Big Data: Die ersten Schritte eines Oracle Experten

Jan Ott
Senior Consultant Business Intelligence
Trivadis AG
Glattbrugg

Schlüsselworte

Big Data, Oracle Connectoren, Hadoop, Claudera, Oracle

Einleitung

Die schöne neue Big Data Welt lockt. Die Verheißungen sind groß. Doch die ersten Schritte in der Welt können überraschend sein. Daher hat der Autor seine ersten Schritte dokumentiert und zeigt dabei, wie schnell und einfach wir die Daten aus der Hadoop- in die Oracle–Welt integrieren können.

Ein paar Worte zu Big Data

Big Data wird immer mit 3, 4 oder 5 V's in Verbindung gebracht.

- Volumen – die Menge an Daten
- Velocity – die Analyse der Datenströme
- Variety – die Arten von Daten
- Veracity – die Unsicherheit von Daten (IBM)
- Value – der Wert der Daten (Microsoft)

Ein Projekt muss nicht alle V's aufweisen um in Big Data Kategorie zu gehören. Es reicht schon eins, wenn es ausgeprägt genug ist. Zum Beispiel wenn hunderte von Terabytes an Daten vorhanden sind.

Ein Big Data Projekt muss nicht zwangsweise Hadoop voraussetzen. Viele können auch mit andern Mitteln, zum Beispiel mit einer Oracle Datenbank, realisiert werden.

Hadoop wurde in meinem Projekt als Möglichkeit große Datenmengen außerhalb der Datenbank kostengünstig zu speichern angesehen.

Hadoop und sein Zoo

Hadoop ist eine Art Zoo. Hadoop wird selten alleine eingesetzt. Es werden zusätzliche Komponenten wie HBase, Hive, Impala und weitere eingesetzt. Dazu kommt noch der Zookeeper mit dem man den ganzen Zoo managed.

Doch damit nicht genug. Weitere Schlüsselworte fallen, wie NoSQL Datenbanken, Semantic Web. Das Ganze wird dann abgeschlossen mit neuen Wegen der Architektur wie – Lambda, die „Big" mit „Fast" verbindet.

Am Ende des Tages geht es jedoch immer darum, aus den Daten Erkenntnisse zu gewinnen.

Hadoop

Hadoop ist ein Filesystem, HDFS.

Der Grundstein legte Google mit mehreren „White Papers“ die ein Filesystem beschrieben das auf

- gängiger Hardware läuft
- Ausfallsicherheit bietet
- massive horizontale Skalierung ermöglicht
- strukturierte und unstrukturierte Daten verarbeitet

Die Dokumente von Google wurden von der Open Source Gemeinschaft mit Freuden aufgenommen. Daraus entstand in kurzer Zeit ein Apache Open Source Project – Hadoop.

Die ersten Schritte in der Big Data – Hadoop Welt

Diese ganze Welt kann verwirren, daher gilt es zu fokussieren und die Aufgabenstellung bewusst einfach zu halten. Daraus entstanden die folgenden Schritte mit dem Ziel, Daten aus der Hadoop Welt in Oracle zu selektieren:

- Daten aus Oracle exportieren und als File speichern
- Das File in Hadoop einfügen
- Externe Tabelle definieren
- Daten in Oracle per SQL selektieren

Weitere Rahmenbedingungen:

- Kein Java
- Bekannte Daten verwenden => EMP aus SCOTT/TIGER
- Im ersten Schritt nur den Weg mit Oracle Connector – SQL Connector gehen
- Die Umgebung muss alles anbieten => VM Big Data Light von Oracle

Umgebung – Oracle Big Data Light

http://www.oracle.com/technetwork/database/bigdata-appliance/oracle-bigdatalite-2104726.htm

Diese enthält:

- Oracle Datenbank 12c (12.1.0.1)
- Cloudera Distribution (CDH4.5)
- Oracle Big Data Connector 2.4, das heisst Oracle SQL Connector for HDFS 2.3.0
- Oracle SQL Developer 4.0

Zudem benötigt es noch die Oracle Virtual Box.

In der Zwischenzeit ist die Version 4.0.1 von Oracle Big Data Light zum Download bereitgestellt worden. Diese enthält 12c (12.1.0.2). Der Vortrag ist jedoch noch mit 2.4.1 erstellt. Die Scripte, die hier vorgestellt werden, laufen auch in 4.0.1.

Informationen zu der VM

- Login – Oracle/welcome1
- Webseite mit Informationen zu den Logins und weiterem Funktionen
  file:///home/oracle/GettingStarted/StartHere.html
- Starten der DB – Icon auf dem Desktop „Start the Oracle DB“

Damit ist die Vorbereitung abgeschlossen.

Oracle stellt ein Beispiel zur Verfügung – Movie Datenbank.

