähig sind.

Was tun wir dann, wenn solche Masken Probleme bereiten? Zum Beispiel, wenn Oracle die alten Forms-Versionen nicht mehr unterstützt und diese modernisiert werden müssen. Die einfachste und stressfreieste Lösung wäre eine Migration nach Forms 11g. Diese Lösung ist aber aufgrund spezifischer Hardware- und Software-Konfigurationen oder Benutzer-Anforderungen nicht für alle Situationen passend.

Obwohl die nach Forms 11g migrierten Masken Web-fähig sind und ganz gut aussehen, sind sie trotzdem auf manchen mobilen Geräten nicht mehr lauffähig, weil die Ressourcen für ein Forms Applet schlicht und einfach fehlen. Die Remote-Desktop-Lösung ist manchmal hilfreich, aber nur mit großen Security- und Konfigurations-Problemen erreichbar. Die Remote-Lösung (Citrix) war in diesem Fall auch nicht kompatibel mit den verfügbaren Geräten.

Die Optionen für Upgrade, Weiterentwicklung, Migration und Neuentwicklung müssen dann gut gewichtet sein, um eine optimale Entscheidung zu finden und die Anwendungslebensdauer um weitere Jahrzehnte zu verlängern. Dafür nahmen die Autoren zuerst die Anforderungen unter die Lupe.

Die neuen Ziele für mobile Datenbank-Anwendungen

Natürlich soll die neue Anwendung auch weiterhin die alten Funktionalitäten genauso gut abdecken wie die alte:

- Endkunden erwarten eine mindestens gleichbleibend tolle Leistung, und das ist nicht bei allen Technologien so selbstverständlich wie bei Oracle Forms.
- Hot-Key-Funktionen sind in Oracle Forms sehr beliebt. Leider sind sie in Web- und Touchpad-Anwendungen nicht üblich und benötigen einen sehr hohen Programmierungsaufwand. In Browser-Anwendungen bedeutet das meistens viel JavaScript und in Desktop-Anwendungen das Überschreiben der normalen Komponenten-Funktionalität. Dies ist nicht zu empfehlen, aber leider auch in vielen Situationen nicht zu vermeiden.

Hinzu kommen meistens folgende Neu-Anforderungen:

- Unterstützung für neue Geräte und neue Betriebssysteme wie Android, iOS, etc., aber auch für alte Geräte. In dieser Situation war die Unterstützung für Microsoft Windows CE 5 nicht zu umgehen.
- Strategische Richtung für andere Programmiersprachen wie Java
- Zertifizierte Laufzeitumgebung
- Möglichst die Verfolgung Oracle-strategischer Empfehlungen und Best-Practices
- Sicherung der Investition, um die zukünftige Entwicklung und den Migrationsbedarf zu verringern
- Gutaussehende Benutzeroberfläche – schließlich sind Endkunden heutzutage modernste Apps für Mobil-Telefone gewöhnt und wünschen sich natürlich, eine entsprechende Benutzeroberfläche auch im Arbeitsumfeld zu haben
- Kiosk/App Modus – also Anwendung nicht einfach im Browser öffnen, sondern als App
- Funktionelle Erweiterungen

Und das alles soll mit minimalem Zeitaufwand und maximaler Qualität erfüllt werden. Für dieses Projekt-Beispiel ging es um zwei Monate Entwicklungszeit. Es war ein Rennen gegen die Zeit und das hat motiviert, ständig nach klugen Lösungen zu suchen. Es war sicherlich eine spannende Research- und Entwicklungs-Zeit. Warum Research? Zwei Wochen waren allein notwendig, um die optimale Technologie-Auswahl und Architektur zu definieren.

Die Technologie

Wenn man nach Alternativen für Forms sucht, dann richten sich die ersten Gedanken meistens auf ADF und Apex. Danach zieht man auch Java-Open-Source und Microsoft .NET in Erwägung. ADF und Apex bieten exzellente Lösungen für mobile Oracle-Datenbank-Anwendungen wegen der hohen Kompatibilität mit den restlichen Oracle-Anwendungen und auch der Möglichkeiten für eine vereinfachte Migration. Die ADF-Optionen für eine mobile Anwendungsentwicklung waren:

- Oracle ADF Faces Rich Client Components
- Oracle ADF Mobile Browser

Leider konnte keine dieser Optionen alleine alle Anforderungen erfüllen. Die Zertifizierungsmatrix für Oracle ADF Mobile Browser sieht beim ersten Blick in Ordnung aus (siehe Tabelle 1). Bei näherer Betrachtung bestätigte sich jedoch die alte Regel: Der Teufel steckt im Detail. Eignungstests haben schnell gezeigt, dass eine ADF-Anwendung in der mobilen Version des IE für Windows CE 5 zwar prinzipiell lauffähig war, jedoch nicht ganz in gewünschtem Umfang. Das Problem ist der für Embedded Systems angepasste Internet Explorer Browser Version 5 oder 6, der dafür nicht ausreichend JavaScript- und PPR-fähig (partial page rendering, alias Ajax) ist. JavaScript und PPR waren für die Hot-Key-Funktionen jedoch ein absolutes Muss.

Die Lösung

Wie eingangs im Titel erwähnt, bestand die Lösung aus einer Kombination aus Oracle-Datenbank, ADF-Anwendung und .NET-Benutzeroberfläche (siehe Abbildung 1). Die Forms-Funktionalität wurde dafür in drei Schichten migriert: eine Thin-.NET-Benutzeroberfläche sowie ein Thick-Datenbank- und ein ADF-Layer, alles schnell in fünf Phasen durchgeführt (siehe Abbildung 2):

- Phase 1: Forms-Vorbereitung, Migration PL/SQL Business Logic in die Datenbank
- Phase 2: Upgrade Oracle Forms 3 nach 11g
- Phase 3: Migration Oracle Forms nach ADF
- Phase 4: Entwicklung der .NET-Benutzeroberfläche und Integration mit ADF
- Phase 5: Test und Go-Live

BROWSER	ADF Mobile
BlackBerry Browser 4	Certified
WebKit-based mobile browsers (iPhone Safari, Android Chrome, Nokia S60)	Certified
Access NetFront 3	Certified
OpenWave (UP Browser) 7	Certified
Opera Mini 8	Certified
Pocket Internet Explorer for Windows Mobile 5, 6	Certified
Mobile Internet Browser 2.0 (Motorola)	Certified
Other Basic XHTML mobile browsers	Supported

Tabelle 1: Mobile Browsers zertifiziert für Oracle ADF Mobile Browser siehe http://www.oracle.com/technetwork/developer-tools/jdev/jdev11gr2-cert-405181.html

Die größten Herausforderungen

Wie migriert man 90 Prozent der Forms-PL/SQL-Business-Logik in die Datenbank, sodass diese optimal von ADF konsumiert werden kann? Was muss man hier berücksichtigen? Nachdem die Lösung für alle Phasen festgelegt war, konnten die Bausteine nacheinander entwickelt und integriert werden. Es war klar, dass der Prozessfluss als Services gebaut werden sollte, um den unabhängigen Informationsaustausch zwischen der Benutzeroberfläche und dem Backend zu erlauben und so die Datenkonsistenz zu erhalten.

