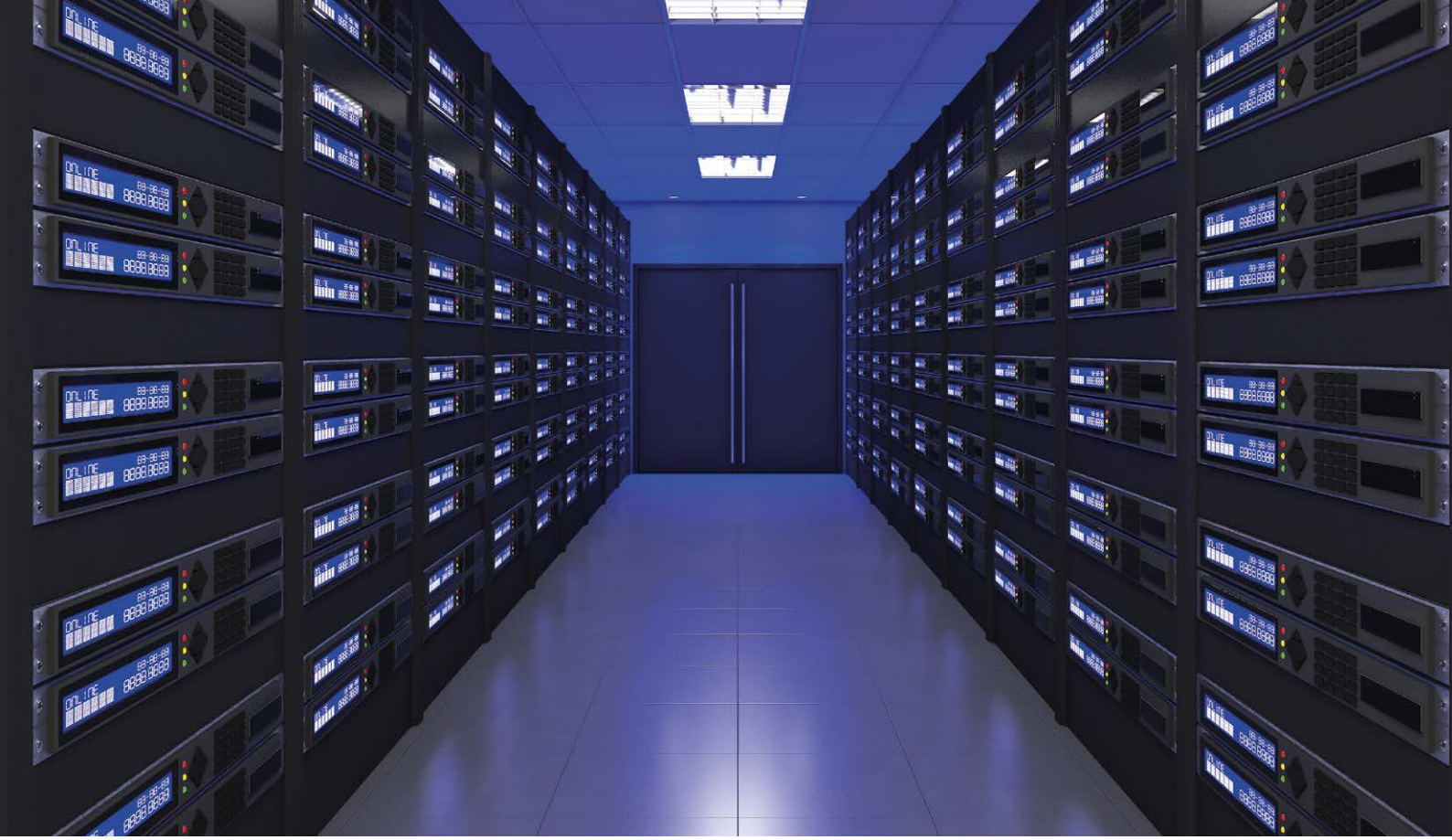

Angewandtes maschinelles Lernen zum automatisierten Management von Oracle-Datenbanken

Mark V. Scardina, Oracle

Rechenzentren wachsen seit jeher, wenn auch nicht mehr im physischen Sinne, doch nach wie vor in ihrer Dichte. Verbesserungen in der Virtualisierung und Hardware haben dazu geführt, dass heute deutlich mehr Systeme betrieben und gemanagt werden als vor wenigen Jahren. Daher werden heute wesentlich mehr Administratoren benötigt.

Im Falle von Datenbanken haben neue Technologien wie die Oracle-Multitenant-Architektur dazu geführt, dass sich die Dichte verzehnfacht hat. Dies ging allerdings nicht mit einem vergleichbaren Anstieg bei der Anzahl der Administratoren einher. Stattdessen müssen sie heute erheblich mehr Datenbanken betreuen, die Effizenz muss also zunehmen.

Das führt zu modernen Strategien auf Basis von Machine Learning, einer Disziplin der Künstlichen Intelligenz, die Gartner als „AIOps“ bezeichnet. Auch wenn diese Abkürzung „Artificial Intelligence for Operations“ bedeutet, definieren die Autoren es mehr als „augmented intelligence“, als „erweiterte“ oder „verbesserte“ Intelligenz: Es geht darum, die Fähigkeiten und Effizienz von Administratoren zu erweitern und zu verbessern, sodass sie mehr Datenbanken bei strengeren SLAs betreiben können.