Die Schritte

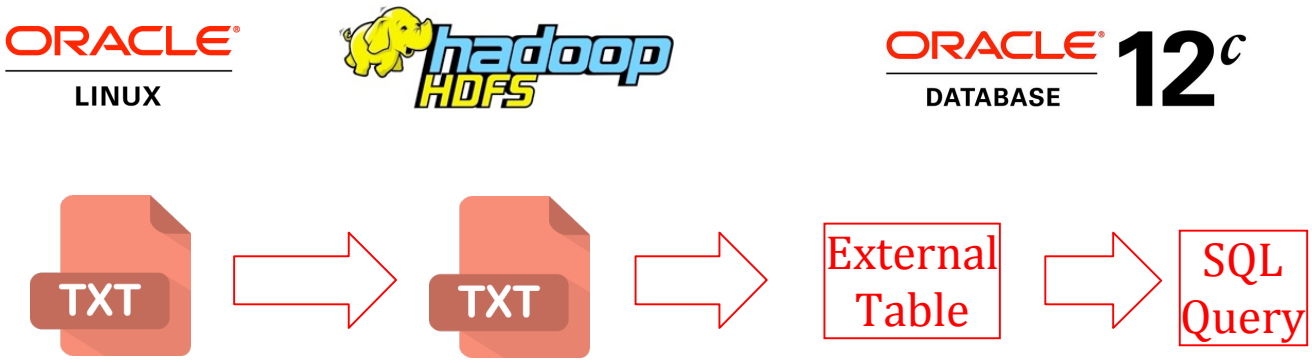

Abb. 1: Datenfluss

1. Schritt Daten aus Oracle in ein File schreiben

EMP aus Oracle SCOTT/TIGER in ein File schreiben – Komma getrennt.

```
SQL> SELECT empno || ',' || ename || ',' || job || ','
          || mgr || ',' || hiredate || ',' || sal || ','
          || comm || ',' || deptno
        FROM emp
       ORDER BY empno;
```

2. Schritt File ins Hadoop – HDFS schreiben

Zuerst ein neues Verzeichnis in HDFS anlegen

```
$ hadoop fs –mkdir demo_hdfs
```

Dann das File hineinkopieren

```
$ hadoop fs -put /home/oracle/Desktop/demo/emp.txt demo_hdfs
```

3. Schritt Oracle vorbereiten und externe Tabelle erstellen

Rechte vergeben.

```
SQL> GRANT READ, WRITE, EXECUTE ON DIRECTORY OSCH_BIN_PATH TO scott;

SQL> CREATE OR REPLACE DIRECTORY demo_dir
         AS '/home/oracle/Desktop/demo';

SQL> GRANT READ, WRITE, EXECUTE ON DIRECTORY demo_dir TO scott;
```

Kommando Zeile zur Erstellung der 1. externen Tabelle verwenden.

```
$ hadoop oracle.hadoop.exttab.ExternalTable \
-D oracle.hadoop.exttab.tableName=EMP_HDFS_EXT_TAB \
-D
oracle.hadoop.exttab.columnNames=empno,ename,job,mgr,hiredate,sal,comm,dept
```

```
no \
-D oracle.hadoop.exttab.locationFileCount=1 \
-D oracle.hadoop.exttab.dataPaths=demo_hdfs/*.txt \
-D oracle.hadoop.exttab.columnCount=8 \
-D oracle.hadoop.exttab.defaultDirectory=demo_dir \
-D oracle.hadoop.connection.url=jdbc:oracle:thin:@localhost:1521:orcl \
-D oracle.hadoop.connection.user=SCOTT \
-D oracle.hadoop.exttab.printStackTrace=true \
-createTable
```

Das erstellt die folgende Tabelle:

```
CREATE TABLE "SCOTT"."EMP_HDFS_EXT_TAB”
(
 "EMPNO"                          VARCHAR2(4000),
 "ENAME"                          VARCHAR2(4000),
 "JOB"                            VARCHAR2(4000),
 "MGR"                            VARCHAR2(4000),
 "HIREDATE"                       VARCHAR2(4000),
 "SAL"                            VARCHAR2(4000),
 "COMM"                           VARCHAR2(4000),
 "DEPTNO"                         VARCHAR2(4000)
)
ORGANIZATION EXTERNAL
(
   TYPE ORACLE_LOADER
   DEFAULT DIRECTORY "DEMO_DIR”
   ACCESS PARAMETERS
   (
     RECORDS DELIMITED BY 0X'0A'
     CHARACTERSET AL32UTF8
     PREPROCESSOR "OSCH_BIN_PATH":'hdfs_stream'
     FIELDS TERMINATED BY 0X'2C'
    MISSING FIELD VALUES ARE NULL
     (
       "EMPNO" CHAR(4000),
       "ENAME" CHAR(4000),
       "JOB" CHAR(4000),
       "MGR" CHAR(4000),
       "HIREDATE" CHAR(4000),
       "SAL" CHAR(4000),
       "COMM" CHAR(4000),
       "DEPTNO" CHAR(4000)
     )
   )
   LOCATION
   (
     'osch-20140319061232-9144-1'
   )
) PARALLEL REJECT LIMIT UNLIMITED;
```

