Aufbau eines Semantic Layers zwischen DB und Hadoop

Matthias Fuchs
Capgemini Deutschland
Nürnberg

Schlüsselworte
Hadoop, BigDataSQL, Connectoren, Hive, Cloudera, Impala, Exadata

Einleitung
Hadoop mit all seinen verschiedenen Ausprägungen, bietet für bestimmte Aufgaben in der Datenhaltung oft eine skalierbare und flexible Lösung, zu einem günstigen Preis. Somit ist es möglich, dass Daten eines Data Warehouse verteilt auf mehreren Systemen vorliegen. Ein Zugriff auf alle Daten, ohne die Daten duplizieren zu müssen, ist wünschenswert.
Wie ist der Aufbau eines Semantic Layers für Daten aus einer RDBMS und z.B. Hive oder NOSQL möglich? Welche Möglichkeiten bietet Oracle? Was ist Big Data SQL? Welche weiteren Vorteile gibt es durch die Verwendung aus der Kombination einer Oracle Datenbank mit einem Hadoop Cluster?

Der Vortrag geht auf die Möglichkeiten der verteilten Datenhaltung zwischen Hadoop und DB ein. Möglichkeiten des Aufbaus der Zugriffsschicht, Anhaltspunkte für die Abfragegeschwindigkeit und Securityaspekte. Ebenso wird auf die zusätzlichen Features einer Exadata – Big Data Appliance Konfiguration eingegangen.

Business Intelligence Semantic Layer
Unter einem Semantic Layer versteht man eine Representationsebene von Daten, die es Endbenutzern ermöglicht Daten ohne Technisches Wissen abzufragen. Der Begriff geht auf Busniess Objects zurück. Der Layer soll es ermöglichen, dass komplizierte Bezeichnungen und Beziehungen auf Objekte wie Produkte, Kunden oder Umsatz zurückgeführt werden können. Ein Semantic Layer ist in den meisten BI Tools von SAP, IBM, SAS oder Oracle ein zentraler Bestandteil.
Im Oracle Business Intelligence Enterprise Edition (OBIEE) findet ein Mapping der Physikalischen Tabellen auf das Business Modell statt. Das Mapping wird im Repository abgelegt und kann von OBIEE Dashboards abgefragt. Eine Verwendung des so erstellten Layers findet im Rahmen von OBIEE statt, eine Migration auf Tools von anderen Herstellern ist meist nicht möglich.

.

