

Datenaustausch Hadoop & Oracle DB
Carsten Herbe
metafinanz Informationssysteme GmbH
München

Schlüsselworte

Hadoop, Sqoop, Sqoop 2, Hive, Oracle Big Data Konnektoren

Einleitung

Neben der klassischen Data Warehouse Welt entsteht im Kontext von Big Data mit Hadoop als zentrale Datenplattform eine technologisch (und auch methodisch) völlig neue Welt. Diese beiden Welten sind jedoch nicht isoliert, ihr volles Potential kann sich nur entfalten, wenn Daten zwischen ihnen ausgetauscht werden können. Z.B. möchte man vorhandene Stammdaten nach Hadoop laden um sie mit den dortigen Daten zu verknüpfen. Oder aggregierte Analyseergebnisse aus Hadoop werden in das Oracle Data Warehouse geladen um die Ergebnisse den Anwendern der klassischen BI Tools zur Verfügung zu stellen.

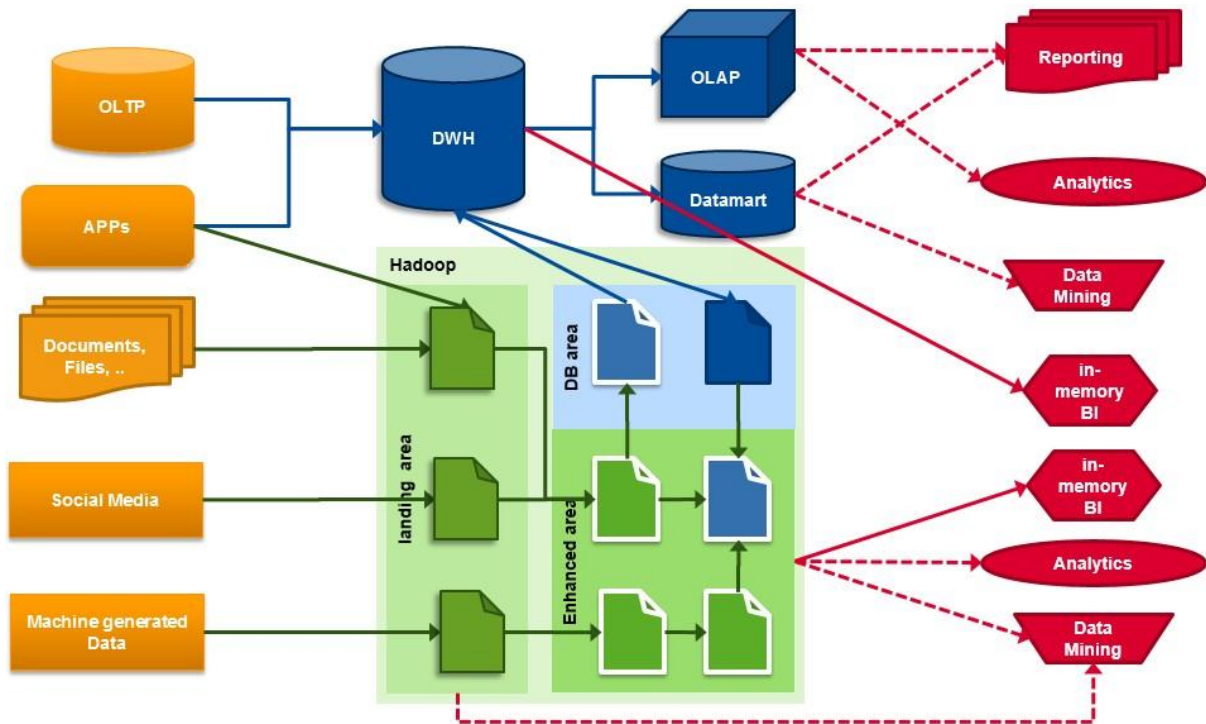


Abb. 1: Beispiel Big Data Architektur

In diesem Dokument wird beschrieben, wie man Daten zwischen Hadoop – bzw. dem Hadoop Distributed File System HDFS - und Oracle Datenbanken austauscht. Dazu werden zwei Lösungen vorgestellt: das Open Source Werkzeug Sqoop und die kommerziellen Big Data Konnektoren von Oracle.

