Was erwarten Sie dabei von einem IT-Unternehmen wie Oracle?
Dürre: Die Kunst wäre, die richtige Mischung und/oder Gewichtung zwischen „open“ und „proprietär“ zu schaffen. Die Märkte in der IT wandeln sich schnell und sind nicht mehr von einem Unternehmen allein beherrschbar. So sind technologische Offenheit und strategische Transparenz wünschenswert und bestimmt langfristig auch nützlich für IT-Unternehmen.

Wie sehen Sie den Stellenwert einer Anwendergruppe wie der DOAG?
Dürre: Ich habe ja eigene Siemens-Erfahrungen, als Mitarbeiter wie als Lieferant. Bei Siemens gab es die Siemens Anwender-Vereinigung. Ich war immer beeindruckt, wie darüber sehr konstruktive Beziehungen gerade zwischen Großkunden und Lieferant möglich waren und welcher Nutzen für beide Seiten entstanden ist.

Welche Erfahrungen aus Ihrem langen Berufsleben können Sie an andere Unternehmer weitergeben?
Dürre: Vielleicht die wichtigste: Immer fähig sein, sich ein eigenes Urteil zu bilden. Und agil bleiben und dogmatische Entscheidungen und Handlungen um alles in der Welt verhindern. Ansonsten versuche ich, die Regeln für modernes Management von Hans Ulrich, dem Erfinder des St.-Gallener-Management-Modells, zu beherzigen. Diese sind: Die Unvorhersehbarkeit von Zukunft als Normalzustand zu akzeptieren, die Grenzen des Denkens weiter zu stecken, sich besser vom „sowohl als auch” als vom „entweder oder” leiten zu lassen, mehrdimensional zu denken, Selbstorganisation und Selbstlenkung als Gestaltungsmodell zu begreifen, Managen als sinngebende und sinnvermittelnde Funktion aufzufassen, sich auf das Wesentliche zu konzentrieren und Gruppendynamik auszunutzen. Allerdings stammen diese Regeln aus den 1980ger Jahren.

SQL und Hadoop – eine gemeinsame Zukunft

Jean-Pierre Dijcks, Product Manager Big Data Oracle Corp.

Kein Zweifel, Big Data wird bleiben. In den letzten 18 Monaten avancierte die Thematik zu einem wichtigen Element in jeder Data-Warehouse-Strategie. Dennoch bleiben eine Menge Fragen über den optimalen Weg zur Realisierung einer Big-Data-Architektur und darüber, welche Tools die richtigen für diese Aufgabe sind. Ein interessantes Lösungs-Szenario ist die Integration von SQL und Hadoop. Der Artikel geht auf die Vor- und Nachteile einer solchen Integration ein und zeigt auf, wie solche Lösungen die Data-Warehouse, aber auch die Hadoop-Landschaft verändern.

Data Warehouse und Hadoop – brauchen wir beides? Die spontane Antwort ist ein klares „Ja“. Allerdings sollte zunächst immer die weitergehende Frage nach dem besten Weg zum Erreichen der jeweiligen Geschäftsziele gestellt werden. Zur Konkretisierung ein Beispiel: Ein erfolgreiches Unternehmen braucht eine klar definierte Geschäftsstrategie, um seine Geschäftsziele zu erreichen. Für einen Automobilhersteller mag dies lauten: „Produziere qualitativ hochwertige Autos, die ein angenehmes Fahrgefühl vermitteln und konstant die Erwartungen der Kunden erfüllen“. Doch wie kann das Unternehmen diese Strategie umsetzen? Wie lässt sich die positive Erfahrung hoher Qualität beim Kunden messen, wie kann man das Erfüllen von Erwartungen prüfen oder ein positives Fahrerlebnis feststellen? Unternehmen brauchen dafür Daten aus der eigenen Organisation, aber auch von außerhalb – also fremde Daten.