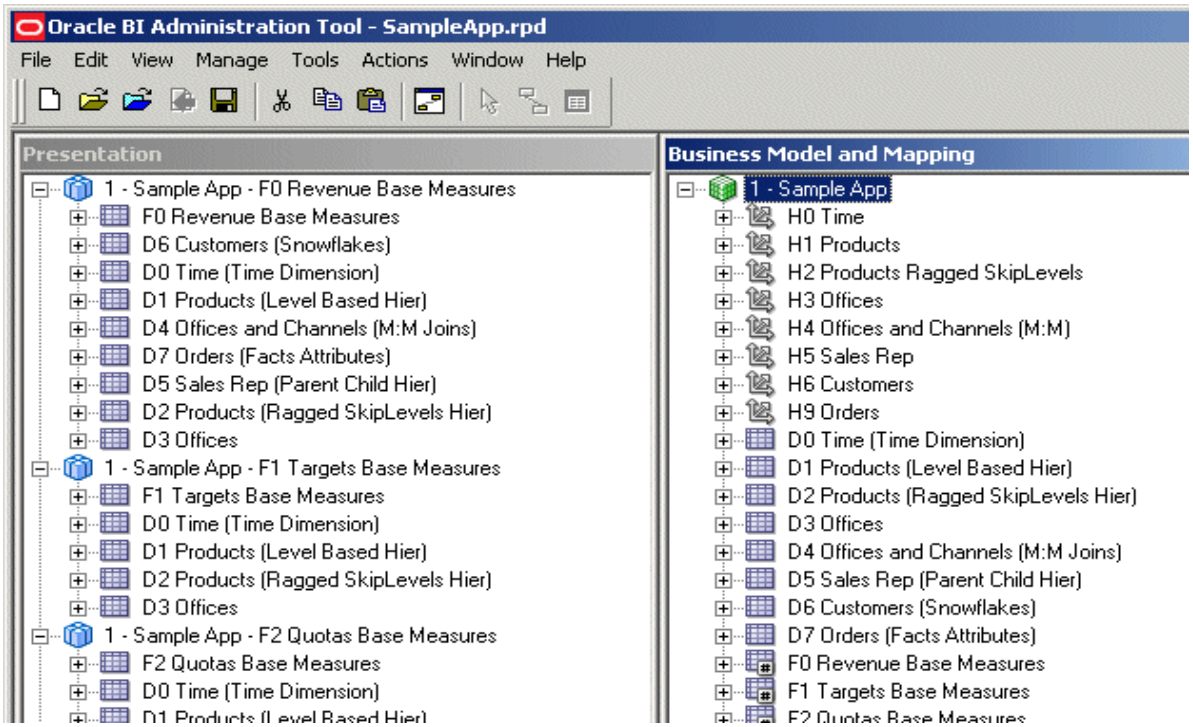

Abb. 1: OBIEE Business Model und Mapping

Andere BI Tools wie SAP Business Objects, IBM Cognos oder MicroStartegy verfahren ähnlich.

Vor- und Nachteiles des Semantic Layers
Um Business User einen einfachen Umgang mit den Daten zu ermöglichen ist ein Semantic Layer zwingend notwendig. Ohne diesen Layer sind Ad Hoc Queries nur machbar, wenn die Struktur aus Tabellen und Views in der Datenbank dem Endbenutzer im Detail bekannt ist. Ebenso sollte der Layer mehr sein, als nur ein direktes Mapping auf die Tabellen und Views in der Datenbank. Andererseits ist ein erstellter Layer zwischen verschiedenen Bi Tools nicht migrierbar oder wiederverwendbar. Das heißt bei Verwendung von mehreren Tools verschiedener Hersteller, muss jedes Mal ein Semantic Layer neu angelegt werden. Änderungen müssen parallel in allen Werkzeugen nachgezogen werden.

Semantic Layer und Big Data
Mit Big Data werden immer mehr und verschiedenere Datenquellen in den Semantic Layer aufgenommen. Es entsteht eine Fülle von neuen Datenquellen. Zusätzlich zu relationalen Daten, sollen auch JSON Daten oder NoSQL Daten einbezogen werden. Der Aufbau des Layers wird schwieriger und aufwendiger. Um die Daten einfach analysieren zu können, haben sich Ansätze ergeben wie Semantic Data Preparation. Dabei wird meist versucht, mit einfach visuellen Tools die Zusammenhänge in den Daten zu finden. Bei Oracle fällt in diese Reihe Oracle Big Data Discovery. Leider ist das aber noch keine Integration in das Datawarehouse. Es werden keine Strukturen aufgebaut, um die Kennzahlen daraus zu generieren. Es ist nur eine Vorverarbeitung, die als Basis für eine Verwendung der Daten im Datawarehouse dient.

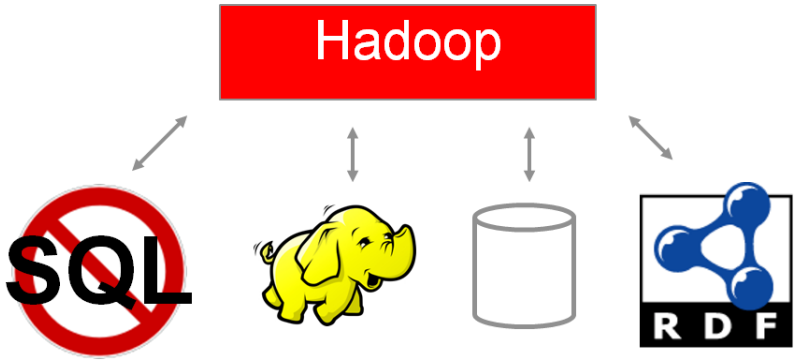

Abb. 2: Vielfalt unter Hadoop

Die Datenvielfalt
In Hadoop können Daten in vielfältiger Weise abgelegt werden. Neben der Ablage in einer relationalen Datenbank, wie etwa Hive können Daten ebenso in Form von beliebigen Files oder als XML/REST Struktur abgelegt werden. Eine direkte Verarbeitung über HDFS auf File Ebene oder über HBase als Key-Value Datenbank wird meist aus Sicht des BI Tools nicht ermöglicht oder ist aufwendig zu implementieren.