Exkurs Hive

Für den Datenaustausch zwischen HDFS und relationalen Datenbanken spielt Hive eine besondere Rolle, daher wird im Folgenden kurz erklärt, was Hive ist.

Hadoop als Datenspeicher für unstrukturierte Daten auf der einen Seite und Oracle als relationale Datenbank für sehr strukturierte Daten auf der anderen Seite, wie passt das zusammen? Wenn man Daten aus der Oracle Datenbank nach HDFS lädt, behalten sie natürlich ihre Tabellenstruktur. Und umgekehrt lassen sich nur Daten aus HDFS nach Oracle laden, die auch in eine Datenbanktabelle hineinpassen.

Mit Hive hat man die Möglichkeit, „Tabellen-Metadaten“ für Daten in HDFS zu definieren. Diese werden weiterhin wie gewohnt in HDFS als z.B. CSV-Dateien gespeichert.

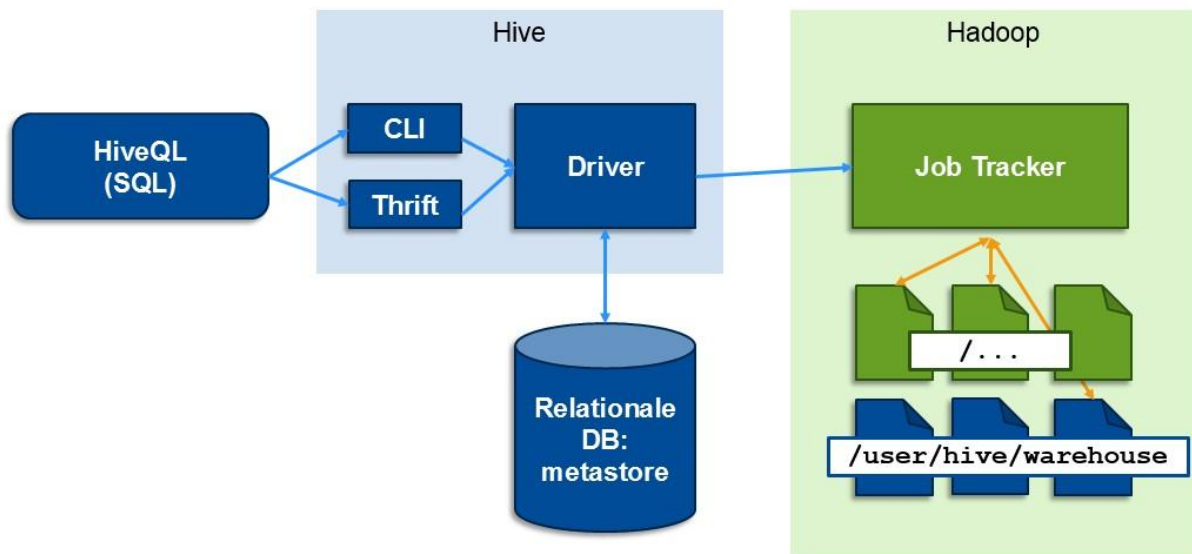


Abb. 2: Hive Architektur

Diese Metadaten werden im Hive Metastore abgelegt. Dort stehen sie anderen Hive-Nutzern zur Verfügung. Und über HCatalog, welches seit März 2013 zu Hive gehört, können diese Metadaten auch von anderen Tools wie z.B. Pig genutzt werden.

Zusätzlich zu den Metadaten bietet Hive mit der SQL-ähnlichen Abfragesprache HiveQL die Möglichkeit, Daten mit SQL in HDFS zu analysieren (SELECT) oder sogar neue Daten in HDFS bzw. Hive zu erzeugen (INSERT). Natürlich gelten hier die gleichen Einschränkungen wie für HDFS: Es gibt kein Update und keine Einzelsatzverarbeitung).