Oracle hat diesen Bedarf nach mehr Automation und effizientem Management großer Flotten von Datenbanken antizipiert und führte 2015 das Autonomous Health Framework ein. Seitdem

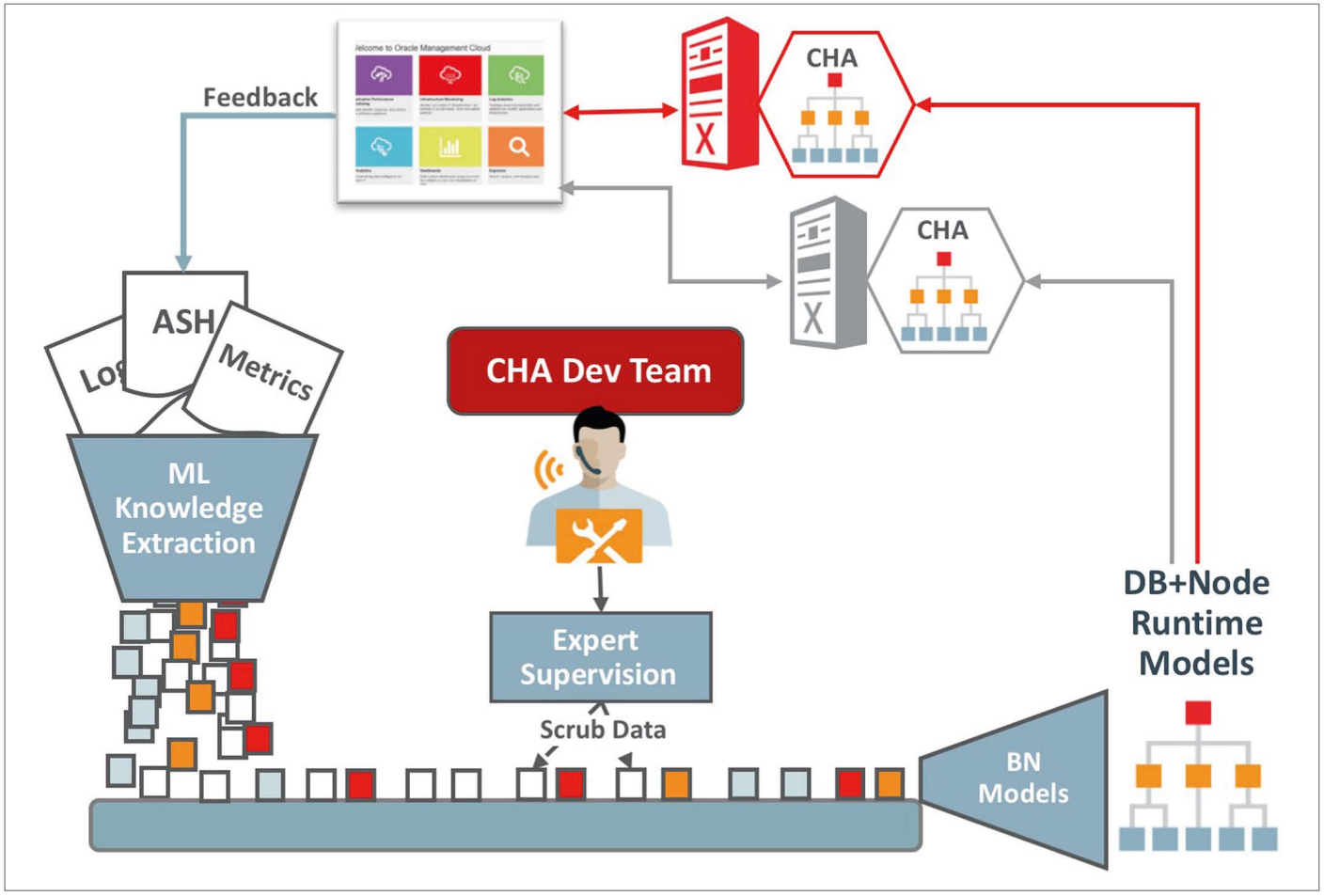

Abbildung 1: Applied Machine Learning Pipeline (Quelle: Mark Scardina)

wurde es kontinuierlich mit Methoden des angewandten maschinellen Lernens erweitert, um Performance zu sichern und schnelle Wiederherstellung im Fehlerfall zu gewährleisten.

Effizienzsteigerungen wurden traditionell durch Zentralisierung und Automatisierung erreicht. Doch gibt es Effekte, die einfache Tasks in großen Umgebungen schwieriger machen, gerade wenn Wartungsfenster immer kleiner werden. Außerdem ist gerade die Netzwerkbandbreite begrenzt, doch viele konventionelle Zentralisierungs-Ansätze führen eine große Menge an Daten, Log-Files und Trace-Files an einem zentralen Ort für Monitoring und Analysen zusammen. Dies ist weder zielführend noch effizient. Zudem sollten in sehr großen Umgebungen Log-Files und Trace-Files nicht aus ihren eigenen Umgebungen entnommen werden, da dies zwangsläufig zu Problemen beim Datenschutz führen kann.