Um diese Services herzustellen, wurde die Forms-Business-Logik (Trigger und Program-Units, die den Prozessfluss abbildeten) in die Datenbank migriert. Dafür hat man die Benutzer-Interaktion, insbesondere die notwendigen Hot-Key-Funktionen, als Ansatzpunkt gewählt und mithilfe der PITSS.CON-Top-Down- und Bottom-Up-Analyse die gesamte Kette von Abhängigkeiten und relevanter Geschäftslogik identifiziert und in Datenbank-Pakete migriert. PITSS.CON-BL-Assistant hat sich um die Migration gekümmert und durch die automatische Erkennung der Forms-Bind-Variablen und die Umwandlung in Parameter viel Projektzeit gespart. Diese Parametrisierung hatte auch eine schöne Nebenwirkung: Viele Prozeduren oder Funktionen, die in den Forms-Modulen vervielfacht waren, konnten durch die Parametrisierung eliminiert werden, was langfristig zu vereinfachter Wartung führt.

Als die Technologien in der Datenbank, in den Web-Services und auf der Benutzeroberfläche unabhängig voneinander funktionierten, waren die absolut getrennten, atomaren Funktionen notwendig. Zum Beispiel wurden in den PL/SQL-Paketen keine globalen Variablen benutzt, um eine geteilte Datenbank-Session zu erlauben. Weil manche PL/SQL-Datentypen kein Interface im JDBC-API haben, wurden stattdessen einfache Datentypen in den PL/SQL-Interfaces durch Web-Services ersetzt. Was man hier noch ganz am Anfang berücksichtigen musste, war die Portierung zentraler Funktionalitäten wie Fehlerbehandlung und Meldungen von Forms zur Datenbank.

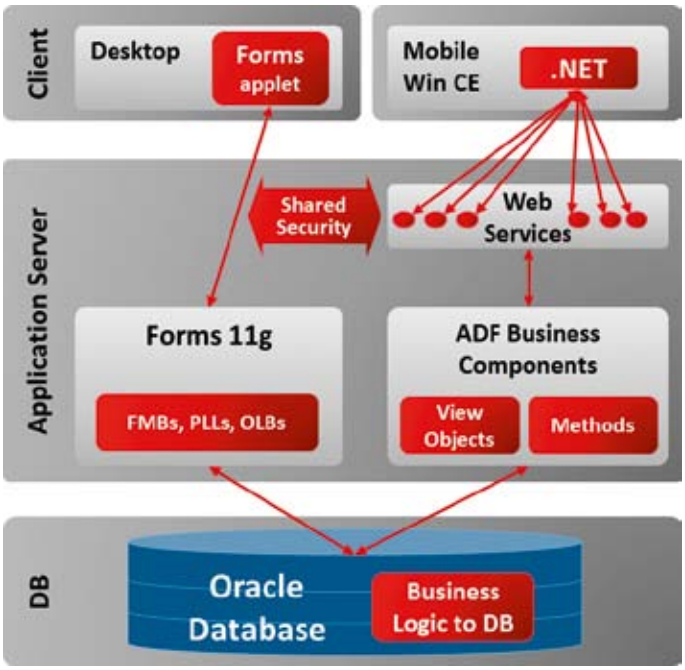


Abbildung 1: Forms, ADF-Server und .NET-Architektur

Solche Funktionalitäten waren in allen Forms-Triggern, Prozeduren und Funktionen präsent und ohne eine Datenbank-zentralisierte Fehlerbehandlung wäre es nicht möglich, die Logik richtig zu migrieren.

Zusätzlich wurde die Logik in der Datenbank in bestimmten Situationen weiter aufgeteilt oder parametrisiert, um den Prozess gegebenenfalls zu unterbrechen und um den Informationsaustausch mit der Benutzeroberfläche zu ermöglichen. Die Datenbank-Schnittstellen hat man mit Parametern ausgebaut, die entweder in der Datenbank-Logik benötigt oder in der Benutzeroberfläche abgebildet wurden.

Die ganze Migration der Logik war eine Herausforderung, auch aufgrund des Debuggens und Testens in unterschiedlichen Programmiersprachen und Entwicklungsumgebungen. Aber es hat sich gelohnt: Die Logik liegt nun sicher in der Datenbank und kann von dort sowohl von Forms als auch von ADF, .NET oder in Zukunft auch von anderen Technologien konsumiert werden.

In der Phase 2 erfolgte das Upgrade von Oracle Forms 3 nach 11g. Die Forms-Module sind durch die vielen Entwicklungsjahre sehr komplex geworden. Wegen dieser Komplexität und auch wegen der teilweise veralteten Dokumentation ist die Weiterentwicklung sehr schwierig geworden. Deshalb wurde hier PITSS.CON eingesetzt, um den Großteil der Anwendung nach Forms 11g zu migrieren. Dies war die einzige passende Lösung und erlaubte innerhalb kürzester Zeit die Umstellung auf eine moderne Technologie.

Eine leistungsfähige ADF-Web-Services-Architektur

In der Phase 3 ging es um die Fragen „Wie baut man eine leistungsfähige ADF-Web-Services-Architektur?“ und „Welche Herausforderungen sind zu bewältigen?“ Die Middle-Tier dieses Systems wurde auch so schlank wie möglich gehalten. Dadurch ist die Wartbarkeit der Kern-Funktionalität in der Datenbank zentralisiert und von den übrigen Teilsystemen weitestgehend entkoppelt. Da das System auch in Umgebungen mit eingeschränkter Netzwerkverbindung eingesetzt werden muss, hat man die Kommunikation zwischen Handheld und Server auf ein Mindestmaß gesenkt.