Werden also Daten mit einer Oracle Datenbank ausgetauscht, bietet es sich an die Hive Metadaten zu nutzen. Dies ist aber keine Voraussetzung.

Sqoop

Sqoop ist ein Open Source Tool zum Austausch von Daten in HDFS (bzw. Hive) und verschiedenen Datenbanken, darunter natürlich auch Oracle. Dazu bietet Sqoop verschiedene Konnektoren, welche per JDBC auf das Datenbanksystem zugreifen.

Auch Sqoop hat einen Metastore, in dem bereits erstellte Sqoop Jobs (d.h. definierte Datentransfers: Quelle, Ziel und welche Daten zu transferieren sind) gespeichert werden. Dieser kann von verschiedenen Nutzern geteilt werden, um bereits definierte Jobs mehrfach auszuführen. In erster Linie ist Sqoop aber eine Client Anwendung, bei der auch die JDBC Treiber lokal installiert werden müssen. Typischerweise wird Sqoop auf einem Hadoop Cluster Knoten installiert.

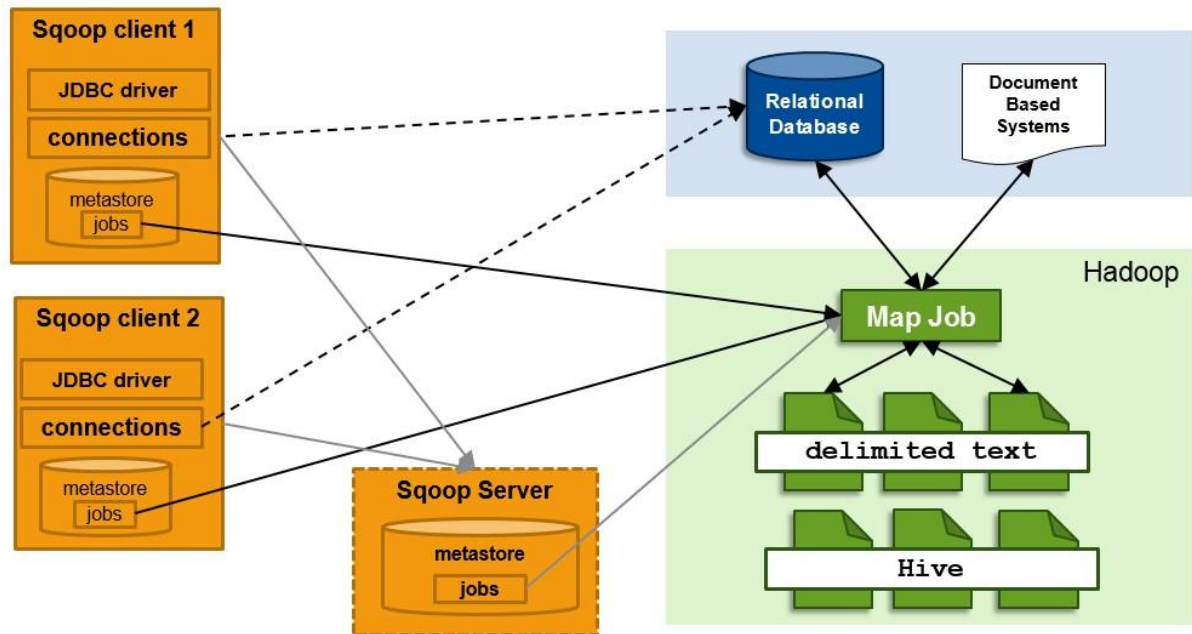


Abb. 3: Sqoop Architektur

Beispiel Import von Oracle nach HDFS:

```
sqoop import
  --connect jdbc:oracle:thin:@//dbserver:1521/orcl
  --username scott
  --table SCOTT.EMP
  --columns "employee_id,first_name,last_name,job_title"
  --where 'rownum=1'
  --target-dir /user/mfhadoop
  -P
  [--hive-import]
```

Der Parameter `-P` gibt an, dass das Passwort von der Konsole gelesen wird. Mit dem optionalen `--hive-import` werden die entsprechenden Hive-Metadaten erzeugt.