Diese Überlegungen führten zu einem vollständig neuen Ansatz, der zunächst auf die Vermeidung von Problemen und Downtimes (Prävention) fokussiert war. Da es aber auch bei Nutzung von MAA (Maximum Availability Architecture) zu Fehlern kommen kann, wurde die schnelle Wiederherstellung („Rapid Recovery“) als ebenso wichtiges Feld erkannt. Der neue Ansatz muss sowohl umfassende Analysen als auch automatisierte Reaktionen umfassen; nicht zuletzt sind manuelle Tätigkeiten zu vermeiden.

Angewandtes maschinelles Lernen

Aus diesen Gründen hat man sich bei der Entwicklung entschlossen, angewandtes maschinelles Lernen zu verwenden, um sowohl proaktive als auch reaktive Operationen effzient durchzuführen.

Abbildung 1 zeigt Oracles Prozess zur Generierung der notwendigen KI-Modelle. Generell benötigen Diagnosen in IT-Umgebungen eine Root-Cause-Analyse, um den korrekten Lösungsweg zu empfehlen. Einfach nur Cluster in Daten zu identifizieren, reicht vielleicht bei Systemen zur Produktempfehlung in Webshops, aber bei der Unterstützung von operationalen Tätigkeiten bedeutet eine Korrelation nicht unbedingt auch einen Kausalzusammenhang. Deswegen bewerten Experten in einem überwachten Lernprozess die Ergebnisse und überwachen sowohl das Training-Set als auch die resultierenden Modelle. Das maschinelle Lernen findet dabei in der Entwicklung statt, es basiert auf Daten von Oracles SaaS und Cloud-Implementierungen, ebenso wie auf Daten ausgewählter Kunden. Das bedeutet, dass die aufwendige Modellierung bei Oracle stattfindet, während nur fertige Modelle und Wissensbanken im späteren praktischen Einsatz verwendet werden. Dies stellt schnelle Reaktionen mit minimalem Overhead im alltäglichen Einsatz sicher. Alle so erstellten Modelle verfügen über entsprechende Feedback-Mechanismen, um die Qualität der Modelle regelmäßig zu verbessen.

Es ist wichtig zu verstehen, dass die beiden Anwendungsfelder „Prävention“ und „Rapid Recovery“ unterschiedliche Ausgangsdaten verwenden und dass dies zum Einsatz unterschiedlicher Algorith-

men innerhalb des maschinellen Lernens führt. Es gibt keinen „universellen Algorithmus“, der beide Bereiche abdecken kann.

Im Bereich Prävention sind vor allem Echtzeitdaten und -metriken die Ausgangsdaten, beispielsweise aus dem Betriebssystem, der Datenbank, der zugrunde liegenden Hardware.

Es geht darum, in diesen Daten in Echtzeit Auffälligkeiten zu indentifizieren und darauf zu reagieren, um die Verfügbarkeit der Datenbank sicherzustellen. Dabei sind Verfahren wie Mustererkennung, autoassoziative multivariate Regression, Filter unter Berücksichtigung bedingter Wahrscheinlichkeiten und ähnliche Verfahren für Diagnosen und Prognosen im Einsatz.

Für Rapid Recovery sind Log- und Trace-Files wichtige Daten für die Analyse, da man in ihnen anormale Events und Muster von Ereignissen (sogenannte Signaturen) finden kann. Auf Basis dieser Signaturen kann man Incidents und Probleme erkennen und entsprechende Gegenmaßnahmen ableiten, um schnell die Verfügbarkeit wiederherzustellen. Typische Methoden sind hierbei K-nearest Neighbor, TF-IDF (term frequency-inverse document frequency) zur Vorhersage und Mustererkennung, sequenzielles Muster-Mining und Entscheidungsbäume zur Abweichungserkennung sowie LSTM und RNN für Prognose und Diagnose.

Anwendung in der Real-time Prevention

Wen man nach einer ganzheitlichen Lösung sucht, um proaktiv Performance und Verfügbarkeit sicherzustellen, denken Benutzer üblicherweise an Aspekte, mit denen man die eigene Session bewerten könnte: Laufen alle Transaktionen so wie gewünscht? Allerdings ist dieser Ansatz kurzsichtig: Um die Benutzer-Sessions zu sichern, muss die Lösung die zugrunde liegende Datenbank-Instanz sowie zugehörige Infrastruktur überwachen und schützen: Server, Netzwerk und Storage. Daher betrachten die nachfolgenden Use Cases diese drei Elemente.

Zunächst ein einfacher Use Case: Eine Datenbank-Session hängt und blockiert kritische Ressourcen, sodass andere Sessions ebenfalls blockiert werden.

Die Maßnahmen zur Vermeidung solcher Szenarien sind direkt in den Code der Datenbank integriert worden: Erster Bestandteil ist ein Modell normal ablaufender Sessions und darauf basierend Modelle zur Erkennung hängender Sessions. Diese Modelle wurden im Labor entwickelt, mit dem Ziel einer Runtime Engine, die effizent und effektiv hängende Sessions erkennen kann, ohne dass ein Datenbankadministrator benachrichtigt werden muss, der dann die Session aufwendig händisch erkennen und beenden muss.

