1. **Verify the cluster health including HDFS and Spark**

Cloudera manager contains everything in one place. There is one Hadoop cluster, educluster1. History Server Web UI shows submitted jobs. Name Node UI shows status of Data Nodes and Name Nodes.

|  |  |
| --- | --- |
| **HDFS** | **Spark2** |
|  |  |
|  |  |
|  |  |

1. **Test the spark environment by executing the spark’s sort.py example.**

* List existing examples

ls /opt/cloudera/parcels/SPARK2/lib/spark2/examples/src/main/python/

* Copy example files to data

cd data

cp /opt/cloudera/parcels/SPARK2/lib/spark2/examples/src/main/python/sort.py sort.py

* Execute it

spark2-submit data/sort.py use\_cases/input\_sort\_py.txt

The input file contains series of numbers in multiple lines. The above job sorted them and printed the final result to the terminal.

0 0 0 1 1 1 2 2 2 2 2 3 3 3 7 7 7 8 8 8 8 8 10 10 14 14 15 15

1. **Try to implement the same example in scala and perform spark-submit.**

mkdir scala\_directory

cd scala\_directory

create build file vi build.sbt

sbt compile

mkdir -p src/main/scala/com/edureka/training

src/main/scala/com/edureka/training

mkdir -p src/main/scala/com/edureka/training/m4/

* Create a new file sort.scala and add scala code

import org.apache.spark.sql.SparkSession

object ScalaSort {

/\*\* Usage: HdfsTest [file] \*/

def main(args: Array[String]) {

if (args.length < 1) {

System.err.println("Usage: HdfsTest <file>")

System.exit(1)

}

val spark = SparkSession.builder.appName("ScalaSortTGA").getOrCreate()

val lines = spark.read.text(args(0)).rdd.map(r => r(0))

val sortedCount = lines.flatMap(line => line.split(' ')).map(\_toInt).sortByKey()

val output = sortedCount.collect()

for (x <- output){

println(x)

}

spark.stop()

}

}

* Compile, Package

sbt compile

sbt package

* Submit

spark2-submit --class com.edureka.training.m4.ScalaSort --deploy-mode client target/scala-2.11/sparkme-project\_2.11-1.0.jar use\_cases/input\_sort\_py.txt

1. **Analyze the behavior of spark application on Spark web UI**

I went to spark web UI <http://bdlabs.edureka.co:50014> and checked the Jobs, Stages, and Executors

**5. Add custom logs in your application and re-execute the application. Once**

**executed check the Spark logs.**

* Create source directory “m4” in project source folder

mkdir -p src/main/scala/com/edureka/training/m4/

- Copy HdfsTest.scala in “m4” cp/opt/cloudera/parcels/SPARK2/lib/spark2/examples/src/main/scala/org/apache/sp ark/examples/HdfsTest.scala src/main/scala/com/edureka/training/m4/

- Change package to

com.edureka.training.m4

* Add this dependency to build.sbt

libraryDependencies += "com.typesafe.scala-logging" %% "scala-logging" % "3.9.0"

* Add this import on top of HdfsTest.scala and add custom logs

import com.typesafe.scalalogging.Logger

logger.info("Hello there!")

* Compile, Package

sbt compile

sbt package

* Submit

spark2-submit --class com.edureka.training.m4.HdfsTest --deploy-mode client target/scala-2.11/sparkme-project\_2.11-1.0.jar use\_cases/input\_sort\_py.txt

- Collect logs from yarn

yarn logs --applicationId application\_1528714825862\_137643

1. **Transfer complete dataset from RDBMS to HDFS**

* Upload 6 csv files using ftp
* Login to mysql, change database, create table

mysql -h mysqldb.edu.cloudlab.com -u labuser --password=edureka

create database instacart672184

use instacart672184

* create tables using this script

**CREATE TABLE `aisles` (**

**`aisle\_id` int(11) NOT NULL AUTO\_INCREMENT,**

**`aisle` varchar(50) DEFAULT NULL,**

**PRIMARY KEY (`aisle\_id`)**

**);**

**CREATE TABLE `departments` (**

**`department\_id` int(11) NOT NULL AUTO\_INCREMENT,**

**`department` varchar(50) DEFAULT NULL,**

**PRIMARY KEY (`department\_id`)**

**);**

**CREATE TABLE `order\_products\_prior` (**

**`order\_id` int(11) DEFAULT NULL,**

**`product\_id` int(11) DEFAULT NULL,**

**`add\_to\_cart\_order` int(11) DEFAULT NULL,**

**`reordered` int(11) DEFAULT NULL**

**);**

**CREATE TABLE `order\_products\_train` (**

**`order\_id` int(11) DEFAULT NULL,**

**`product\_id` int(11) DEFAULT NULL,**

**`add\_to\_cart\_order` int(11) DEFAULT NULL,**

**`reordered` int(11) DEFAULT NULL**

**);**

**CREATE TABLE `orders` (**

**`order\_id` int(11) NOT NULL,**

**`user\_id` int(11) DEFAULT NULL,**

**`eval\_set` varchar(20) DEFAULT NULL,**

**`order\_number` int(11) DEFAULT NULL,**

**`order\_dow` int(11) DEFAULT NULL,**

**`order\_hour\_of\_day` int(11) DEFAULT NULL,**

**`days\_since\_prior\_order` float(3,1) DEFAULT NULL,**

**PRIMARY KEY (`order\_id`)**

**);**

**CREATE TABLE `products` (**

**`product\_id` int(11) NOT NULL,**

**`product\_name` varchar(100) DEFAULT NULL,**

**`aisle\_id` int(11) DEFAULT NULL,**

**`department\_id` int(11) DEFAULT NULL,**

**PRIMARY KEY (`product\_id`)**

**);**

**CREATE TABLE `sample\_submission` (**

**`order\_id` int(11) NOT NULL,**

**`products` varchar(20) DEFAULT NULL,**

**PRIMARY KEY (`order\_id`)**

**) ;**

show tables;