Beispiel Export von HDFS nach Oracle:

```
sqoop export
--connect jdbc:oracle:thin:@//dbserver:1521/orcl
--table SCOTT.EMP
--export-dir /hdfs/transfer
--username scott
--fields-terminated-by '\t'
--lines-terminated-by '\n'
```

Sqoop 2

Die Architektur von Sqoop beinhaltet einige Nachteile. Abgesehen davon, dass JDBC Treiber lokal zu installieren sind, müssen auch die Verbindungsparameter zur Datenbank inklusive Passwort dem Nutzer bekannt sein.

Mit Sqoop 2 wird eine Client-Server-Architektur eingeführt. Die JDBC Treiber werden auf dem Sqoop 2 Server installiert. Dort werden auch die Datenbankverbindungen (Connections) gespeichert. So können diese einmalig von einem Administrator angelegt werden und anschließend von verschiedenen Nutzern verwendet werden, ohne dass diese die Passwörter kennen müssen.

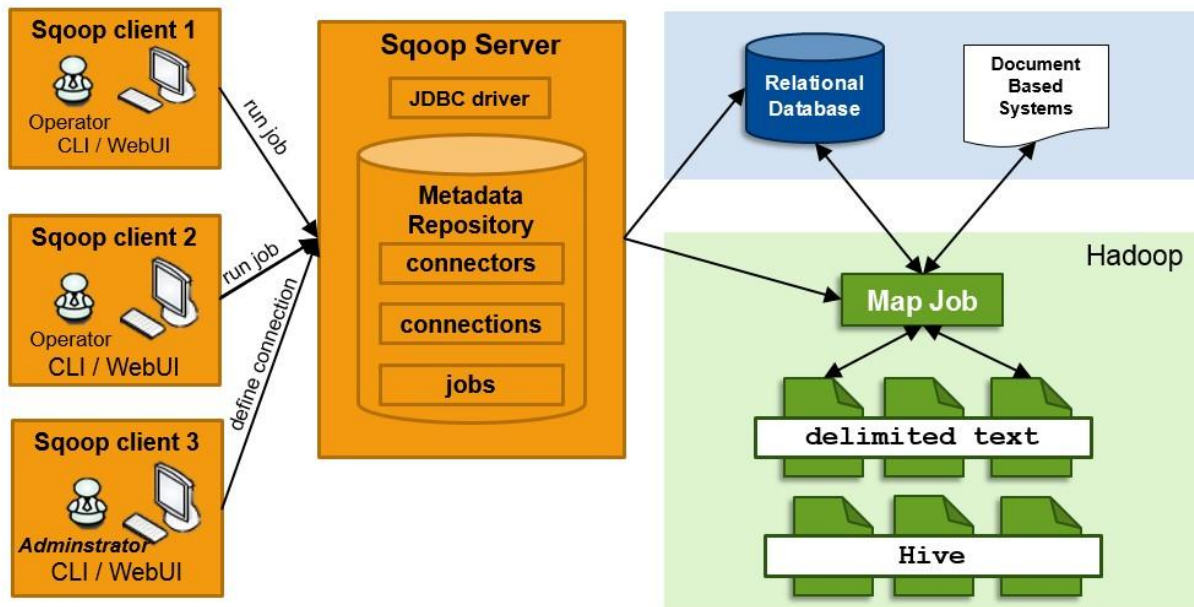


Abb. 4: Sqoop 2 Architektur

Oracle Big Data Konnektoren

Speziell für das Laden von Daten aus Hadoop in eine Oracle Datenbank bietet Oracle mit den Big Data Konnektoren ein kommerzielles Produkt an. Insgesamt gibt es vier verschiedene Konnektoren:

- Oracle SQL Connector for HDFS
- Oracle Loader for Hadoop
- Oracle R Connector for Hadoop
- Oracle Data Integrator Application Adapter for Hadoop

In diesem Dokument beschränken wir uns auf die beiden ersten Konnektoren, da diese für den Austausch mit der Oracle Datenbank gedacht sind.