Diese Engine ist als Code implementiert, der „Hang Manager“ genannt wird und im DAIO-Hintergrundprozess mitläuft. Er macht Snapshots des Zustandes der verschiedenen Sessions und vergleicht über die Zeit, wie sich diese weiter entwickeln. Wenn das Verhalten auf eine hängende Session hindeutet, wird ein Abhängigkeits-Baum aufgebaut, um die ursprünglich blockierende Session zu identifizieren. Wenn diese Root Session erkannt wurde, wird mit Modellen geprüft, ob auch diese hängt. Ist das der Fall, wird sie beendet. Schlägt das fehl, wird der zugehörige Schattenprozess ebenfalls beendet. Dies passiert alles vollautomatisch in weniger als zwei Minuten, ohne dass sein Administrator eingreifen muss. Die Tätigkeit des Hang Manager wird im Alert Log dokumentiert, ein Trace-File wird angelegt. Diese Funktionalität ist seit Version 11gR2 standardmäßig aktiv in der Datenbank und wird seitdem kontinuierlich weiterentwickelt.

Im zweiten Beispiel gehen wir von einer Datenbank-Instanz aus, deren Performance wegen ausgelasteter Ressourcen nicht zufriedenstellend ist.

Für dieses Szenario wurde mittels angewandten maschinellen Lernens ein Modell gängiger RAC-Datenbank-Probleme in konsolidierten Umgebungen erstellt. Weitere Modelle, die auf normale Betriebszustände kalibriert wurden, wer-

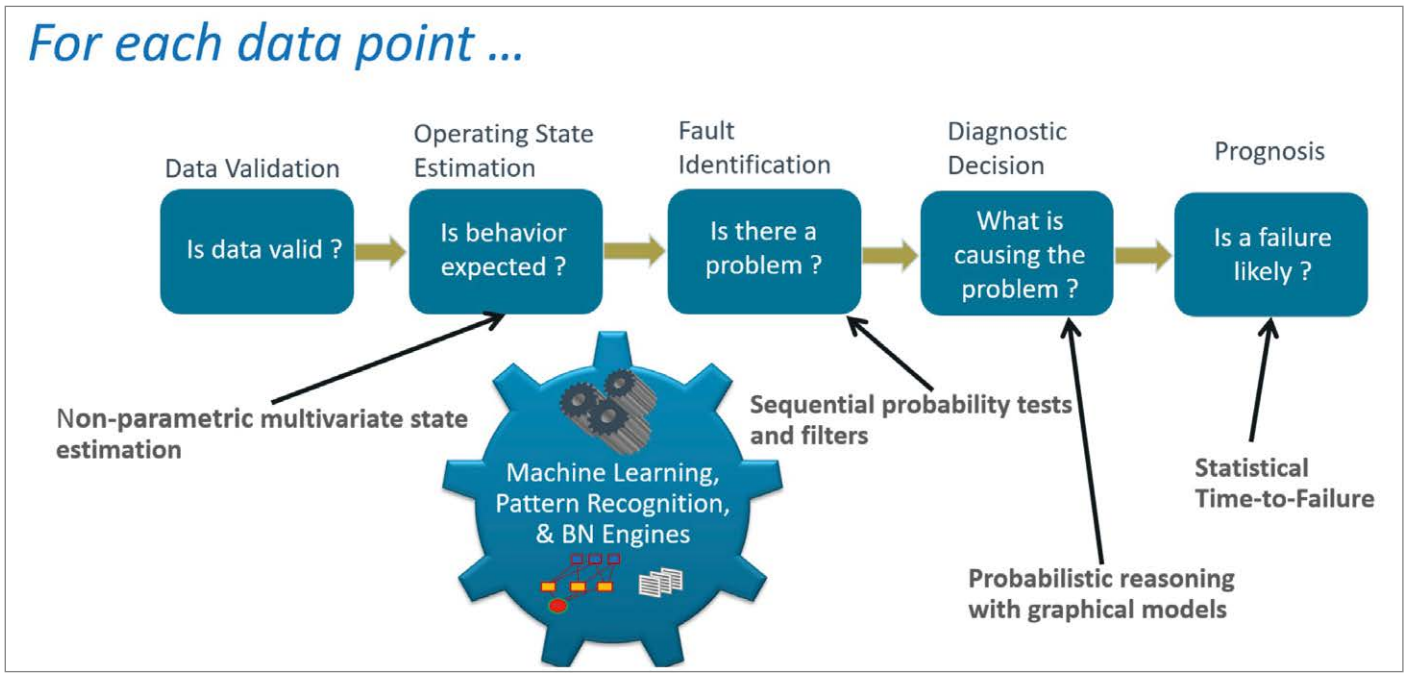

Abbildung 2: Cluster Health Advisor Runtime ML Pipeline (Quelle: Mark Scardina)