Eine der Eigenschaften von Forms ist die Möglichkeit, UI und Business Logic (BL) ganz einfach zusammenzulegen. Was in Forms effektiv sein kann, erweist sich in Migrationen auf Model-View-Controller-Frameworks (MVC) üblicherweise als trickreicher Spießrutenlauf. Genau dies war eine der weiteren Herausforderungen an das neue verteilte System. Die Hauptaufgabe war hier die Entflechtung von UI und BL, um möglichst viel BL in die Datenbank zu verlagern (Wartbarkeit). Typisch sind hier Prozessstränge, die, durchsetzt mit Rückfragen an den Benutzer, unterschiedliche Verläufe annehmen können. Es gilt auch hier, den besten Mittelweg unter Reduzierung des Communication-Payload und der Anzahl der Request-Response-Zyklen (Server-Roundtrips) zu finden sowie dabei die vorhandene Funktionalität zu gewährleisten und sogar zu verbessern. Im Web-Services-Bereich wurden größtenteils Standard-Komponenten von Oracle ADF eingesetzt.

Ein weiterer Punkt war der großzügige Einsatz von Datenbank-Prozedur-Aufrufen. Wo Forms üblicherweise mit dedizierter Datenbank-Verbindung arbeitet, wird im ADF-Bereich Connection Pooling eingesetzt, das Connection Management also extern (etwa vom WebLogic Server) verwaltet. Der Vorteil hierbei ist, dass der Server selbst

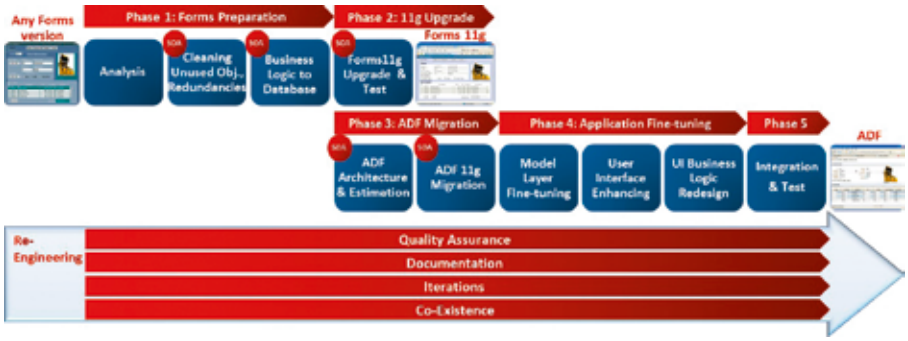


Abbildung 2: Forms nach 11g und ADF Modernisierungsprozess mit PITSS.CON

die Kontrolle über diese Ressourcen hat und seine Leistungsfähigkeit diesbezüglich dynamisch steuern kann. Die Herausforderung hier war, trotz wechselnder Datenbank-Connections und Sessions einen konsistenten Datenstand zu bewahren.

Zusätzlich dazu war es Bedingung, dass auf dem WebLogic Server, abhängig von Handheld und Benutzer (also eingehenden Web-Service-Requests), unterschiedliche Datenbanken angesprochen werden konnten. Das heißt, während der Laufzeit musste dynamisch die richtige Connection-Art aus dem Pool herausgefischt werden. Eine weitere wichtige Anforderung war, das System ohne große Änderungen an mehreren Standorten einsetzen zu können. Dabei sollten identische Datenstrukturen, aber mit unterschiedlichem Datenstand, angesprochen werden. Mehrsprachigkeit war ebenfalls eine Bedingung. Meldungen an den Benutzer müssen in der für ihn eingestellten Sprache erfolgen. Selbstverständlich muss dies alles unter den Gesichtspunkten einer gesicherten Kommunikation erfolgen.

Abbildung 3: Die neue Benutzeroberfläche

Alles außer Routine

In Phase 4 erfolgten in sechs Wochen die Entwicklung der .NET-Benutzeroberfläche sowie die Integration von .NET und ADF. Die Anwendung für die Handhelds soll an mehreren Standorten zum Einsatz kommen. Aus einer solchen Konstellation heraus ergibt sich fast unweigerlich, dass auch unterschiedliche Geräte, zum Teil auch von mehreren Herstellern, mit unterschiedlichen Eigenschaften und Fähigkeiten in Gebrauch sind. Für die Entwicklung einer solchen Anwendung hat dies zur Folge, dass man sich praktischerweise für eine Implementierungs-Plattform entscheiden muss, die von allen eingesetzten Geräten unterstützt wird. In dem vorliegenden Fall hat sich für die gestellten Anforderungen das „.NET Compact Framework 2.0 SP2“ als die geeignetste Plattform herauskristallisiert.

Der Einsatz von .NET bietet viele Vorteile, nicht zuletzt auch den Umstand, dass die für das Compact Framework kompilierten Anwendungen in den meisten Fällen auch unter der Desktop-Version des Frameworks lauffähig sind. Das hat erhebliche Vorteile beim Support und der Wartung: Man kann die Anwendung sehr gut auf einem ganz normalen PC betreuen, ohne die Notwendigkeit, immer ein Handheld bereitzuhalten.

Doch es gibt auch eine Schattenseite: Das Compact Framework hat gegenüber der großen Desktop-Version einen etwas reduzierten Funktionsumfang. Damit nicht genug – man muss auch feststellen, dass Microsoft bei der Wahl der wegzulassenden Features nicht immer ein glückliches Händchen hatte. So fehlen einige Funktionalitäten, die schmerzlich vermisst werden und die man sich dann selbst nachprogrammieren muss.

Oberflächen-Gestaltung

Die Trends im „Consumer Electronics“-Bereich (Smartphones) sind klar: größer, schärfer, brillanter. Immer höhere Auflösung – HD Displays gehören ja schon fast zum Standard, es gibt sogar schon einige Exemplare mit einer Full-HD-Auflösung. Bei den mobilen Geräten für die Industrie sieht es noch anders aus. Hier zählen völlig anderen Werte: Stoßfestigkeit, Kältebeständigkeit, Kratz- und Staub-Unempfindlichkeit, Bedienbarkeit mit Handschuhen oder Integration mit Bar-Code-Scannern.

Aber nicht selten sind auch Geräte mit Display-Auflösungen von nur 240x300 Pixeln in Gebrauch. Insbesondere der letztgenannte Punkt kann einen Full-HD-Desktop- verwöhnten Entwickler vor unerwartete Herausforderungen stellen. Schon allein die bündige Positionierung der Controls im visuellen Designer für diese Auflösung wird zur gänzlich neuen Erfahrung und erfordert eine sichere Hand – und die Geduld eines Uhrmachers.

Überhaupt ist die Aufgabe, das richtige Layout für die einzelnen Masken zu finden, keine einfache. Beim Entwurf muss man gleich mehrere, zum Teil widersprüchliche Aspekte berücksichtigen. In erster Linie sind das natürlich die Hardware-Möglichkeiten der eingesetzten Geräte. Nicht alle Geräte verfügen über einen Touchscreen. Und selbst wenn sie es täten, entspräche eine rein Touch-zentrische Bedienung nicht den Gegebenheiten des Einsatzorts. Ein Touchscreen mit einer Auflösung von nur 240x300 Pixeln im Winter auf einem kalten Parkplatz mit dicken Handschuhen treffsicher bedienen zu müssen – keine allzu verlockende Vorstellung.