Das Data Warehouse ist heute oft die Basis, auf der Unternehmen ihre Mission verfolgen: Es integriert Daten aus allen operativen Systemen sowie aus der Fertigung und dem Stammdaten-Management. Erfolgreiche Data-Warehouse- und Business-Intelligence-Anwendungen steuern heute Bedarfsplanung, Lieferanten-Management, Lagerhaltung und das Monitoring von Qualität und Kundenzufriedenheit. Allerdings reichen diese traditionellen Datenquellen und Managementverfahren heute immer weniger aus. Sich ausschließlich auf diese gut strukturierten und über die Jahre gereiften Daten und Informationen zu verlassen, führt zu verpassten Chancen oder zum Nachsehen, wenn Wettbewerber solche Chancen nutzen.

Man denke beispielsweise an das Potenzial und die Auswirkungen dessen, was wir heute „Connected Cars“ nennen. In früheren Fahrzeuggenerationen wurden gerade mal die zurückgelegte Kilometer-Leistung, der Füllstand im Benzintank und der Öldruck gemessen. Es war eine „analoge Welt“. Messwerte außerhalb dieser beschränkten Möglichkeiten zu erhalten, war äußert schwierig oder nur unter besonderen Laborbedingungen möglich. Heute verfügen moderne Fahrzeuge über weit mehr als hundert unterschiedliche Sensoren zur Messung von Zuständen.

Sie ermöglichen einen permanenten Check des Fahrzeugs: Konsistenz und Alter des Motor-Öls, Laufleistung, Geschwindigkeit, Temperatur, Drehzahl, Luftdruck im Reifen, Verwendung elektronischer Geräte durch den Fahrer, die Art der Beschleunigung, Art der Bremsvorgänge, Häufigkeit des Schaltens. Geo-Positionen mit GPS und lokale Wetterdaten bilden weitere Dimensionen möglicher Analysen. Darüber hinaus hinterlassen die Fahrer der Fahrzeuge durch Nutzung mobiler Geräte, Telefonieren, Interaktion über soziale Netzwerke, Nutzen von Diensten oder Teilnahme an Bonusprogrammen eine individuelle Spur persönlicher Verhaltensweisen und Merkmale.

Was bedeutet das für den Automobilhersteller, dem durch diese „Connected Cars" sowieso schon ein riesiges Angebot an Daten zur Verfügung steht? Zunächst sind die Analyseverfahren nicht mehr nur beschränkt auf direkt im Fahrzeug entstehende Daten. Einblicke in reale Lebenssituationen von Kunden, Aufgreifen von Stimmungen und Verhalten im Umgang mit Mobilität beziehungsweise mit dem Fahrzeug sind möglich. Richtig umgesetzt liefert dies ganz neue Ideen und Impulse für die Entwickler neuer Features und Eigenschaften von Fahrzeugen. Zusätzlich kann man die so gewonnenen Erkenntnisse für das Aufstellen von Vorhersagemodellen und zur Steuerung von Bordcomputern nutzen. Fahrzeuge bleiben seltener liegen, weil schon vor Eintreten von Defekten durch vorgezogene Wartungsarbeiten reagiert werden kann.

Die technische Herausforderung des Sammelns und Integrierens von großen Volumen schwach strukturierter Messdaten der „Connected Cars" ist mittlerweile durch die rasch entwickelten, neuen Big-Data-Techniken gelöst worden. Hadoop- und NoSQL-Datenhaltungen bieten heute eine ökonomisch sinnvolle Lösung.

Bei Big Data redet man oft von „Low Value"-Daten. Diagnose-Daten einzelner Fahrzeuge werden je nach Nutzung Tausende Male pro Tag eingesammelt. Wir sammeln alles, was kommt. Einzelne Messdaten gehen jedoch durch Messfehler, Funklöcher, Softwarefehler etc. verloren. Im Vergleich zur Masse der Daten fallen solche vereinzelten „Dirty Data" nicht auf. Sie werden die Gesamt-Tendenz einer Analyse kaum oder gar nicht beeinflussen, die eine Menge von Millionen oder gar Milliarden Einzel-Informationen verwertet. Bei traditionellen „High-Value"-Daten wäre dies ein Drama. Man stelle sich vor, Kontobewegungen einer Bank würden verloren gehen oder bestellte Produkte kämen nicht bei Kunden an, weil der Bestelldatensatz fehlerhaft ist.