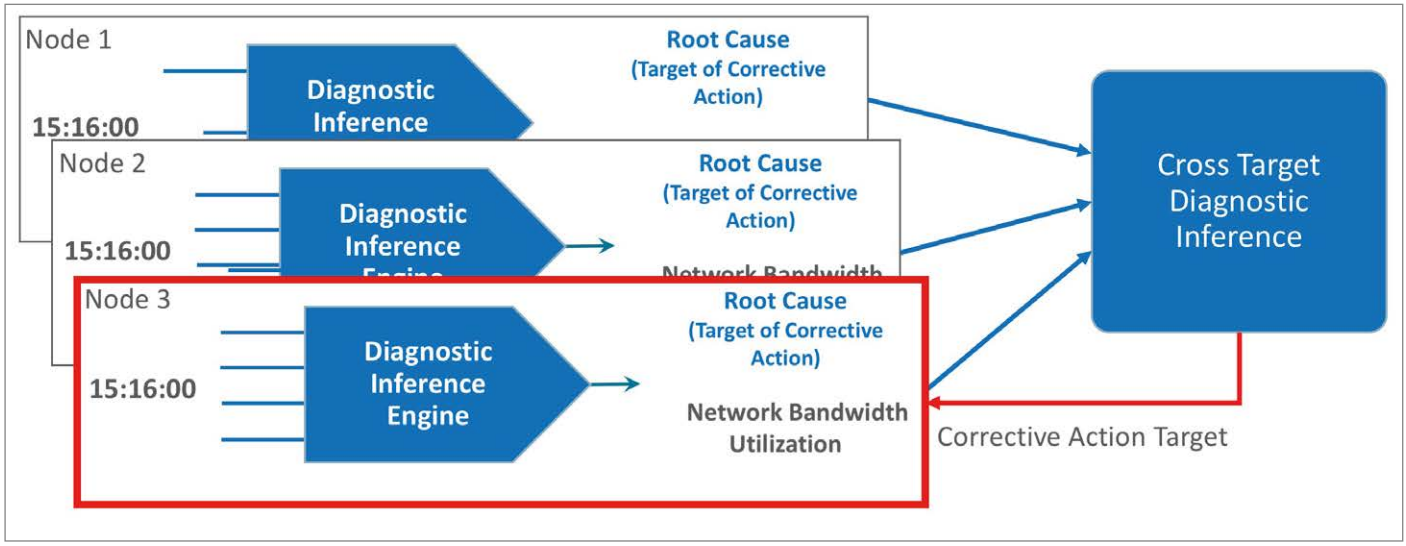

Abbildung 3: Cross Target Diagnostic Inference (Quelle: Mark Scardina)

den genutzt, um früh entstehende Abweichungen zu erkennen. Auf Basis dieser Abweichungen werden dann Situationsspezifische Bayes'sche Netze verwendet, um eine Root-Cause-Analyse durchzuführen und notwendige präventive Maßnahmen zu ergreifen. Das Training dieser Netze fand mit Daten der Oracle Cloud, Oracle SaaS und ausgewählter Kundenumgebungen statt. Die trainierten Modelle werden dann kompiliert eingesetzt für die Echtzeit-Prognose und -Diagnose.

Im Einsatz wurden die Umgebungsdaten von einem schlanken Prozess gesammelt, der auf jedem Knoten eines RAC-Clusters Datenbank- und Betriebssystemdaten aus dem Hauptspeicher ausliest. Im Falle des Betriebssystems wurden die Daten aus dem Cluster-Health-Monitor-Prozess ausgelesen, der diese Daten sowieso bereits erfasst. Im Falle der Datenbank wurden die Monitoring-Daten aus einem Bereich des Shared Memory ausgelesen, in dem Hintergrund-Prozesse bereits Metriken, Wait Events etc. ablegen. Die Daten werden in Intervallen von 5 Sekunden gelesen und ausgewertet, um frühzeitig Signale entstehender Probleme zu erkennen. Sollten diese auftreten, wird der oben beschriebene Prozess zur Root-Cause-Analyse gestartet, Gegenmaßnahmen eingeleitet und die Administratoren alarmiert.

Die dafür notwendige Prozess-Pipeline *(siehe Abbildung 2)* startet mit einem Vektor von 150 Betriebssystem- und Datenbank-Metriken, die in einem 5-Sekunden-Raster mit einer Auflösung von einer Sekunde synchron erhoben werden. Jeder der Vektoren wird zunächst validiert, dann wird geprüft, ob die Werte in einem zu erwartenden Verhältnis zueinanderstehen oder ob es erste Signale gibt, die auf ein Problem hindeuten. Werden Abweichungen erkannt, werden eine weitere Analyse angestoßen und auf Basis der Bayes'schen Entscheidungsnetze Maßnahmen getroffen. Bei bestimmten Problemen, beispielsweise wenn der Ressourcen-Verbrauch relativ hoch ist, kann eine Prognose über die zu erwartende Time-to-Failure abgegeben werden.

Zusätzlich kann es im Umfeld von RAC-Umgebungen dazu kommen, dass die Ursache eines Problems eines Node auf einem anderen Node liegt. Mit der Version 18c wurde die Analyse um eine „Cross Node"-Komponente ausgedehnt, die in *Abbildung 3* dargestellt wird. Diese Komponente erlaubt es Administratoren, die erwähnten möglichen Abbhängigkeiten bei einer Analyse zu berücksichtigen. Die zugrunde liegenden Technologien sind im Cluster Health Advisor bereits seit 12cR2 enthalten.

Als nächstes Beispiel betrachten wir ein Performance-Problem, das von der Infrastruktur ausgeht: Eine ASM-Instanz sei hängen geblieben und blockiere den Datenbank-IO.