HCatalog – Metadatenlayer in Hadoop
Die Daten in Hadoop sollen möglichst über eine einheitliche Schnittstelle benutzbar werden. Als Basis dient hier Hive. Hive unterstützt die meisten Datentypen, die auch in einer relationalen Datenbank, wie etwa Oracle, vorhanden sind. Dazu kommt als Ergänzung die Funktion der SerDes. SerDes – Serializer – Deserializer ermöglichen es, beliebige Objekte aus dem Hadoop Filesystem oder anderen Applikationen zu lesen und in Hive zu verarbeiten. Anschließend können diese wieder geschrieben (desierialized) oder an andere Programme weitergegeben werden. SerDes sind für viele Applikation bzw. Dateitypen vorhanden (HBase, Json, Avro etc) können aber auch selber in Java entwickelt werden.
Die Konfiguration, wie man auf die Daten zugreift, legt Hive im HCatalog ab. Dadurch entsteht eine Art semantischer Layer, der beschreibt wo und in welcher Form die Daten abgelegt wurden. Der HCatalog kann über eine Rest Schnittstelle abgefragt werden. Somit wird der Katalog nicht nur von Hive genutzt, sondern auch von z.B. Cloudera Impala, Oracle BigDataSQL oder Pivotal HAWQ. Die Tools lesen die Informationen und können darauf Tabellen und Views aufbauen. Der Zugriff auf die Daten findet dann nicht über Hive statt, sondern direkt zwischen der Applikation und HDFS.

Performanceunterschiede – Prozesslayer
Die Performance ist gerade bei Abfragen oder Dashboards entscheidend. Das Hadoop Filesystem (HDFS) ist optimiert für lesen und schreiben. Es kann große Datenmengen schnell, parallel verarbeiten. Als Processschicht diente ursprünglich MapReduce. MapReduce kann sehr große Datenmengen in Batchabläufen verarbeiten. Die Anforderungen an die Performance stiegen und somit wurde auch die Processschicht weiterentwickelt. Es entstanden z.B. Tez und Spark als Verarbeitungslayer. Ebenso entstanden Applikation mit eigener Prozessschicht wie Cloudera Impala oder Oracle BigDataSQL.

Einbindung in den Semantic Layer
Für die Einbindung in den Semantic Layer ergeben sich somit vielfältige Möglichkeiten. Die Verwendung des HCatalogs erscheint sehr hilfreich. Eine Kombination mit einer performanten Process Schicht erscheint ebenso hilfreich, um die Abfragen im Report oder Dashboard im Bereich von Sekunden zu halten. Der Connect zu Hive wird von fast allen Tools unterstützt, die

Abfragesprache HiveQL ist dem SQL sehr ähnlich. Hive wiederum unterstützt den Connect auf vielfache Daten in Hadoop. Ein Zugriff auf Big Data ist somit gewährleistet. Natürlich ist oft eine Einbindung von Query Optimierten SQL Datenbank auf Hadoop wie Cloudera Impala oder Pivotal HAWQ möglich. Dort ist die Query Performance deutlich höher als bei Hive und es sind ACID (Atomicity, Consistency, Isolation, Durability) konforme Datenbanken. Diese Funktionalität wird aber von Oracle abgedeckt und ein Einsatz zweier gleichartiger Datenbanktypen ist nicht wünschenswert. Stattdessen sucht man Möglichkeiten Hadoop Daten möglichst einfach und performant in bestehende Systeme einzubinden. Daraus ergeben sich aus Oracle Sicht zwei Möglichkeiten Hadoop einzubinden:

- Einbindung über die Datenbank mittels BigDataSQL bzw. Connectoren
- Direkte Einbindung über HiveQL in die BI Schicht

Details dazu werden im Vortrag aufgezeigt.

BigDataSQL weitere Performanceoptimierung
Diese Einbindung wird im Rahmen des Vortrages präsentiert und der prinzipielle Ablauf geschildert.

Security
Werden die Daten über BigDataSQL oder Connectoren im Rahmen der Datenbank abgerufen, greifen die gleichen Security Mechanismen, wie sie bereits in der Datenbank vorhanden sind. Es sind keine neuen Regeln zu implementieren. Es kann mit dem gewohnten Standard weiter gearbeitet werden.

Zusammenfassung
Oracle bietet mit seinen vielfältigen Integrationen zwischen Datenbank und Hadoop ein Framework welches einfach zu integrieren ist und eine höchste Performance bietet. Einer Verwendung von Hadoop Daten im BI Layer steht nichts im Wege. Mit Hadoop ergeben sich Möglichkeiten auf Daten zuzugreifen, deren Aufbewahrung im Datawarehouse aufgrund der Größe nicht möglich ist.

Kontaktadresse:
Matthias Fuchs
Capgemini Deutschland
Bahnhofstr. 11c
90402 Nürnberg

Telefon:	+49 911 30096 176
Fax:	+49 151 4025 1964
E-Mail	matthias.fuchs@capgemini.com
Internet:	www.de.capgemini.com