Abbildung 1: Die klassische Optimierung

Analyse-Methoden bleiben unverändert

Die klassischen Analyse-Verfahren zur Optimierung von Geschäftsabläufen und Zielerreichung sind gemeinhin bekannt:

- Zunächst formuliert man ein Ziel wie die Verbesserung der Produktqualität
- Man sammelt alle zur Verfügung stehenden Daten, von denen man glaubt, dass man sie braucht und sucht mithilfe von „Predictive Analytics" – also Data-Mining-artiger und heuristischer Zusammenhangssuche – nach Abhängigkeiten in diesen Daten
- Sind Abhängigkeiten gefunden, formuliert man diese als Regeln (sogenannte „Modelle") und wendet sie zur Überprüfung auf neue Datenbestände an
- Sind die Modelle verifiziert, nutzt man sie zur Steuerung von operativen Systemen
- Zur Kontrolle setzt man ein Monitoring auf die Ergebnisse in den operativen Prozessen und misst den Erfolg über die Laufzeit der Anwendung

Das Ganze ist ein immer wiederkehrender Prozess (siehe Abbildung 1). Mit der Zeit verfeinert und verändert sich das Modell – je nach den bereitgestellten Daten und Randbedingungen. In einem Fahrzeug mag sich erst nach Jahren eine Fehlkonstruktion eines Bauteils herausstellen, wenn Erfahrungen unter bestimmten Fahrbedingungen und klimatischen Verhältnissen gesammelt wurden. Nicht nur das Bauteil, sondern auch das zugrunde liegende Modell wird geändert. Die Ingenieure berücksichtigen diese Erfahrungen bei Folgeentwicklungen neuer Bauteile. Klingt aufwändig und zeitintensiv. Da ist es nachvollziehbar, dass man solche Modell-Iterationen optimieren und beschleunigen will.

Hadoop oder relationale Datenbank

Das Hinzufügen von schwach ausgeprägten zusätzlichen Datenmengen ändert nichts an diesem Analyse-Prozess. Man zieht nur andersartige Daten mit heran und sie mögen auch anders verarbeitet werden. Am Ende geht es je-

doch immer darum, die Ergebnisse von Analysen in Modelle zu überführen, um Geschäftsprozesse zu optimieren. Die Frage, wo die technische Verarbeitung stattfindet – in Hadoop oder in einer relationalen Datenbank –, ist von einer Reihe von Faktoren abhängig:

- Entwicklungsreife von Werkzeugen
- Performance der Verarbeitung
- Security-Aspekte
- Fähigkeit, Daten schnell entgegen zu nehmen
- Wirtschaftlichkeit bei der Speicherung von „Low Value"-Daten
- Einfache ETL-Prozesse
- Vollständigkeit der Daten (Machen zu viele Lücken den Datenbestand sinnlos?)
- Unterschiedlichkeit der Datenstrukturen (Variety)
- Komplexität des Daten-Managements

Für einige Faktoren weist jede der beiden Plattformen Stärken auf, potenziell sind sicher alle Faktoren von beiden Plattformen lösbar. Hadoop ist „Schema-los". Die Daten sind also zunächst ohne Struktur abgelegt. Diese Struktur wird erst durch die lesenden Programme erkennbar („Schema on Read"). Diese Flexibilität hat Vorteile:

- Das extrem schnelle Laden der Daten in einen Hadoop Data Store. Die Daten werden praktisch ohne nähe-

ren Struktur-Syntax-Check einfach „abgekippt".
- Wenn sich die Formate der Fahrzeug-Diagnose-Sätze mit der Zeit ändern, dann hat das wenig Einfluss auf die ETL-Strecken. Die Änderungen müssen nur das Lesen berücksichtigen.
- Das Map-Reduce-Framework arbeitet massiv parallel. Die geringe Strukturierung der Daten kommt dieser hohen Parallelisierung sehr entgegen.
- Die überschaubaren Kosten für Storage und Rechenleistung für die zum Teil sehr hohen Datenmengen.

Relationale Systeme hingegen haben diese Stärken:

- Die bestehende Strukturierung ermöglicht „Schema on Write", also Prüfung von Daten im Zuge der Speicherung.
- Es gibt eine hohe Zahl von gut entwickelten Werkzeugen für eine einfache, reibungslose und performante Verarbeitung beziehungsweise Datenanalyse.
- Anwender können ohne besondere Hürden mit einfachen Zugriffen auf einem standardisierten Datenmaterial ad hoc und multidimensional analysieren.
- Die Daten sind in der Regel qualitätsgesichert und die Analyse-Verfahren sind erprobt.