Der Oracle R Connector ist für das Arbeiten mit Oracles Version der statistischen Programmiersprache R gedacht.

Der Adapter für den Oracle Data Integrator nutzt die "normalen" Konnektoren. Er kann mittels generierten Hive-Code auch Daten innerhalb von Hadoop verarbeiten.

Oracle SQL Connector for HDFS

Der Oracle SQL Connector for HDFS ermöglicht über einen External Table aus der Oracle Datenbank auf Daten in HDFS zuzugreifen. Die Daten in HDFS werden durch sogenannte Location Files beschrieben, welche mit Hilfe des Konnektors angelegt werden müssen.

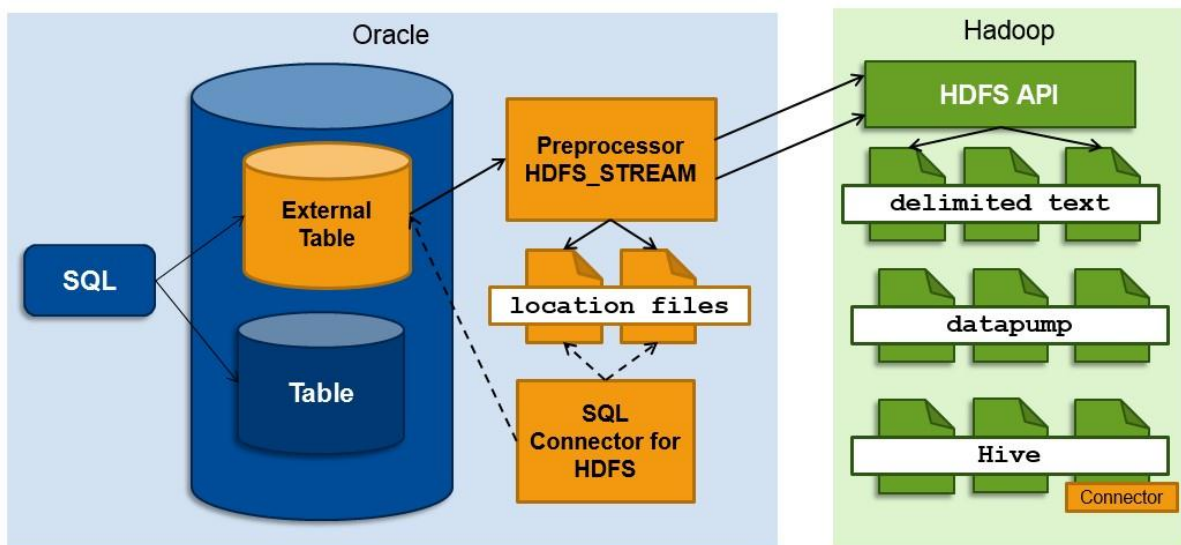


Abb. 5: Oracle SQL Connector

Per Kommandozeilentool des SQL Connectors werden die External Table und die Location Files erzeugt. Hier ein Beispiel um alle *.dat Dateien aus dem Verzeichnis ../olh_sales in dem External Table SALES_DT_XTAB bereitzustellen:

```
hadoop jar OSCH_HOME/jlib/orahdfs.jar \  
oracle.hadoop.exctab.ExternalTable \  
-D oracle.hadoop.exctab.tableName=SALES_DT_XTAB \  
-D oracle.hadoop.exctab.locationFileCount=4 \  
-D oracle.hadoop.exctab.dataPaths=  
    "hdfs:///user/scott/olh_sales/*.dat" \  
-D oracle.hadoop.exctab.columnCount=10 \  
-D oracle.hadoop.exctab.defaultDirectory=SALES_DT_DIR \  
-D oracle.hadoop.connection.url=  
    jdbc:oracle:thin:@//myhost:1521/my servicename \  
-D oracle.hadoop.connection.user=SCOTT \  
-createTable
```

