YCBS-257 - Data at Scale

Workshop 6

Hadoop File Formats

General Instructions:

Finding the right file format for your particular dataset can be tough. Different Hadoop applications also have different affinities for file formats. The purpose of this workshop is to manipulate most popular Hadoop's file formats, Avro and Parquet, in order to help you to understand each one and to help you to choose the best one for your dataset.

Online resources:

https://avro.apache.org/
http://kitesdk.org/docs/1.1.0/

1 - Avro

Exercise 1: Creating Avro Files

In this exercise you will convert the Bixi dataset to be stored as an Avro file(s). To perform this task you will use **Kite Software Development Kit** which is an open source tool developed by Cloudera and available in the Cloudera distribution.

- In this exercise we assume that the Bixi files are stored on your local file system under `/home/cloudera/Downloads/Bixi/Data`
- To use kite-sdk, run this command at the command line: `kite-dataset`
- `kite-dataset help` will print kite-dataset usage details

1. Open a new Terminal window
2. Navigate to your Bixi data folder and enter this command to print an Avro schema inferred from the Stations csv file.

```
$ kite-dataset csv-schema Stations_2018.csv --
  record-name Station
```

3. Save the inferred schema to a file and name it station.avsc

```
$ kite-dataset csv-schema Stations_2018.csv --
  record-name Station -o station.avsc
```

4. Create a new empty Avro dataset container based on the stations.avsc schema and name it stations_avro

```
$
```

5. Print the information about the new Avro container

```
$
```

6. Import all Stations_2018.csv rows into the Avro stations container

$

7. Compare how many rows imported into the Avro container with the original csv file rows count. Is there any difference?

   $

At this stage, kite-sdk has created a new Avro file and stored it into the default path of Hive on HDFS.

8. List the content of the Hive HDFS path (/user/hive/warehouse/)

   $

9. What is the size of the avro table? Compare it to the original file size.

   $

Let's read the new avro table using Impala

10. Open a new Impala session

    $

11. Check for stations_avro table

    $

12. Write a query to show the first 10 rows of the stations_avro table

    $

13. Close the impala session

    $

2 - Parquet

Exercise 1: Creating Parquet Files

In this exercise you will convert the **ratings.csv** file to be stored as a Parquet file. While creating the Parquet file you will also create a Hive partition (`year ,month`) and then load the file into a Hive / Impala table. Same as the previous section, you will use **Kite Software Development Kit** to perform this task.

- In this exercise we assume that the **ratings.csv** file is stored on your local file system under `/home/cloudera/Downloads/ml-20m/`

Preparing the dataset:

- The ratings.csv dataset has a column timestamp (column 4). To be able to read this timestamp using Hive we should multiply the value by 1000.
- Create a new Pig Latin script to normalize the `ratings.csv` file prior converting it to Parquet format. We assume that the output file is `ratingsNorm.csv`.

- Creating Partioned Parquet file from the ratings dataset:

1. Open a new Terminal window and navigate to your ml-20m folder
2. Infer the schema from ratingsNorm csv file and save it to a file as . `rating.avsc`
   $
3. Create a new **partition config** and save the partition information into a json file `year-month.json`
   $
4. Create a new empty Parquet dataset named ratings based on the ratings.avsc schema and using the partition config file `year-month.json`
   $
5. Print the information about the new Parquet container
   $
6. Import all ratingsNorm.csv rows into the Parquet ratings container
   $

At this stage, kite-sdk has created a new parquet file and stored it into the default path of Hive on HDFS.

7. List the content of the Hive HDFS path `(/user/hive/warehouse/)`
   $
8. What is the size of the parquet table? Compare it to the original file size.
   $

Open a new Impala session

9. Show the information about ratings table
10. Write a query to list the content of the partition `year=2015, month=03`