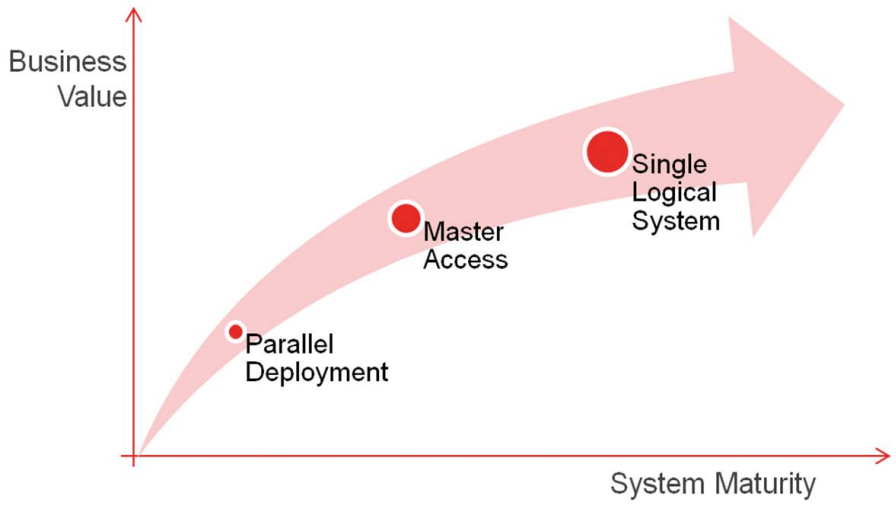

Abbildung 2: Reife-Phasen von Hadoop- und RDBMS-Implementierungen

Die Herausforderung heute lautet also nicht „Können wir alles in Hadoop oder in einem Data Warehouse machen?", sondern „Wie können beide Technologien so verzahnt werden, dass die spezifischen Vorteile der jeweiligen Technologien am besten die Unternehmensziele unterstützen?" Tatsächlich ist die Koexistenz der Technologien nicht nur kurzfristig als Übergangszeit zu sehen. Die vornehmliche Speicherung bestimmter Daten auf der einen oder anderen Technologie wird zwar bleiben, aber die Kombination Data Warehouse/Hadoop wird ein fester Bestandteil künftiger IT-Infrastrukturen werden.

Gemischte Datenablage und Abfragesprache für Big Data

Werden Daten mit beiden Technologien gespeichert, stellen sich die Fragen „Wie sollen Daten in einer gemischten Umgebung, also aus Hadoop und aus dem Data Warehouse, gelesen und verarbeitet werden, ohne dass dies zu doppeltem beziehungsweise für Endbenutzer nicht zumutbarem Aufwand führt?" und „Mit welcher Abfragesprache soll dies geschehen?"

Mit zunehmender Reife der Hadoop-Technologie und der wachsenden Akzeptanz in den Unternehmen nimmt auch der Nutzen einer integrierten Verwendung von Hadoop und Data Warehouse zu (siehe Abbildung 2). Die Hauptgründe sind:

- Reduzierung von Komplexität
- Schnellere Verfügbarkeit der zu analysierenden Daten (Wegfall von Datenbewegungen/ETL)
- Die Möglichkeit, Daten beliebig zu kombinieren beziehungsweise korrelierende Zusammenhänge zu analysieren

Das Adaptieren neuer Technologie erfolgt in den Unternehmen meist in mehreren Phasen. Auf dem Weg zur integrierten Verwendung von Hadoop und des Data Warehouse lassen sich drei Entwicklungsstadien erkennen. Wo die einzelnen Industrien heute stehen, lässt sich beantworten, indem man diese näher betrachtet.