Beim Abfragen der External Table greift der HDFS_STREAM Preprocessor parallel per HDFS API auf die Dateien im Hadoop Cluster zu. Pro Location File wird eine eigene Verbindung genutzt.

```
CREATE TABLE hdfuser.weather_hdfs_ext_tab (  
    stations_id VARCHAR2(4000 BYTE),  
    mess_datum VARCHAR2(4000 BYTE)  
)  
ORGANIZATION EXTERNAL (  
    TYPE ORACLE_LOADER  
    DEFAULT DIRECTORY weather_ext_dir ACCESS PARAMETERS (  
        RECORDS DELIMITED BY 0X'0A'  
        ...  
        PREPROCESSOR osch_bin_path:hdfs_stream  
        FIELDS ...  
    )  
    LOCATION ('osch-...-1', ..., 'osch-...-4')
```

Der Vorteil des SQL Connectors ist der komfortable Zugriff per External Table auf die Daten in HDFS. Datenbank-User oder Anwendungen können so transparent auf Daten in HDFS zugreifen. Zu beachten ist allerdings, dass die WHERE-Bedingung erst in der Oracle Datenbank greift, also alle Daten vor dem Filtern in die Oracle Datenbank transferiert werden müssen.

Es reicht den SQL Connector auf dem Oracle Server zu installieren. Eine Installation auf dem Cluster ist nur notwendig, wenn Hive verwendet wird.

Oracle Loader for Hadoop

Mit dem Oracle Loader for Hadoop werden Daten per Map Reduce Job direkt in die Zieltabelle geladen. Alternativ kann ein Datapump-File in HDFS erzeugt werden. Auf dieses kann man dann mit Hilfe des SQL Connectors per External Table zugreifen.

Die Quelldaten können in den verschiedensten Formaten vorliegen: Delimited Text, Avro, Hive und sogar Custom Formats.

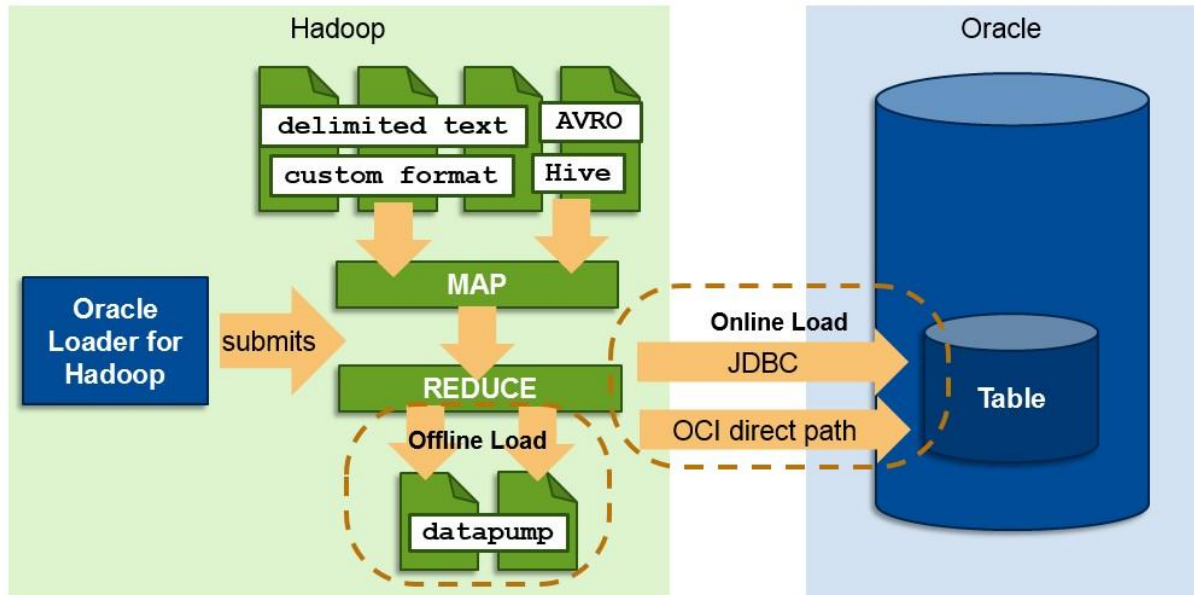


Abb. 6: Oracle Loader for Hadoop

Beim Online Load, wenn die Daten also direkt in die Zieltabelle geschrieben werden, kann dies per JDBC oder per OCI direct path Load gemacht werden.