Andererseits sind Smartphones und andere mobile Geräte (Tablets) heutzutage so verbreitet, dass sie zu einer Art Etalon des GUI-Designs geworden sind. Daher muss man sich wirklich anstrengen, wenn die eigene Oberfläche im Vergleich dazu nicht als altbacken aussehend wahrgenommen werden soll. Selbst wenn es sich nur um eine Industrie-Anwendung handelt.

Dieser Einfluss fand seinen stärksten Ausdruck in der Gestaltung des Anwendungs-Menüs. Große Icons, die man gegebenenfalls auch noch mit Handschuhen gut treffen kann und das Scrollen mittels Wischbewe-

gungen (falls vom Gerät unterstützt) machen das Auffinden der gewünschten Funktionalität zum Kinderspiel (die Bedienung mit den Pfeiltasten der Tastatur ist aber natürlich auch möglich). Bei der Gestaltung der restlichen Masken überwogen die funktionalen Anforderungen und das Bestreben, bei der – relativ gesehen – geringen Auflösung so viele relevante Informationen wie möglich darzustellen. Einer der Tricks, um das letztgenannte Vorhaben zu verwirklichen, war beispielsweise die farbliche Hervorhebung (eine Art „dritte Dimension“) der Datensätze mit speziellen Merkmalen (etwa Stornierungen) anstatt dedizierter Felder (siehe Abbildung 3).

Wichtig bei der Entscheidung war auch die Tatsache, dass das Haupt-Eingabegerät in den meisten Fällen der Barcode-Scanner ist. Wo sich die direkte Tastatur-Eingabe nicht vermeiden ließ, wird dem Anwender mit allerlei Komfort-Features unter die Arme gegriffen – etwa mit einem automatischen Sprung zum nächsten Eingabefeld, nachdem die Eingabe im aktuellen Feld als vollständig erkannt wurde. Die restliche Bedienung mittels Hotkeys orientierte sich an den Anforderungen und dem Umstand, dass es sich hierbei um die Migration einer bestehenden Anwendung handelt. Die Beibehaltung der gewohnten Abläufe trägt entscheidend zur schnellen Akzeptanz der Neuerung bei den Endanwendern bei.

Ein weiterer Bereich, der sich stark an modernen mobilen GUIs orientiert, ist die Navigation zwischen den einzelnen Seiten. Es wurde (mit Ausnahme von Fehlermeldungen und wichtigen Benachrichtigungen) weitestgehend auf Pop-up-Fenster verzichtet und stattdessen auf das sogenannte „Browser-Konzept“ gesetzt – für eine zusätzliche Information oder durchzuführende Aktion (Beispiel: LOV-Auswahl) wird eine separate Seite angezeigt, die die gesamte Bildschirmfläche einnimmt. Die alte Seite und deren Inhalt werden jedoch nicht verworfen, sondern beibehalten und nach Abschluss der Aktion wieder angezeigt. Nicht unähnlich einer Internet-Browser-Navigation, daher auch der Name. Dadurch wird verhindert, dass die wertvolle Anzeigefläche durch „Fenster-Chrome“ überproportional vereinnahmt wird.

Besondere Aufmerksamkeit erforderte auch das Thema „Mehrsprachigkeit“. Hier hat sich zum einen der reduzierte Umfang des Compact Frameworks bei der Steuerung der Ländereinstellungen des Benutzers bemerkbar gemacht: In der Desktop-Version des Frameworks ist es vergleichsweise einfach, die Sprache des UI durch Setzen der Eigenschaft „CurrentUICulture“ zu Laufzeit zu ändern. Leider wurde diese Eigenschaft in der Compact-Version des Frameworks eingespart und das UI läuft immer mit den Spracheinstellungen des Betriebssystems. Dies ist allerdings nicht immer ausreichend. Manchmal ist es einfach wünschenswert, die Oberflächensprache den (oft in der Datenbank gespeicherten) Einstellungen des aktuell angemeldeten Benutzers dynamisch anzupassen. So war das auch im vorliegenden Fall.

Eine weitere Komplikation bestand in der Tatsache, dass die ursprüngliche Forms-Anwendung ihre lokalisierten Texte in der Datenbank speicherte. Immerhin ist es einfacher, die Überschriften für eine neue Sprache in eine Datenbank-Tabelle einzupflegen (oder die für eine bereits existierende zu bearbeiten), als eine neue Dialogvorlage in Visual Studio zu bearbeiten und die Anwendung neu kompilieren zu müssen. Doch ganz so einfach ist es nicht: Einige Frameworks – und darunter auch das eingesetzte Windows Forms – betrachten die Lokalisierung in etwas breiterem Kontext und beziehen zusätzlich zum Text auch einige weitere Eigenschaften mit ein – etwa die Positions- und Größen-Angaben der einzelnen Controls.

Wenn man bei der Suche nach einer Lösung nicht auf Dienste des visuellen WinForms Designer in Visual Studio verzichten möchte, wird die Lösung des Problems noch zusätzlich durch die Tatsache erschwert, dass eben dieser Designer im Endeffekt als Code-Generator arbeitet, mit eher wenigen Möglichkeiten für den Entwickler, sich in den Prozess einzuklinken. Aber mit ein paar Tricks wurde auch diese Hürde gemeistert und alle beschriebenen Plattform-Verschiedenheiten zu einem einheitlichen Ganzen verschmolzen.

Test und Go-Live

Nach der Benutzeroberflächen-Gestaltung und -Logik kommt die Test-Phase: Die Anwender konnten in Ingolstadt und Wolfsburg bereits die mobile Masken produktiv nutzen und sind sehr zufrieden: Die Applikation funktioniert sehr gut und sieht hübsch aus. Die Performance ist auch in Ordnung – kein sichtbarer Unterschied zum früheren Forms-System. Und weil die Benutzeroberfläche nicht strukturell geändert, sondern modernisiert und teilweise vereinfacht wurde, konnten die Anwender sich gleich im neuen System zurechtfinden.

Es war wirklich ein sehr sportliches Projekt und es hat sich gelohnt. Die Anwendung wurde schnell geliefert und man hat erreicht, was man erreichen wollte: Dass die Endbenutzer die gleichen Funktionalitäten auch mit moderneren Mobile-Geräten bedienen können und die gesamte Oracle-Anwendung nun ein neues Leben hat.