Phase 1: Parallel Deployment

In der ersten Phase erfolgen Hadoop-Implementierungen parallel zu bestehenden RDBMS-Umgebungen. Die Hadoop-Umgebungen sind isoliert. In einer Art „Labor-Situation" kann das Unternehmen den Umgang mit Hadoop erlernen und Best Practices mit allen Stärken und Schwächen von Hadoop erfahren. Aktuell ist eine große Anzahl von Unternehmen in dieser Phase. Es gibt Test-Umgebungen, in denen man Sinn und Zweck von Hadoop für das eigene Unternehmen erforscht. Das Lösen technischer Klippen und spezieller Use-Cases steht im Vordergrund.

Hadoop besitzt ein mächtiges Map-Reduce-Framework, das allerdings für die meisten Mitarbeiter – ob Fachabteilung oder IT – fremd ist. Folglich ist einer der ersten Schritte die Beschäftigung mit Hive oder der HiveQL-Abfragesprache. Hive bietet eine (RDBMS-) Datenbank-analoge Beschreibung von Hadoop-Daten (HDFS) in Tabellen, Spalten und sogar Partitionen. Auf relationalen Metadaten bietet HiveQL eine deklarative, SQL-artige Abfragesprache an, die die Abfragen im Hintergrund in Map-Reduce-Programme umwandelt. Der wichtigste Vorteil von Hive ist die Wiederverwendung der SQL-Skills durch SQL-erfahrene Benutzer, die damit leicht auf Daten in Hadoop zugreifen können. Der große Nachteil (verglichen mit einer traditionellen Datenbank) sind die schlechteren Antwortzeiten von Standard-Abfragen. HiveQL wird über Map-Reduce-Programme abgearbeitet, was Zeit kostet.

Erkennen Unternehmen einen Sinn in der Hadoop-Praxis, entsteht der Wunsch nach einer verstärkten integrierten Anwendung von Hadoop mit bestehenden Datenbank-basierten Anwendungen – abgesehen von spezifischen Anwendungsfällen, in denen ein isoliertes Hadoop-System einen Sinn ergibt. Der Übergang zur Phase 2 ist beschritten, jetzt wird SQL ein wichtiger Bestandteil der Big-Data-Debatte.

Phase 2: Hybrid Master Access

Hat sich ein Unternehmen für den Einsatz von Hadoop entschieden, wird überlegt, wie und in welcher Form Endbenutzer auf diese neue Datenwelt zugreifen und Analysen durchführen. Man benötigt die Möglichkeit, auf beide Datenwelten gleichzeitig zuzugreifen, und erwartet eine Art konsolidierte Sicht, einen sogenannten „Master Data Point". Dieser ist in drei Varianten vorstellbar:

1. Das zentrale Data Warehouse mit klassischen ETL-Prozessen. Hier betrachtet man das Hadoop-System lediglich als weitere Datenquelle.
2. Ein BI-Tool, das Zugriffe auf beide Welten erlaubt und die Konsolidierung im BI-Tool vornimmt.
3. Ein führendes Hadoop-System, das Daten auch aus einem Data Warehouse in sich aufnimmt.

Die meisten Unternehmen, die sich schon sehr früh mit Hadoop-Techniken beschäftigten, befinden sich mittlerweile in dieser zweiten Phase. Pragmatischerweise nutzen sie eine Mischform der drei genannten Varianten. Solche hybriden Lösungen bieten Zugriffe auf ein Subsystem an, während man gleichzeitig Daten über ETL austauscht – es wird vorsortiert und stellenweise aggregiert. Greift beispielsweise ein Teil der Benutzer über Hive auf Hadoop-Daten zu, liest ein großer Teil der übrigen Benutzer über ein BI-Tool Daten aus einem Data Warehouse mit voraggregierten Kennzahlen.

Diese pragmatische Hybrid-Lösung führt natürlich zu dem Nachteil der Komplexität von systemübergreifenden Zugriffen und Daten-Replikationen. Fährt man ein föderatives System, belässt also Daten an ihrer originären Stelle in Hadoop beziehungsweise im Data Warehouse, so heimst man sich „Latency-„Probleme ein: Typische HiveQL-Abfragen sind langsamer als SQL-Abfragen in einem Data Warehouse, ein Join von Daten in beiden Systemen fällt auf die Geschwindigkeit des langsamsten Systems zurück. Das Resultat sind unzufriedene BI-Benutzer.