Der Loader wird mittels einer XML Datei konfiguriert. Die Zieltabelle muss dabei in der Datenbank schon existieren.

```
<?xml version="1.0" ?>
<configuration>
  <property>
    <name>mapreduce.inputformat.class</name>
    <value>oracle.hadoop.loader.lib.input.DelimitedTextInputFormat</value>
  </property>
  <property>
    <name>mapred.input.dir</name>
    <value>/user/mfhadoop/produkt_klima_Tageswerte_20111025_20121023_15000.txt</value>
  </property>
  <property>
    <name>oracle.hadoop.loader.input.fieldNames</name>
    <value>F0,F1,F2,F3,F4,F5,F6</value>
  </property>
  <property>
    <name>mapreduce.outputformat.class</name>
    <value>oracle.hadoop.loader.lib.output.JDBCOutputFormat</value>
  </property>
</configuration>
```

```

<property>
  <name>oracle.hadoop.loader.targetTable</name>
  <value>HDFSUSER.KLIMA_TAGESWERTE</value>
</property>
<property>
  <name>mapred.output.dir</name>
  <value>/user/oracle/loader_output</value>
</property>
<property>
  <name>oracle.hadoop.loader.connection.url</name>
  <value>jdbc:oracle:thin:@//localhost:1521/orcl</value>
</property>
<property>
  <name>oracle.hadoop.loader.connection.password</name>
  <value>mfhadoop</value>
</property>
<property>
  <name>oracle.hadoop.loader.connection.user</name>
  <value>HDFSUSER</value>
</property>
</configuration>

```

Der Oracle Loader for Hadoop lässt sich per Kommandozeile starten. Die Daten werden abhängig von der Konfiguration direkt in die DB oder in ein Data Pump File in HDFS geschrieben.

```

> export OLH_HOME="/home/oracle/connectors/oraloader-2.2.0-h1"
>
> export HADOOP_CLASSPATH="$HADOOP_CLASSPATH:/home/oracle/connectors/oraloader-
2.2.0-h1/jlib/*"
> hadoop jar $OLH_HOME/jlib/oraloader.jar oracle.hadoop.loader.OraLoader
    -conf loader_config.xml

```

Der Oracle Loader for Hadoop ist auf allen Cluster Knoten zu installieren.

Fazit

Für den Datentransfer von HDFS in die Oracle Datenbank stehen mit dem Open Source Tool Sqoop und dem kommerziellen Oracle Big Data Konnektoren verschiedene Alternativen zur Auswahl. Als erstes wird man sicherlich aus Kostengründen auf Sqoop zurückgreifen. Sollte die Performance nicht ausreichen oder ist ein Zugriff aus der Datenbank per External Table notwendig, können die Oracle Konnektoren in Betracht gezogen werden.

Wenn Daten von der Oracle DB nach HDFS transferiert werden, ist Sqoop zurzeit die einzige Möglichkeit, da die Oracle Big Data Konnektoren (Version 2.2) dies zumindest aktuell nicht unterstützen. Generell bietet es sich dabei an, die Daten als Hive Daten abzulegen, so dass die Tabellen-Metadaten über die Struktur der Tabelle auch in Hadoop zur Verfügung stehen.

Kontaktadresse:

Carsten Herbe
metafinanz Informationssysteme GmbH
Leopoldstr. 146
D-80804 München

Telefon: +49 (0) 89 360531-5039
Fax: +49 (0) 89 360531-5015
E-Mail: carsten.herbe@metafinanz.de
Internet: www.metafinanz.de