Madi Serban, Mathias Waedt,
Rastislav Misecka, Bahar Us
mserban@pitss.de

David Christmann
david.christmann@volkswagen.de

Hidden Secrets: I/O-Durchsatz-messung mit Datenbank-Werkzeugen

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

Die Implementierung eines I/O-Subsystems mit einem hohen Durchsatz ist integraler Bestandteil der Infrastruktur für eine zeitgemäße Applikation, sowohl im Data-Warehouse- als auch im OLTP-Umfeld. Defizite im Design des Gesamtsystems wie zum Beispiel eine zu geringe Anzahl eingesetzter Festplatten oder eine nicht ausreichende Netzwerk-Bandbreite machen sich dann im laufenden Betrieb in Form schlechter Antwortzeiten unangenehm bemerkbar.

Um den Kunden zur Vermeidung von Performance-Engpässen eine Hilfe zu geben, hat Oracle einen Architektur-Blueprint entwickelt, der eine auf allen Ebenen ausgewogene System-Landschaft beschreibt. Nach den Prinzipien dieser sogenannten „Well-balanced Architecture“ ist unter anderem die Exadata Database Machine aufgebaut. Den ausgewogenen Idealzustand findet der DBA jedoch in historisch gewachsenen Umgebungen häufig nicht vor; vielmehr hat er für Anwenderklagen über mangelhafte Antwortzeiten Ursachen und Lösungen zu finden.

Der Prozess der Problemlösung beginnt üblicherweise mit der Wait-Event-Analyse eines AWR-Reports. Hierbei deutet sich durch erhöhte Werte für I/O-bezogene Wait-Events (Wait Class „User I/O“ oder „System I/O“) ein Engpass im I/O-Durchsatz an. Um diesen Ansatz weiterzuverfolgen, ist es notwendig, die maximale Rate der I/O-Operationen der Datenbank zu messen, die diese zuverlässig bereitstellen kann. Dieser Vor-

```
[oracle@sccloud034 bin]$ more mytest.lun
/opt/oracle/oradata/CONT_FS/datafile/o1_mf_sysaux_93jpxn9t_.dbf
/opt/oracle/oradata/CONT_FS/datafile/o1_mf_system_93jq0yl9_.dbf
/opt/oracle/oradata/CONT_FS/datafile/o1_mf_undotbs1_93jq6wd7_.dbf
/opt/oracle/oradata/CONT_FS/datafile/o1_mf_users_93jq6rmc_.dbf
/opt/oracle/oradata/CONT_FS/datafile/o1_mf_sysaux_93jqbkf2_.dbf
/opt/oracle/oradata/CONT_FS/datafile/o1_mf_system_93jqbkl7_.dbf

[oracle@sccloud034 bin]$ ./orion -run simple -testname mytest -num_disks 6 -hugenotneeded
ORION: ORacle IO Numbers -- Version 12.1.0.1.0
mytest_20131002_1222
Calibration will take approximately 44 minutes.
.....
[oracle@sccloud034 bin]$
```

Listing 1: Einfache I/O-Durchsatz-Messung mit ORION

```
[oracle@sccloud034 bin]$ more mytest_20131002_1222_summary.txt
ORION VERSION 12.1.0.1.0

Command line:
-run simple -testname mytest -num_disks 6 -hugenotneeded

These options enable these settings:
Test: mytest
Small IO size: 8 KB
Large IO size: 1024 KB
IO types: small random IOs, large random IOs
Sequential stream pattern: one LUN per stream
Writes: 0%
Cache size: not specified
Duration for each data point: 60 seconds
Small Columns:,      0
Large Columns:,      0,      1,      2,      3,      4,      5,      6,      7,      8,
9,     10,     11,     12
Total Data Points: 43

Name: /opt/oracle/oradata/CONT_FS/datafile/o1_mf_sysaux_93jpxn9t_.dbf   Size: 1184899072
Name: /opt/oracle/oradata/CONT_FS/datafile/o1_mf_system_93jq0yl9_.dbf   Size: 828383232
Name: /opt/oracle/oradata/CONT_FS/datafile/o1_mf_undotbs1_93jq6wd7_.dbf Size: 94380032
Name: /opt/oracle/oradata/CONT_FS/datafile/o1_mf_users_93jq6rmc_.dbf    Size: 5251072
Name: /opt/oracle/oradata/CONT_FS/datafile/o1_mf_sysaux_93jqbkf2_.dbf   Size: 639639552
Name: /opt/oracle/oradata/CONT_FS/datafile/o1_mf_system_93jqbkl7_.dbf   Size: 262152192
6 files found.

Maximum Large MBPS=97.70 @ Small=0 and Large=3

Maximum Small IOPS=7100 @ Small=16 and Large=0
Small Read Latency: avg=2249 us, min=342 us, max=85196 us, std dev=1336 us @ Small=16 and Lar-
ge=0

Minimum Small Latency=1182.64 usecs @ Small=7 and Large=0
Small Read Latency: avg=1183 us, min=291 us, max=181059 us, std dev=2216 us @ Small=7 and Lar-
ge=0
......
[oracle@sccloud034 bin]$
```

Listing 2: Zusammenfassung ORION-I/O-Durchsatzmessung

gang heißt „calibration“. Dessen Ziel hängt natürlich auch vom Lastprofil ab, mit der die Datenbank betrieben wird:

- OLTP-Last mit Fokus auf IOPS und Latenz
- DWH-Last mit Fokus auf I/O-Durchsatz

Es gibt seit der Datenbank 11*g*R2 zwei verschiedene Möglichkeiten der „calibration“, die voll unterstützt sind und auch in der aktuellen Version 12c nutzbar sind. Mit vorgestellten Verfahren können I/O-Engpässe in bestehenden Umgebungen aufgezeigt oder auch die I/O-Spezifikation einer neuen Systemlandschaft verifiziert werden. Am Ende steht ein Vergleich beider Ansätze:

- „calibration“ mithilfe eines unabhängigen Utilitys (ORION)
- „calibration“ mithilfe der Oracle-Datenbank