Oft ist das Data Warehouse der primäre Zugriffspunkt für die Benutzer, weil hier bereits viele wohlsortierte und überprüfte Daten zu finden sind. Die Umgebung ist zudem sicher und erfüllt oft regulatorische Anforderungen. Die Performance ist in der Regel besser als in Hadoop-Systemen.

Hat man jedoch die Masse der Rohdaten im Hadoop-System belassen und nur Extrakte beziehungsweise Aggregate in das Data Warehouse überführt, werden nicht alle Analysen und Anfragen über das Data Warehouse zu lösen sein. Tritt eine solche Situation ein, muss man Rohdaten in dem Hadoop-System nachladen, zusätzliche Filter bemühen und Daten erneut in das Data Warehouse überführen beziehungsweise aggregieren.

Zur Lösung solcher Probleme entstehen immer mehr technische Anwendungen. Auf der Hadoop-Seite werden SQL-Engines entwickelt, die gegenwärtig neu auf den Markt kommen. Diese ersetzen die Map-Reduce-Komponente und führen neue Optimizer zur Plan-Ausführung von Abfragen ein. Solche Anwendungen sind jedoch erst in ihrer frühen Entwicklungsphase und verfügen noch nicht über gewohnte Features von eingeführten RDBMS-Datenbanken. Sie schließen schon gar nicht die Lücke zwischen der Hadoop-Welt und einem Data Warehouse. Sie sind tauglich als Einzelzugriffs-Tools innerhalb der Hadoop-Welt.

Mittlerweile verfügen auch BI-Tools über einen direkten Hive-Zugriff und können Hadoop-Daten in ihrem Server-Cache für begrenzte Analysen vorhalten (wie Oracle Exalytics). Solche Lösungen sind heute schon stabiler und ausgereifter als SQL-Engines innerhalb von Hadoop. Die meisten der fortgeschrittenen Einsätze in den Unternehmen nutzen die hier dargestellten hybriden Lösungen und nehmen das Bewegen von Key-Daten beziehungsweise aggregierte Daten in Kauf.

Phase 3: Single Logical System

Die sinnvollste Lösung zur Eliminierung aller Nachteile der jeweiligen Plattformen ist ein integrierter Datenspeicher, der die spezifischen Vorteile von Hadoop auf der einen und eines relationalen Datenbanksystems auf der anderen Seite für Analysezwecke vereint. Nahezu keine aktuelle Implementierung erreicht gegenwärtig diese Phase. Die meisten Entwicklungen fokussieren sich auf eine der beiden Tech-

nologien, anstatt an einem einheitlichen logischen System zu arbeiten.

Das in der dritten Phase beschriebene, logisch integrierte System ist Voraussetzung für neue Möglichkeiten einer breiteren Anwenderschicht. Damit einher geht die Standardisierung der Zugriffssprache auf diesen neuen Datenspeichern. Es muss eine deklarative Sprache sein, die Zugriff auf beide Welten (Hadoop und RDBMS) erlaubt. Wie bereits ausgeführt, entwickelt sich SQL zu der bevorzugten Abfragesprache in der Hadoop-Welt, so wie sie es im Data Warehouse bereits seit vielen Jahren ist. Obwohl es momentan noch andere Abfragesprachen für Hadoop gibt, wird SQL die Abfragesprache der Wahl für integrierte Systeme werden.

Beim Einsatz von SQL haben alle bestehenden Werkzeuge (BI und sonstige Anwendungen) uneingeschränkten Zugang auch zu Hadoop-Daten, ohne dass man an dem bestehenden Ökosystem etwas ändern muss. Ein solches System muss jedoch Zugriffe quer über das Data Warehouse und Hadoop hinweg möglichst optimiert erlauben, ohne die Nachteile, die man heute bei Hive findet. Hadoop muss Hadoop bleiben, mit der Möglichkeit, schnell und massenweise Daten zu importieren und diese Daten flexibel und einfach auswerten zu können, bei gleichzeitiger Skalierbarkeit.

Abbildung 3 zeigt den Bauplan für eine solche Lösung. Sie umfasst ein Hadoop- und ein RDBMS-System für die jeweiligen beschriebenen spezifischen Anforderungen. Zwischen beiden Bestandteilen sollte ein High-Speed-Netzwerk liegen. Beispiele dafür sind InfiniBand oder die neueste Ethernet-Technologie (> 10GigE).

