ODI und Big Data
Möglichkeiten und ein Erfahrungsbericht

Dr. Holger Dresing
Oracle Deutschland B.V. & Co. KG
Hannover

Schlüsselworte
Oracle Data Integrator – ODI, Big Data, Hadoop, MapReduce, Hive, HDFS, PIG, Spark, Big Data SQL, Oracle Loader für Hadoop (OLH); Oracle SQL Connector for HDFS (OSCH), Knowledge Module

Einleitung
„Big Data“ ist zur Zeit ein Hype-Begriff, der auch von Oracle mit entsprechenden Produkten und Lösungen unterstützt wird. Dabei geht es um große Datenmengen, die gespeichert und analysiert werden müssen. Das Thema Datenintegration spielt auch hier eine wesentliche Rolle. Der Oracle Data Integrator ist das strategische Werkzeug zur Steuerung aller Datenintegrationslösungen. Der ODI unterstützt auf Basis des Hadoop Ecosystems verschiedenste Technologien und Oracle Produkte. Wichtig ist, daß mit dem ODI die Integrationsprozesse wie bekannt erhalten bleiben und über die Codetemplates, die beim ODI Knowledge Module genannt werden, die Integration gesteuert wird. Gleichwohl werden Technolgien wie HDFS, Hive, PIG, Sparc oder Oozie unterstützt. Zudem gibt es Features, die aus dem Oracle Datenbank-Umfeld kommen wie Big Data SQL oder Oracle Loader for Hadoop, die in den ODI integriert wurden. Aus Entwicklersicht bleibt die bekannte Vorgehensweise mit Mapping und Workflows erhalten, lediglich der generierte Code paßt sich den neuen Technologien von Big Data an.

Oracle Data Integration
Einsatzgebiete des ODI sind ETL, Migration von Applikationen, Datensynchronisationen oder auch Data Quality Szenarien. Aus der Sicht eines ETL-Werkzeugs bietet er Mappings, um Datentransformationen auf Tabellenebene zu modellieren, Packages, um Workflow zu definieren, eine Protokollsicht auf die ausgeführten Mappings und Packages und weitere Funktionen, z. B. das ODI SDK für individuelle Erweiterungen. Die Codegenerierung basiert auf Codetemplates, die bei ODI Knowledge Module genannt werden. Mit diesen Modulen lassen sich die Funktionalität und die unterstützten Quell- und Zielsysteme erweitern.

Big Data bei Oracle
Big Data wird an den 4 „V“s festgemacht: Volume, Velocity, Variety and Value.
Bei Volume geht es um große Datenmengen, die nur wenig verdichtet sind wie z. B. WebSeiten oder Social Media. Hier geht es um die Verdichtung von Daten, im Unternehmen geht man jedoch von sehr großen Datenmengen aus.
Bei Velocity geht es um die Umschlagshäufigkeit der Daten. Die höchste Umschlagshäufigkeit schreibt die Daten direkt in den Hauptspeicher, niedrigere Umschlagshäufigkeit schreibt sie auf die Platte.
Unter Variety werden unstrukturierte Datentypen wie Texte, Audio- oder Videodaten verstanden. Wichtig ist, dass diese Daten den gleichen Anforderungen wie strukturierte Daten unterliegen: Aggregierung, Lineage oder Sicherheit. Die Struktur kann sich abrupt ändern, die Daten können im Batch und Realtime angeliefert werden und müssen den Anforderungen an Transaktionen oder an Auswertungssystemen entsprechen.

Bei Value geht es darum, den Nutzen und die Inhalte der Daten zu erfassen. Der Nutzen ist bei Big Data Technologien sehr viel schwieriger zu erfassen.
Gegenüber dem relationalen Ansatz mit seinen Schemata ist der Big Data-Ansatz sehr viel komplexer, da sich Strukturen laufend ändern können. In stukturierten Systemen ist die Syntax und deren Semantik festgelegt, bei Big Data können sich Syntax und Semantik fortlaufend ändern.
Das Thema läßt sich ausweiten auf die Analyse solcher Strukturen und die Architektur dieser Strukturen. An dieser Stelle ist jedoch die Integration mit ODI gefragt und daher soll auch nur dieser Bereich weiter dargestellt werden. Im relationalen „schema-on-write"-Ansatz werden die Daten aus den Source-Systemen mit einer ETL Lösung in ein Schema gebracht und stehen anschließend für Auswertungen zur Verfügung. Im Big Data „schema-on-read"-Ansatz werden die Daten aus den Quellen in ein sogenanntes Raw Data Reservoir gebracht. Ein Data Reservoir ist eine große Datensammlung. Dann wird dafür Code generiert, die die Daten bearbeitet und daraus verschiedene Schemata ableitet. Der bekannteste Codegenerator ist Hadoop und der Map Reduce Algorithmus. Details dazu hat Oracle in der angegebenen Literatur beschrieben.