I/O-Durchsatzmessung mit ORION

Bis zur Version 10*g* war der DBA aus schließlich auf die Nutzung eines unabhängigen Utilitys angewiesen, das für die eingesetzte Plattform und die Datenbank-Version vorliegen und installiert werden muss. Seit 11*g*R2 gehört zu diesem Zweck das Tool ORION zum Standardumfang der Datenbank und wird somit auch voll unterstützt. Es erzeugt unabhängig von der Datenbank eine synthetische Last auf dem Speichersystem. Diese Last entspricht von der Charakteristik, also der Verteilung und Art der I/O-Operationen sowie den genutzten Betriebssystem-Aufrufen, einer Last, die eine laufende Datenbank erzeugt.

```
SQL> COLUMN name         FORMAT a40
SQL> COLUMN value        FORMAT a15
SQL> COLUMN ts_name      FORMAT a10
SQL> COLUMN container    FORMAT a10
SQL> COLUMN asynch       FORMAT a10
SQL> SELECT name
  2  ,      value
  3  FROM   v$parameter
  4  WHERE name IN (,timed_statistics‘
  5                ,‘filesystemio_options‘
  6                ,‘disk_asynch_io‘);

NAME                                 VALUE
------------------------------------ ---------------
timed_statistics                     TRUE
filesystemio_options                 SETALL
disk_asynch_io                       TRUE

SQL> SELECT c.con_id     CON_ID
  2  ,      c.name       CONTAINER
  3  ,      t.name       TS_NAME
  4  ,      d.name       NAME
  5  ,      i.asynch_io  ASYNCH
  6  FROM  v$containers  c
  7  ,     v$datafile    d
  8  ,     v$tablespace  t
  9  ,     v$iostat_file i
 10  WHERE TO_NUMBER(c.con_id) = TO_NUMBER(d.con_id)
 11  AND   TO_NUMBER(c.con_id) = TO_NUMBER(t.con_id)
 12  AND   d.ts#               = t.ts#
 13  AND   d.file#             = i.file_no
 14  AND   i.filetype_name     = ,Data File‘
 15  ORDER BY c.con_id
 16  ,        t.name;

CON_ID CONTAINER  TS_NAME  NAME                                 ASYNCH
------ ---------- -------- ------------------------------------ --------
     1 CDB$ROOT   SYSAUX   /opt/.../o1_mf_sysaux_93jpxn9t_.dbf   ASYNC_ON
     1 CDB$ROOT   SYSTEM   /opt/.../o1_mf_system_93jq0yl9_.dbf   ASYNC_ON
     1 CDB$ROOT   UNDOTBS1 /opt/.../o1_mf_undotbs1_93jq6wd7_.dbf ASYNC_ON
     1 CDB$ROOT   USERS    /opt/.../o1_mf_users_93jq6rmc_.dbf    ASYNC_ON
     2 PDB$SEED   SYSAUX   /opt/.../o1_mf_sysaux_93jqbkf2_.dbf   ASYNC_ON
     2 PDB$SEED   SYSTEM   /opt/.../o1_mf_system_93jqbkl7_.dbf   ASYNC_ON
......
     5 SAMPLEPDB2 EXAMPLE  /opt/.../o1_mf_example_93oyhw77_.dbf  ASYNC_ON
     5 SAMPLEPDB2 SYSAUX   /opt/.../o1_mf_sysaux_93oyhw0w_.dbf   ASYNC_ON
     5 SAMPLEPDB2 SYSTEM   /opt/.../o1_mf_system_93oyhwbj_.dbf   ASYNC_ON
     5 SAMPLEPDB2 USERS    /opt/.../o1_mf_users_93oyhwg5_.dbf    ASYNC_ON

17 rows selected.

SQL>
```

Listing 3: Prüfung der Voraussetzungen für „dbms_resource_manager.calibrate“

```
SQL> SET SERVEROUTPUT ON
SQL> DECLARE
  2   lat  INTEGER;
  3   iops INTEGER;
  4   mbps INTEGER;
  5  BEGIN
  6   --DBMS_RESOURCE_MANAGER.CALIBRATE_IO(, , iops, mbps, lat);
  7   DBMS_RESOURCE_MANAGER.CALIBRATE_IO (6, 10, iops, mbps, lat);
  8   DBMS_OUTPUT.PUT_LINE (,max_iops = , || iops);
  9   DBMS_OUTPUT.PUT_LINE (,latency  = , || lat);
 10   DBMS_OUTPUT.PUT_LINE (,max_mbps = , || mbps);
 11  END;
 12  /
max_iops = 6699
latency  = 10
max_mbps = 98

PL/SQL procedure successfully completed.

SQL>
```

Listing 4: I/O-Durchsatz-Messung mit „dbms_resource_manager.calibrate"

Nach Durchführung der Messung hat der DBA einen Richtwert für den Durchsatz der Hardware. Die Mess-Ergebnisse können benutzt werden, um Fehlerquellen abseits der Datenbank auszuschließen oder eine neue Hardware auf deren Eignung hin zu überprüfen. ORION lässt sich für zahlreiche Einsatzmöglichkeiten parametrisieren, an dieser Stelle soll nur ein kurzes Beispiel gezeigt werden.

Die zur Verfügung stehenden LUNs müssen in einem File angegeben werden, das den Namen des durchzuführenden Tests tragen muss. Anschließend wird in dem Beispiel in Listing 1 ein einfacher Test mit Default-Parametern aufgerufen, wobei der Parameter „–hugenotneeded" anzeigt, dass auf der Testumgebung keine „huge pages" zur Verfügung stehen. Der Test fand auf einer virtualisierten Oracle-Linux-Plattform mit sehr schmalen Ressourcen statt. Das Ergebnis ist dann in einer Datei zusammengefasst, von der Listing 2 einen Ausschnitt zeigt. Zusätzlich werden noch weitere Ergebnisdateien (in diesem Beispiel „mytest_<date_time>_mbps.csv", „mytest_<date_time>_iops.csv" und „mytest_<date_time>_lat.csv") erzeugt, die im „csv"-Format vorliegen und nach erfolgter Bearbeitung in MS Excel die Messungen visualisieren.

I/O-Durchsatzmessung mit dem Resource Manager

Ab Oracle Database 11.1 lässt sich die I/O-Durchsatzmessung auf Ebene des Root Containers durchführen. Das Werkzeug zur „calibration" des I/O-Systems steht als Erweiterung des bewährten Database Resource Manager bereit. Das API „DBMS_RESOURCE_MANAGER.CALIBRATE_IO()" erzeugt eine gemischte „Read-only"-Last, bestehend aus folgenden Aktionen:

- Zufällige I/Os in der Größe der parametrisierten „db_block_size"
- Sequenzielle I/Os mit 1 MByte Blockgröße

Da die Prozedur „CALIBRATE_IO()" den Oracle Call-Stack nutzt und die I/O-Operationen gegen Blöcke laufen, die in der Datenbank gespeichert sind, kann man die erzielten Mess-Ergebnisse als sehr realistisch für die tatsächlich erreichbare Performance ansehen. Da „calibration" möglicherweise eine starke Belastung der Datenbank darstellt, ist es sinnvoll, diesen Vorgang zu Zeiten durchzuführen, an denen die Datenbank sehr gering ausgelastet ist. Zu beachten ist auch, dass Datenbanken, die auf den gleichen Speicher zugreifen und zum Zeitpunkt der „calibration" aktiv sind, das Mess-Ergebnis verfälschen können. Um „calibration" mit dem Database Resource Manager nutzen zu können, müssen verschiedene Voraussetzungen eingehalten werden:

- Eingesetzte Datenbank-Version ab 11.1
- Mit Datenbank-Version 12*c* auf „cdb$root"-Container
- Der aufrufende Benutzer hat „SYSDBA"-Privileg
- Asynchrones I/O ist aktiviert („DISC_ASYNCH_IO=TRUE" und „FILESYSTEMIO_OPTIONS=SETALL")
- Zum Zeitpunkt der „calibration" geringe Last auf der Datenbank

Der Initialisierungsparameter „DISC_ASYNCH_IO" ist bereits standardmäßig auf den Wert „TRUE" gesetzt, während der Standardwert für den Parameter „FILESYSTEMIO_OPTIONS" abhängig vom eingesetzten Betriebssystem automatisch auf den optimalen

```
SQL> SELECT * FROM v$io_calibration_status;

STATUS        CALIBRATION_TIME                  CON_ID
------------- --------------------------------- ------
READY         01-OCT-13 04.20.35.397 PM              0

SQL> SELECT start_time
  2  ,       end_time
  3  ,       max_iops           IOPS
  4  ,       max_mbps           MBPS
  5  ,       latency            LAT
  6  ,       num_physical_disks DISKS
  7  FROM dba_rsrc_io_calibrate;

START_TIME                END_TIME                    IOPS MBPS
LAT DISKS
------------------------- ------------------------- ---- ----
--- -----
01-OCT-13 04.10.04.485273 01-OCT-13 04.20.35.397176 6699   98
10     6


SQL>
```

Listing 5: Ergebnis I/O-Durchsatz-Messung im Data Dictionary

	dbms_resource_manager	ORION
PRO	Unterstützung von RAC Verfügbar mit der Datenbank	Stand-alone-Betrieb möglich Grafische Aufbereitungsmöglichkeit
CONTRA	Benötigt laufende Datenbank	Calibration eines RAC nur manuell

Tabelle 1

Wert für diese Plattform gesetzt wird. Im Zweifelsfall ist an dieser Stelle die plattformspezifische Dokumentation zu Rate zu ziehen. „FILESYSTEMIO_OPTIONS“ kann folgende Werte annehmen:

- *asynch*
  Asynchronous I/O wird verwendet, sofern das Betriebssystem das unterstützt.
- *directIO*
  Direct I/O wird verwendet, sofern das Betriebssystem das unterstützt. Direct I/O umgeht den Unix-Buffer-Cache.
- *setall*
  „Asynchronous“ und „Direct I/O“ wird aktiviert.
- *none*
  Deaktiviert „Asynchronous“ und „Direct I/O“. Oracle benutzt normale „synchronous writes” ohne „Direct I/O“-Options.

Die Voraussetzungen lassen sich durch Abfrage des Data Dictionary verifizieren (siehe Listing 3). Anschließend wird „calibration“ durch Aufrufen des API in einem kleinen PL/SQL-Block gestartet. Das API benötigt dazu zwei Parameter als Eingabe:

- *NUM_DISKS*
  Die Anzahl der Platten, die der Datenbank zur Verfügung stehen. Hier ist zu beachten, dass bei ASM-Nutzung lediglich die physikalischen Platten anzugeben sind, die für Daten genutzt werden. Platten für eine „Flash Recovery Area“ sind also auszusparen.
- *MAX_LATENCY*
  Die maximale Latenz in Millisekunden für einen Plattenzugriff.

Mit diesen Informationen aufgerufen, bekommt man nach kurzer Zeit das Ergebnis der „calibration“ angezeigt (siehe Listing 4).

Der Status der Messung ist in der View „V$IO_CALIBRATION_STATUS“ festgehalten, was dem DBA auch während einer laufenden Messung einen Überblick verschafft. Nach Ende der „calibration“ ist das Ergebnis in der Data-Dictionary-View „DBA_RSRC_IO_CALIBRATE“ festgehalten und kann jederzeit abgefragt werden (siehe Listing 5).

Fazit

Mit den beiden vorgestellten Methoden besitzt der DBA ein Portfolio, aus dem er das geeignete Utility auswählen kann. Tabelle 1 gibt eine Auswahlhilfe zur Abgrenzung beider „calibration“-Tools.

Die Erweiterung des Database Resource Manager stellt ein sehr wertvolles Tool dar, um die Einschränkungen der vorliegenden I/O-Architektur zu verstehen. Nach Beendigung einer „calibration“ verfügt der DBA über die Informationen, die notwendig sind, um die Größe und das Design des I/O-Systems anforderungsgerecht zu gestalten.

Weiterführende Informationen

- http://www.oracle.com/webfolder/technetwork/de/community/dbadmin/tipps/io_calibration/index.html
- http://www.oracle.com/webfolder/technetwork/de/community/dbadmin/tipps/orion/index.html
- https://support.oracle.com/CSP/main/article?cmd=show&type=NOT&id=1297112.1

Frank Schneede
frank.schneede@oracle.com

Dr. Frank Schönthaler
Leiter der Business Solutions Community

Die DOAG 2013 Applications Konferenz + Ausstellung

280 Anwender trafen sich vom 9. bis zum 11. Oktober 2013 in Berlin auf der führenden Oracle-Applications-Konferenz in Europa zu informativen Fachvorträgen, anregenden Diskussionen und interessanten Gesprächen.

Thematische Schwerpunkte der dreitägigen Konferenz bildeten Strategien, Geschäftsprozesse, Applikationen und die zugrunde liegenden Oracle-Technologien, vor allem mit Blick auf den deutschsprachigen Raum. Die Konferenz rund um die Oracle Business Solutions fand auch in diesem Jahr wieder in Kooperation mit internationalen Oracle-Anwendergruppen statt. Aus Management-, Anwender- und IT-Sicht wurden aktuelle Fragen behandelt, Konzepte diskutiert und Praxislösungen vorgestellt.

In einer Zeit des gesellschaftlichen und ökonomischen Wandels sind intelligente Business Solutions gefordert. Den Konferenz-Teilnehmern wurde gezeigt, dass Oracle über eine sehr innovative Technologie und funktional reichhaltige Business-Applikationen verfügt, auf der anderen Seite allerdings noch große Anstrengungen notwendig sind, um diese Tatsache auch dem deutschsprachigen Markt zu vermitteln. So gilt es ein enormes Potenzial im Unternehmenssoftware-Markt gemeinsam zu nutzen. Dr. Dietmar Neugebauer, Vorstandsvorsitzender der DOAG: „Es kommen große Herausforderungen sowohl auf die Kunden als auch auf den Hersteller zu. Der Einsatz der Fusion Applications spielt insbesondere auf dem deutschsprachigen Markt derzeit noch keine bedeutende Rolle."