Gleichzeitig wird ein „Location File“ mit dem folgenden Inhalt erstellt.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<locationFile>
    <header>
        <version>1.0</version>
```

```
        <fileName>osch-20140319061232-9144-1</fileName>
        <createDate>2014-03-19T18:12:32</createDate>
        <publishDate>2014-03-19T06:12:32</publishDate>
        <productName>Oracle SQL Connector for HDFS Release 2.3.0 -
Production</productName>
        <productVersion>2.3.0</productVersion>
    </header>
    <uri_list>
        <uri_list_item size="605"
compressionCodec="">hdfs://bigdatalite.localdomain:8020/user/oracle/demo_hd
fs/emp.txt</uri_list_item>
    </uri_list>
</locationFile>
```

Somit bleibt nur noch das Selektieren in Oracle über die externe Tabelle. Dies dauert auf einem Notebook mit Big Data Light Version 2.4.1 eine ganze Minute und das reproduzierbar.

Dabei wird ein Log – File und wenn nötig ein Bad – File geschrieben. Hier ein Auszug des Log – Files:

```
cat /home/oracle/Desktop/demo/EMP_HDFS_EXT_TAB*
 LOG file opened at 03/19/14 20:06:31

KUP-05007:   Warning: Intra source concurrency disabled because the
preprocessor option is being used.

Field Definitions for table EMP_HDFS_EXT_TAB
  Record format DELIMITED, delimited by 0A
  Data in file has same endianness as the platform
  Rows with all null fields are accepted

  Fields in Data Source:

    EMPNO                           CHAR (4000)
      Terminated by "2C”
      Trim whitespace same as SQL Loader
    ENAME                           CHAR (4000)
      Terminated by "2C”
      Trim whitespace same as SQL Loader
...
```

Das Log – File sieht aus, wie wir uns das von externen Tabellen oder dem SQL Loader von Oracle gewöhnt sind.

Nochmals - was machen wir: 14 Rows werden über eine externe Tabelle in Oracle gelesen. Die Daten sind jedoch in Hadoop. Da müssen verschiedene Prozesse gestartet werden. Dieser Vorlauf kostet Zeit. Klar kann man das Verbessern indem man verschiedene Parameter in Hadoop – Cloudera ändert. Das hat Oracle auch gemacht und in Version 4.0.1 sind es noch 2 Sekunden. Jedoch sind es immer noch nicht die Hundertstel-Sekunden, die mit Oracle sonst normal sind.

Projekt

Für dieses Projekt ist man auf Hadoop gewechselt für die angelieferten Files. Diese wurden in der Vergangenheit nach dem Laden gelöscht. Das wird nun nicht mehr gemacht.

- Die Files werden 1 zu 1 in Hadoop geladen.

- Die Files werden nicht gelöscht.
- Keine Entscheide betreffend Tabellendefinition (Schema-less)
- Keine Änderungen auf den Files.
- Die Daten werden über externe Tabellen wie vorher in den DWH – Staging Bereich geladen.
- Die Daten sind aber für die Analysten in Hadoop verfügbar.

Für die Zukunft geht man davon aus, dass die Analysten in den Daten Perlen finden werden. Da man die ganze Historie der Daten hat, kann man das Datawarehouse komplett neu laden. Auch wenn in Zukunft entschieden werden sollte, dass man auf einer tieferen Granularität sein will, gibt es zwei Optionen. Erstens die Daten nachzuladen oder zweitens, sie über externe Tabellen aus Hadoop zur Verfügung zu stellen.

Zudem wird im Moment analysiert, was die nächsten Schritte sind. Was kann aus dem Hadoop Zoo verwendet werden, um den ETL Prozess aus der Datenbank auszulagern und die Datenbank dadurch zu entlasten.

Zusammenfassung

Hadoop ist nicht Big Data. Sondern ein Tool mit dem Big Data Aufgaben gelöst werden können. Big Data ist ein Überbegriff. Big Data kann auch mit einer Oracle Datenbank und bestehenden Prozessen realisiert werden.

Ganz wichtig: Hadoop ist in erster Linie ein Filesystem!

Hadoop – HDFS hat eine Blockgrösse von 128 MB (default) und ist somit nichts für kleine Files.

Hadoop kennt keine Indexe. Dazu müssen andere Module aus dem Zoo verwendet werden. Dies bedeutet, dass es immer „full table scans“ sind.

Es ist eine neue Welt. Fangt an Daten zu sammeln.

Kontaktadresse:
Jan Ott
Trivadis AG
Europa-Strasse 5
CH-8152 Glattbrugg

Telefon:	+41 (0) 58-459 55 55
Fax:	+41 (0) 58-459 55 95
E-Mail	jan.ott@trivadis.com
Internet:	www.trivadis.com