Big Data und Hadoop
Hadoop ist ein Java-basiertes Framework der Apache Software Foundation, das auf dem MapReduce-Ansatz von Google basiert. Es hat verschiedene Bestandteile und Erweiterungen, die auch von ODI unterstützt werden. Als Beispiele seien genannt:
Hadoop Distributed File System (HDFS) eignet sich zur Speicherung großer Datenmengen auf den Dateisystemen mehrerer Rechner (Knoten). Die Dateien werden in Datenblöcke mit fester Länge zerlegt und redundant auf alle Knoten verteilt.
MapReduce ist ein von Google entwickelter Algorithmus, um große Datenmengen über verteilte Rechnerknoten verarbeiten zu können.
HBase ist eine Datenbank zur Verwaltung großer Datenmengen innerhalb eines Hadoop Clusters. Diese ist für Daten geeignet, die sich selten ändern, dafür aber sehr häufig ergänzt werden.
Hive erweitert Hadoop um Data Warehouse-Funktionen, insbesondere die Abfragesprache HiveQL. Das ist eine auf SQL basierende Abfragesprache und ermöglicht die Verwendung einer SQL-ähnlichen Syntax.
PIG ist eine auf MapReduce basierende Sprache, um Programme in der Pig Latin zu erstellen.
Scoop ist ein Apache Projekt, um Daten von einer relationalen Datenbank nach Hadoop zu transferieren.
Oracle Loader for Hadoop (OLH) ist ein MapReduce Werkzeug, um das Laden von Daten aus Hadoop in eine Oracle Datenbank zu optimieren.
SQL Connector for HDFS (OSCH) ermöglicht es, von einer Oracle Datenbank direkt auf HDFS zuzugreifen.

Integration von Big Data und ODI
Prozesse erstellen und Codegenierung in ODI
In ODI wird zwischen einem logischen und einem physischen Design unterschieden. Beim logischen Design setzt der Entwickler die Business Logik um, im physischen Design wird mit Hilfe der Knowledge Module die Codegenerierung festgelegt. Diese Vorgehensweise ist immer die gleiche, auch im Big Data-Umfeld. Der Entwickler sieht auf der logischen Ebene ausschließlich Schemata und deren „quasi" relationale Strukturen. Auch im Big Data- Umfeld ändert sich nichts.

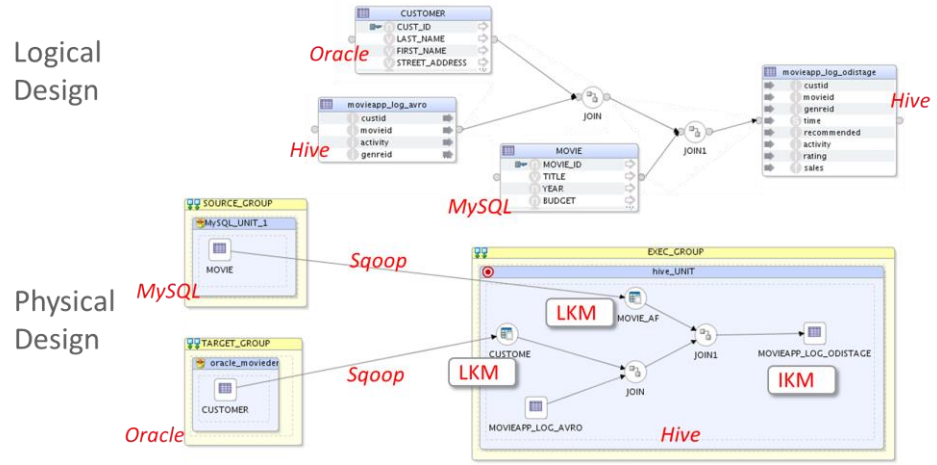

Abb. 1: Logisches und physisches Design in ODI

Passend dazu gibt es Knowledge Module auch für die Big Data-Technologien. Beispielhaft seien die folgenden genannt:

- IKM File To Hive (Load Data)
- IKM Hive Control Append
- IKM Hive Transform
- IKM File-Hive to Oracle (OLH)
- CKM Hive
- RKM Hive
- LKM HBase to Hive
- IKM Hive to HBaseRKM Hbase
- IKM SQL to Hive- HBase-File (SQOOP)
- IKM File-Hive to SQL (SQOOP)

Die Vorteile von ODI bleiben erhalten: Es wird nativer Code für verschiedenste Datenquellen generiert, auch unter Performance-Aspekten. Eine Transformation, die heute auf einem relationalen System läuft, kann in der Zukunft genauso auf einem Hadoop Cluster oder in der Zukuntft in Hadoop-Sprachen generiert werden, sobald die entsprechenden Umgebungen verfügbar sind. Es wird keine zusätzliche Middleware benötigt, wodurch die Kosten gesenkt werden. Der deklarative Ansatz bleibt erhalten. Dadurch vereinfacht sich die Entwicklung und ist wiederverwendbar mit unterschiedlichen Technologien.

Demonstration

In der virtuellen Maschine namens „Big Data Lite“ hat Oracle dafür ein umfangreiches Beispiel aufgebaut, mit denen veranschaulicht wird, wie die verschiedenen Big Data-Technologien zusammenspielen. Unter dem Link zur „Big Data Lite“ sind die Details zu finden. An dieser Stelle wird nur ein kurzer Überblick gegeben. Die Demos basieren auf der in Abb. 2 dargestellten Umgebung. Abhängig von der Version der „Big Data Lite“ kann die Quelle auch eine MySQL-Datenbank und die Flat Files Json Files sein.

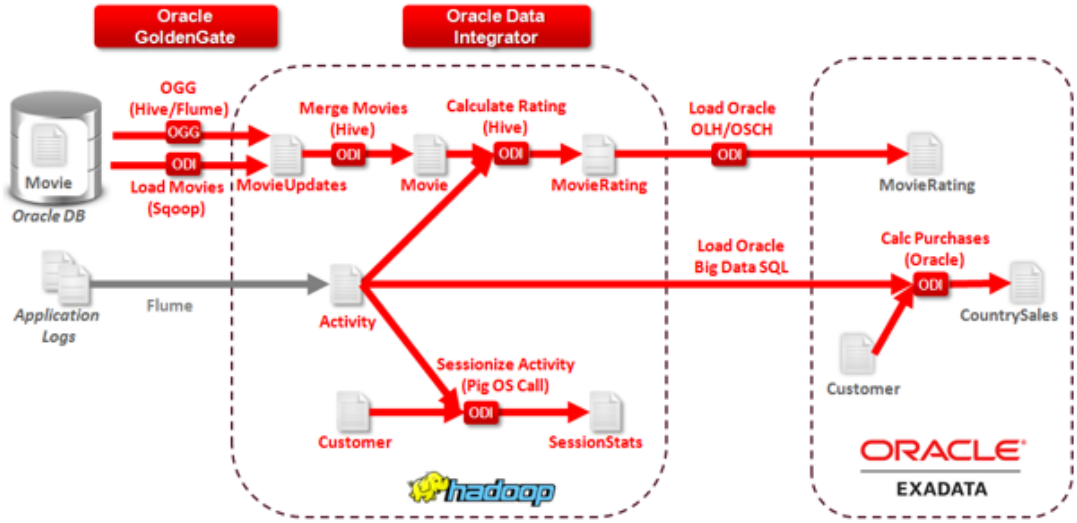

Abb. 2: Beispiele in Big Data Lite

Demonstration 1:	Laden von Daten via Scoop
Quelle:	Oracle DB
Ziel	Hive
Mapping Name in ODI	A - Load Movies (Scoop)
Knowledge Modul	IKM SQL to HBase-Hive-File (Scoop)

An dieser Stelle gibt es auch eine Demo zu Oracle Golden Gate genannt. Dazu gibt es einen weiteren Vortrag.