Auch hier greift der oben erwähnte Hang Manager, der auch für ASM-Instanzen entsprechend angepasst implementiert wurde. Dieser Hang Manager spricht allerdings zusätzlich permanent mit den angebundenen Datenbank-Instanzen. Sollte ein Prozess nun hängen bleiben, kann entweder die hängende Session oder der gesamte ASM-Prozess terminiert werden. Dies bedeutet nicht, dass die Datenbanken terminiert werden, da sie sich dank FlexASM zu anderen ASM-Instanzen verbinden können.

Ein weiteres häufiges Infrastruktur-Problem im Umfeld von Cloud und konsolidierten Umgebungen ist ein Performance-Einbruch wegen zu hoher Nutzung physischer Ressourcen.

Auch hier wird mit den oben beschriebenen Mechanismen der Mustererkennung gearbeitet: Modelle, die normale Betriebszustände abbilden, überwachen permanent die Metriken von Betriebssystem, Netzwerken und Storage. So werden entstehende Lastspitzen rechtzeitig erkannt und Administratoren informiert. Diese Features sind seit 12cR2 vorhanden und wurden mit 18c deutlich erweitert.

Rapid Recovery Use Cases

Nun ein Blick auf den zweiten Anwendungsbereich: Rapid Recovery, wenn es nach einem Event oder einem Incident einer schnellen Wiederherstellung bedarf. Der erste Use Case betrachtet hier den Fall, dass man schnell wichtige Events in Gigabytes von Log- und Trace-Files finden muss.

Um dieses Problem anzugehen, werden Methoden des maschinellen Lernens angewandt, um aus Log-Files ungewöhnliche Ereignisse herauszufiltern und diese in einer Timeline zu korrelieren und anzuordnen. Die ersten sieben Schritte (Training) der Pipeline in *Abbildung 4* werden

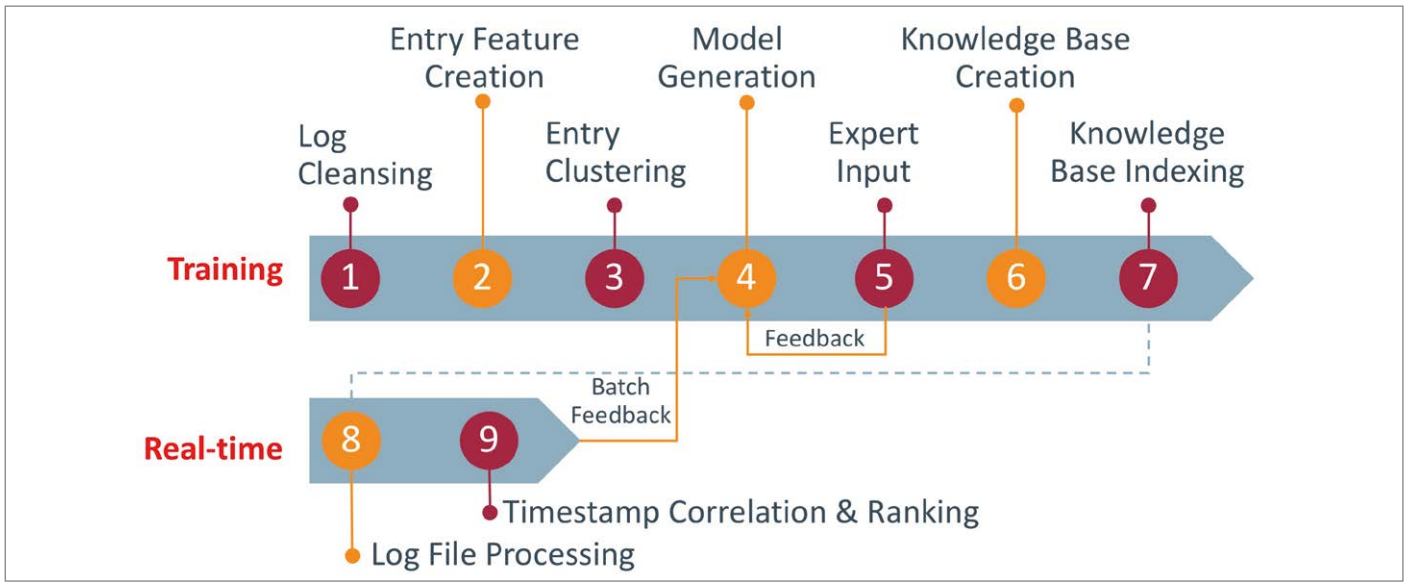

Abbildung 4: Trace File Analyzer ML Pipeline (Quelle: Mark Scardina)

auf Basis der Files durchgeführt, die sich über Jahre des Betriebs angesammelt haben. Als Resultat existieren eine Wissensbasis und ein Index, der dafür genutzt werden kann, in Echtzeit Anomalien und außergewöhnliche Ereignisse aus Log-Files herauszufiltern *(siehe Schritte 8 und 9 in Abbildung 4)*.

Zunächst werden dazu die Log-Files gesäubert und in ihre variablen und konstanten Anteile aufgespalten. Darauf aufbauend werden Merkmale identifiziert und extrahiert, die bei bestimmten Ereignissen auftreten. Dieses Auftreten wird dann geclustert, um die Bedeutung und Wichtigkeit zu bewerten und um letztlich Signaturen bestimmter Ereignisse abzuleiten. Diese Signaturen werden dann von Experten bewertet mit dem Ziel, Entscheidungsbäume zu erzeugen, die Anomalien über die Signaturen erkennen und klassifizieren können. Die Erkennung und Klassifizierung wird durch Feedback-Zyklen mit Trainingsdaten weiter verbessert. Erst dann werden Indizes der Muster für die spätere Erkennung generiert.