Die Zugriffe mittels SQL müssen über einheitliche Metadaten erfolgen, die sich über beide Systeme erstrecken. Hive und HCatalog stellen die Basis für den Hadoop-Anteil des Metasystems dar und sind mit dem Datenbank-Katalog verbunden, sodass ein einheitliches SQL-System alle Objekte sowohl im Hadoop als auch in der Data-Warehouse-Datenbank zusammenhängend abfragen kann.

Die Benutzer können ohne Änderung ihrer Umgebung oder ihrer Abfragesprache in beide Welten hinein abfragen und in einer einzigen Abfrage Daten aus beiden Welten zusammen in ihre Analyse miteinbeziehen. Die Performance der Abfragen ist gut, weil die Abarbeitung direkt auf Prozesse in der jeweiligen Schicht übertragen und dort erledigt wird. Es findet keine aufwändige Datenbewegung statt, außer in den Systemen selbst. Nur die Ergebnisse fließen an die Benutzer zurück.

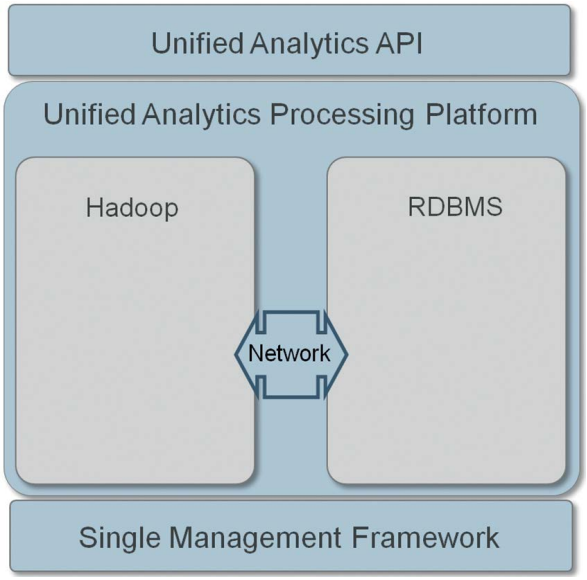

Abbildung 3: Vereinheitliche Zugriffsplattform Hadoop/RDBMS

Auch das föderative Konzept lässt sich sinnvoll umsetzen. BI-Server müssen jetzt nicht mehr alle Daten in ihren Cache laden, sondern können sich auf die wesentlichen Daten fokussieren. Die Hadoop-spezifische Verarbeitung erfolgt im Hintergrund durch die jeweiligen Werkzeuge innerhalb von Hadoop.

Wie gesagt, bewegen sich die technischen Entwicklungen im Augenblick auf dem Level der zweiten Phase. Hadoop, oft verstanden als Erweiterung des Data Warehouse, leistet bereits an einigen Stellen seinen Wertbeitrag in den Unternehmen. Anwender sollten sich mit den eingestellten Erfolgen jedoch nicht zufriedengeben, sondern ein Ziel-Szenario anstreben, in dem SQL als einfaches und standardisiertes Zugriffsmedium zu allen relevanten Daten vorhanden ist.

Fazit

Hadoop spielt in den kommenden Jahren eine immer bedeutsamere Rolle in Data Warehouse-Architekturen und SQL wird sowohl innerhalb von Hadoop als auch als eine übergreifende Abfragesprache wichtiger denn je. Heutige Lösungen ziehen Daten entweder hoch in den BI-Layer oder sie kopieren Daten beziehungsweise Aggregate in eine Data-Warehouse-Umgebung. SQL-Lösungen innerhalb von Hadoop leiden heute noch an Performance-Limitierungen. Eine sinnvolle Lösung sind SQL-gestützte Brückenlösungen über beide Datenhaltungen hinweg. Diese Lösungen werden in den kommenden Jahren zum Hauptstandbein, wenn Unternehmen über alle ihre Daten hinweg Nutzen ziehen wollen.

Jean-Pierre Dijcks

Übersetzt und angepasst von
Alfred Schlaucher
alfred.schlaucher@oracle.com