Demonstration 2:	Laden von Daten mit Hive
Quelle:	Hive
Ziel	Hive
Mapping Name in ODI	B - Merge Movies (Hive)
Knowledge Modul	IKM HBase Control Append

Bei dieser Demo werden Daten mit Aggregations- und Join-Operatoren von einem Hive-Objekt in ein anderes Hive-Objekt geladen.

Demonstration 3:	Berechnungen in Hive
Quelle:	Hive
Ziel	Hive
Mapping Name in ODI	C – Calc Ratings (Hive-Pig-Spark)
Knowledge Modul	LKM File to Spark/IKM HBase Control Append

Bei dieser Demo werden Daten mit Aggregations-, Filter- und Join-Operatoren von einem Hive-Objekt in ein anderes Hive-Objekt geladen.

Demonstration 4:	Berechnungen mit Json
Quelle:	File/HDFS
Ziel	Hive
Mapping Name in ODI	D – Calc Ratings (Json)
Knowledge Modul	LKM File to Spark/IKM HBase Control Append

Bei dieser Demo werden Daten mit Aggregations-, Filter- und Join-Operatoren von einem HDFS-Objekt in ein Hive-Objekt geladen.

Demonstration 5:	Oracle Loader für Hadoop (OLH)
Quelle:	File/HDFS
Ziel	Oracle
Mapping Name in ODI	E – Load Oracle (OLH)
Knowledge Modul	LKM Hive to Oracle OLH-OSCH Direct

Bei dieser Demo werden Daten mit Aggregations-, Filter- und Join-Operatoren von einem HDFS-Objekt in ein Hive-Objekt geladen. ODI stellt für OLH und OSCH ein spezielles Knowledge Modul zur Verfügung.

Demonstration 6:	Big Data SQL
Quelle:	Oracle/Hive
Ziel	Oracle
Mapping Name in ODI	F – Calc Sales (Big Data SQL)
Knowledge Modul	LKM Hive to Oracle (Big Data SQL)

Bei dieser Demo werden Daten mit Aggregations-, Filter- und Join-Operatoren von einem HDFS-Objekt in ein Hive-Objekt geladen.

Demonstration 7:	Arbeiten mit Pig
Quelle:	Hive
Ziel	Hive
Mapping Name in ODI	G – Sessionize Data (Pig)
Knowledge Modul	LKM Hive to Pig/LKM Pig to Hive

Bei dieser Demo werden Daten mit Aggregations-, Sort- und Join-Operatoren von einem HDFS-Objekt in ein Hive-Objekt geladen.

Ausblick

Der ODI-Entwickler arbeitet mit Hadoop-Technolgien wie mit jeder anderen Datenquelle (in ODI Datastore genannt). Er muß lediglich die Knowledge Module ersetzen und die entsprechende Toplogy (in ODI Schema genannt) konfigurieren. Schon seit ODI 11g wird das Hadoop Framework unterstützt. Die Demonstration basiert auf ODI 12.1.3.0.1. Neu in diesem Patchset sind SPARC, Pig und Oozie. Für diese drei Technolgien wird die Oracle Data Integrator Enterprise Edition Advanced Big Data Option benötigt.

Weiterführende Hinweise

Oracle: An Enterprise Architect's Guide to Big Data, Reference Architecture Overview, Oracle Enterprise Architecture White Paper, May 2015, Link: http://www.oracle.com/technetwork/topics/entarch/articles/oea-big-data-guide-1522052.pdf vom 15.09.2015

Oracle: Information Management and Big Data – A Reference Architecture, September, 2014, Link: http://www.oracle.com/technetwork/database/bigdata-appliance/overview/bigdatarefarchitecture-2297765.pdf vom 16.09.2015

Oracle: Oracle Big Data Lite Virtual Machine, http://www.oracle.com/technetwork/database/bigdata-appliance/oracle-bigdatalite-2104726.html vom 16.09.2015

Kontaktadresse:

Dr. Holger Dresing
Oracle Deutschland B.V. & Co. KG
Thurnithistraße 2
D-30519 Hannover

Telefon: +49 (0) 95787-118
Fax: +49 (0) 95787-118
E-Mail holger.dresing@oracle.com
Internet: www.oracle.de