Im Echtzeit-Betrieb werden dann Log-Files permanent gegen diese Muster geprüft, um so anormale Ereignisse zu erkennen und diese schnell für weitere Diagnosen und die Lösung der Probleme zu klassifizieren.

Diese Methoden stehen im Trace File Analyzer 18c oder im Oracle Cloud Admin Dashboard für Analysen und Diagnose-Unterstützung zur Verfügung. Die

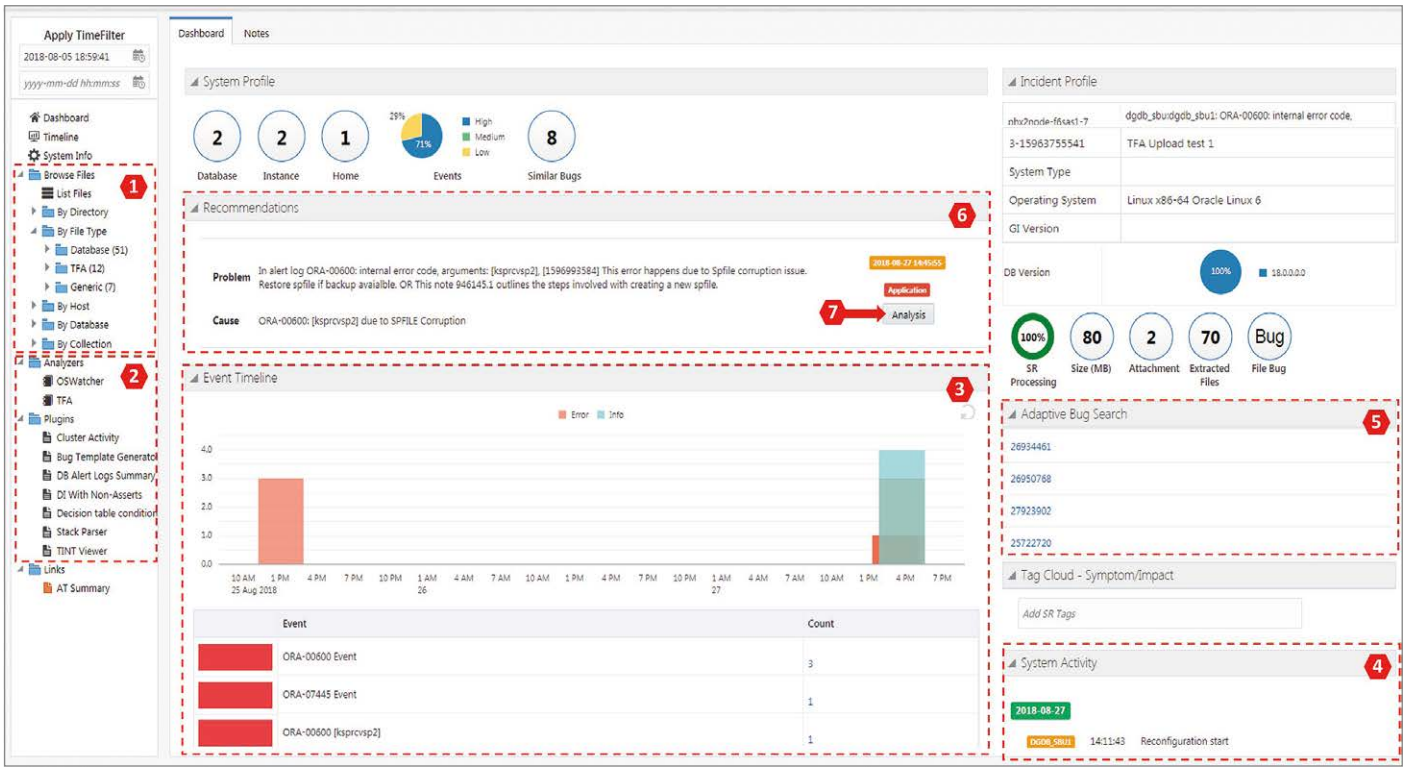

Abbildung 5: Cloud Administrator Incident Dashboard (Quelle: Mark Scardina)

Algorithmen laufen vollständig autonom, sobald der Smart Collector auf dem zu überwachenden System eine Alarmierung auslöst.

Viele Administratoren entwickeln im Laufe der Zeit eine Art Gespür, mit dem sie aufgrund von Einträgen in Log-Files Probleme vorhersehen können. Dies ist mehr als ein Gespür, es basiert auf Fakten, es handelt sich um typische Muster in Log-Files, die auf ein Problem hindeuten. Solche Muster aufzuspüren, ist der zweite Ansatzpunkt für maschinelles Lernen im Kontext von Log- und Trace-Files.