Ein Highlight der Konferenz war der erstmalig stattfindende Business-Intelligence-Track. Längst als volkswirtschaftlicher Produktionsfaktor etabliert, hat sich die Ressource „Wissen" in vielen Branchen zum wettbewerbsentscheidenden Erfolgsfaktor entwickelt. Wissen entscheidet über die Wirksamkeit von Unternehmensstrategien, über Prozess-, Service- und Produkt-Qualität und ist Treiber für ein zeitgemäßes Human Capital Management. Dieses weite Spektrum wurde exzellent durch Vorträge aus den folgenden Bereichen gebündelt: Multidimensionale Online-Analyse mit Essbase, Information Discovery mit Oracle Endeca, Big Data, Mobile BI und Oracle BI Foundation. Fach- und Führungskräfte aus Business und IT sowie IT-Consultants, die sich mit der Entwicklung und Administration von Oracle-BI-Technologien befassen, schöpften hier aus dem Vollen.

Zusammenfassend informierte die Konferenz ihre Teilnehmer über bewährte Lösungen, Geschäftsprozesse und Anwendungsszenarien. Keynotes, Fachvorträge und Workshops zeigten, wie diese Lösungen mittels Oracle-Applikationen und -Technologien implementiert sowie im Unternehmen eingeführt und genutzt werden können. In der begleitenden Fachausstellung stellten Software-, Lösungs- und Service-Anbieter ihre Leistungsfähigkeit unter Beweis. Moderne Networking-Elemente, aktive Präsentationsformen und ein tolles Rahmenprogramm rundeten die Veranstaltung ab und machten sie auch in diesem Jahr zu einem unvergesslichen Erlebnis für die gesamte Oracle Business Solutions Community.

Ausblick:

Die DOAG 2014 Applications findet vom 21. bis 23. Oktober 2014 statt.

Impressum

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

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

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

Redaktion:
Fried Saacke, Carmen Al-Youssef, Mylène Diacquenod, Dr. Dietmar Neugebauer, Christian Trieb, Dr. Frank Schönthaler

Titel, Gestaltung und Satz:
Alexander Kermas &
HEILMEYERUNDSERNAUGESTALTUNG

Titelfoto: Štěpán Kápl / Fotolia.com
Foto S. 9: © adinas / Fotolia.com
Foto S. 13: © Thomas Jansa/ Fotolia.com
Foto S. 27: © Minerva Studio / Fotolia.com
Foto S. 41: © Sashkin / Fotolia.com
Foto S. 47: © marigold88 / Fotolia.com
Foto S. 60: © Oracle / www.oracle.com

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

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

Wir begrüßen unsere neuen Mitglieder

Persönliche Mitglieder

Jörg Lang
Andreas Fischer
Max Leythäuser
Arne Weyres
Thomas Bauer
Volker Klages
Stefan Demmel
Oliver Herges
Arne Knobel
STephan Kiewald
Thomas Bott
Holger Kalinowski
Susanne Schneider
Nabil Antar
Sam Winchester
Stefan Werner
Dirk Krautschick
Gerhard Zlotos
Steffen Gruner
Rainer Lang

Firmenmitglieder

Erftverband Bergheim, Wolfgang Ullmann
iSYS Software GmbH, Michael Sailer
Silbury IT Solutions Deutschland GmbH, Markus Neubauer

09.12.2013
Regionaltreffen München/Südbayern
Daten- und Datenbanksicherheit
Franz Hüll, Andreas Ströbel
regio-muenchen@doag.org

09.12.2013
Regionaltreffen Bielefeld/Münster
Andreas Kother, Klaus Günther
regio-osnabrueck@doag.org

10.12.2013
Regiotreffen Jena/Thüringen
Data Warehouse Best Practices / Zeichensätze in Oracle-Datenbanken
Jörg Hildebrandt
regio-thueringen@doag.org

13.12.2013
DOAG Webinar: emcli, das unterschätzte Tool für Cloud Control
Johannes Ahrends, Christian Trieb
sig-database@doag.org

19.12.2013
Regionaltreffen Nürnberg/Franken
André Sept, Martin Klier
regio-franken@doag.org

14.01.2014
Regionaltreffen NRW
Stephan Kinnen, Andreas Stephan
regio-nrw@doag.org

21.01.2014
Regionaltreffen München/Südbayern
Franz Hüll, Andreas Ströbel
regio-muenchen@doag.org

21.01.2014
SIG Middleware
Entscheidungen im Unternehmen: Mit der Oracle Plattform auf dem Weg in die Zukunft der Arbeit
Jan-Peter Timmermann, *Björn Bröhl, Hajo Normann,*
Torsten Winterberg
sig-middleware@doag.org

23.01.2014
Regionaltreffen Stuttgart
Jens-Uwe Petersen
regio-stuttgart@doag.org

23.01.2014
Regionaltreffen Dresden
Helmut Marten
regio-sachsen@doag.org

28.01.2014
Regionaltreffen Hannover
Andreas Ellerhoff
regio-hannover@doag.org

28.01.2014
Regionaltreffen NRW (APEX Community)
Stephan Kinnen, Andreas Stephan, Niels De Bruijn
regio-nrw@doag.org

30.01.2014
DevCamp: moderne Softwareentwicklung im Oracle-Umfeld
München
Andreas Badelt, Christian Schwitalla, Robert Szilinski
office@doag.org

11.02.2014
Regionaltreffen Hamburg / Nord
Jan-Peter Timmermann
regio-nord@doag.org

13.02.2014
Regionaltreffen München/Südbayern
Franz Hüll, Andreas Ströbel
regio-muenchen@doag.org

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

30.01.2014

DevCamp

Let's play together

Thema: Moderne Softwareentwicklung im Oracle-Umfeld

Allianz-Arena, München

DOAG

http://barcamp.doag.org

Gut zu wissen, dass es in der Firma läuft.

Gestalten Sie Ihr Leben sorgenfreier. Und Ihre IT leistungsfähiger. Denn wir haben das richtige Servicemodell für Sie. Von der Pflege und dem Support für Ihre Software und BI-Lösungen über den hochverfügbaren Betrieb Ihrer IT-Infrastruktur bis hin zum Outsourcing oder Cloud-Services. Immer effizient und innovativ. Trivadis ist führend bei der IT-Beratung, der Systemintegration, dem Solution-Engineering und bei den IT-Services mit Fokussierung auf Oracle- und Microsoft-Technologien im D-A-CH-Raum. Sprechen Sie mit uns. www.trivadis.com | info@trivadis.com

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

trivadis
makes IT easier.