Indem viele Bugs und Service Requests mit Methoden des Data Mining untersucht wurden, konnten wiederholende Muster erkannt und so Zeitreihen sowie zeitliche Signaturen identifiziert werden, die auf das Vorhandensein bekannter Probleme hindeuten oder hinweisen. Diese Muster werden im Trace File Analyzer hinterlegt, sodass dort diese Muster gesucht und erkannt werden können. Dies erlaubt es, unter Umständen vor dem eigentlichen Auftreten eines Problems dieses bereits zu erkennen und korrigierende Maßnahmen zu empfehlen.

Der letzte Use Case zeigt Ihnen einen Anwendungsfall hinter den Kulissen beim Oracle Support: Es geht um die Effizienz, mit der innerhalb der Service-Request- und Bug-DB-Infrastruktur nach relevanten Bugs und Patches gesucht wird. Es kommt häufig vor, dass bestimmte Bugs oder Issues bei Kunden bereits in anderen Fällen gelöst wurden oder Patches schon bereitstehen. In diesen Fällen kommt es darauf an, schnell den jeweiligen Patch oder Workaround zu identifizieren. Liefert man ausreichende Input-Daten, beispielsweise aus speziellen Kollektoren, können schneller relevante Bugs oder ähnliche Service Requests identifiziert werden. Hier greifen die Methoden des Machine Learning, die kontinuierlich SRs sowie die BugDB durchforsten, um identische oder sehr ähnliche Bugs und Incidents zu identifizieren.

Dies wurde für mehr als 400 Oracle-Produkte implementiert, um mithilfe logistischer Regressionen potenzielle Duplikate zu identifizieren und den Support Engineers oder Cloud Admins direkten Zugriff auf diese zu geben. Der Prozess wird kontinuierlich durch das Feedback der Nutzer verbessert.

Die Ähnlichkeit von Bugs und Incidents wird dabei mit Merkmalen aus Error Stacks, Trace-Files, den Beschreibungen etc. abgeleitet. Aus diesen werden sogenannte „Issue Signatures" generiert, welche die Grundlage für Regressionsmodelle zur Bewertung unterschiedlicher Merkmale sind. Auf Basis des so gebildeten gewichteten Modells können dann die Ähnlichkeiten bewertet werden. Das Modell wird kontinuierlich auf Basis des Feedbacks der Benutzer weiterentwickelt.

Momentan sehen nicht nur Administratoren diese Funktionalitäten. Sie wurde auch Entwicklern mit Zugriff auf die Bug-DB zur Verfügung gestellt, um schneller geeignete Lösungen für Bugs zu finden und die Bearbeitungszeiten durch eine kollaborative Bearbeitung zu verkürzen. Ein Einsatz in weiteren Bereichen ist geplant.

Alles aus einem Guss

Abbildung 5 zeigt das Dashboard eines Cloud Admins, auf dem dieser über einen Incident informiert wird. Es wurde so entworfen, dass die Problemlösung schnell und sicher erfolgen kann. Dazu dienen verschiedene Sektionen auf dem Dashboard. In der Sektion 1 werden zunächst die Daten zusammengestellt, die analysiert wurden. Diese sind bereits indiziert und klassifiziert und werden in einer hierarchischen Ansicht mit Drill-Down-Funktionalität bereitgestellt.

Unterhalb davon, in der 2. Sektion, finden sich Diagnose- und Analyse-Plug-ins, die auf Basis der oben beschriebenen KI-Modelle auf die Dokumente in der Sektion 1 angewandt werden. Diese Plug-ins stellen gefundene, wichtige Ereignisse in Form einer Timeline in Sektion 3 dar, von wo man schell und einfach weitere Analysen und Aktionen starten kann.

Die gesammelten und als relevant eingestuften Log-File- und Trace-Informationen werden in Sektion 4 dargestellt, sodass hier nur Auffälligkeiten zu finden sind (und kein Grundrauschen). In Sektion 5 werden relevante, bekannte Bugs dargestellt, die zu den gefundenen Abnormalitäten passen.

Abschließend wird zentral in Sektion 6 das Ergebnis der Root-Cause-Analyse aufbereitet und dargestellt, inklusive der detaillierten, empfohlenen Lösungsschritte, die mit einem Klick durchgeführt werden können (7).

Zusammenfassend bleibt zu erwähnen, dass Oracles Anwendung maschinellen Lernens im Bereich des Operatings und der Administration von Analysten als sinnvoll und zielführend angesehen wird. Gartner *[1]* veröffentlichte beispielsweise eine klare Handlungsempfehlung für CIOs, Methoden der KI in allen Bereich zu nutzen.

Noch spezifischer sind die Empfehlungen der IDC *[2]*: Sie sehen die Anwendung von KI in Produkten wie der Autonomous Database als eine außerordentliche Verbesserung an, die Hunderte, wenn nicht gar Tausende von Stunden an administrativem Aufwand pro Jahr und Datenbank einsparen wird.

Mehr Informationen über das Oracle Autonomous Health Framework finden Sie auf der Webseite und im User's Guide: www.oracle.com/goto/AHF.

Quellen

[1] Predicts 2019: Artificial Intelligence Core Technologies - Chirag Dekate, et al, Gartner, 2019

[2] IDC PERSPECTIVE: Oracle's Autonomous Database: AI-Based Automation for Database Management and Operations - Carl W. Olofson & David Schubmehl, IDC

Mark V. Scardina
mark.scardina@oracle.com