Tables\_in\_instacart672184

| aisles |

| departments |

| order\_products\_prior |

| order\_products\_train |

| orders |

| products |

| sample\_submission |

+---------------------------+

* Load csv into table

load data local infile '/mnt/home/edureka\_672184/data/aisles.csv' into table aisles FIELDS TERMINATED BY ',' ENCLOSED BY '\"' LINES TERMINATED BY '\n' IGNORE 1 LINES;

load data local infile '/mnt/home/edureka\_672184/data/departments.csv' into table departments FIELDS TERMINATED BY ',' ENCLOSED BY '\"' LINES TERMINATED BY '\n' IGNORE 1 LINES;

load data local infile '/mnt/home/edureka\_672184/data/order\_products\_\_prior.csv' into table order\_products\_prior FIELDS TERMINATED BY ',' ENCLOSED BY '\"' LINES TERMINATED BY '\n' IGNORE 1 LINES;

load data local infile '/mnt/home/edureka\_672184/data/order\_products\_\_train.csv' into table order\_products\_train FIELDS TERMINATED BY ',' ENCLOSED BY '\"' LINES TERMINATED BY '\n' IGNORE 1 LINES;

load data local infile '/mnt/home/edureka\_672184/data/orders.csv' into table orders FIELDS TERMINATED BY ',' ENCLOSED BY '\"' LINES TERMINATED BY '\n' IGNORE 1 LINES;

load data local infile '/mnt/home/edureka\_672184/data/products.csv' into table products FIELDS TERMINATED BY ',' ENCLOSED BY '\"' LINES TERMINATED BY '\n' IGNORE 1 LINES;

load data local infile '/mnt/home/edureka\_672184/data/sample\_submission.csv' into table sample\_submission FIELDS TERMINATED BY ',' ENCLOSED BY '\"' LINES TERMINATED BY '\n' IGNORE 1 LINES;

* Sqoop import

sqoop import --connect jdbc:mysql://sqoopdb.edu.cloudlab.com/instacart672184 --username labuser -password edureka --table aisles -m 1 --target-dir /user/edureka\_672184/use\_cases/instacart/aisles/

sqoop import --connect jdbc:mysql://sqoopdb.edu.cloudlab.com/instacart672184 --username labuser -password edureka --table departments -m 1 --target-dir /user/edureka\_672184/use\_cases/instacart/ departments /

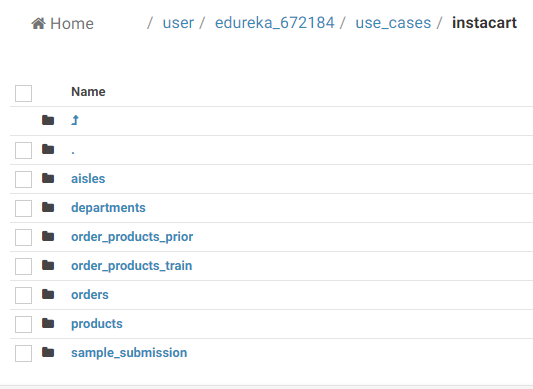
sqoop import --connect jdbc:mysql://sqoopdb.edu.cloudlab.com/instacart672184 --username labuser -password edureka --table order\_products\_prior -m 1 --target-dir /user/edureka\_672184/use\_cases/instacart/order\_products\_prior/

sqoop import --connect jdbc:mysql://sqoopdb.edu.cloudlab.com/instacart672184 --username labuser -password edureka --table order\_products\_train -m 1 --target-dir /user/edureka\_672184/use\_cases/instacart/order\_products\_train/

sqoop import --connect jdbc:mysql://sqoopdb.edu.cloudlab.com/instacart672184 --username labuser -password edureka --table orders -m 1 --target-dir /user/edureka\_672184/use\_cases/instacart/orders/

sqoop import --connect jdbc:mysql://sqoopdb.edu.cloudlab.com/instacart672184 --username labuser -password edureka --table products -m 1 --target-dir /user/edureka\_672184/use\_cases/instacart/products/

sqoop import --connect jdbc:mysql://sqoopdb.edu.cloudlab.com/instacart672184 --username labuser -password edureka --table sample\_submission -m 1 --target-dir /user/edureka\_672184/use\_cases/instacart/sample\_submission/

****

**7. Validate the loaded data by comparing the statistics of data both in source and HDFS**

hdfs dfs -cat /user/edureka\_672184/use\_cases/Instacart/aisles/\*|wc -l

SELECT count(\*) FROM aisles;

= 134

hdfs dfs -cat /user/edureka\_672184/use\_cases/Instacart/products/\*|wc -l

SELECT count(\*) FROM products;

= 49355, hdfs shows 49400

hdfs dfs -cat /user/edureka\_672184/use\_cases/Instacart/orders/\*|wc -l

SELECT count(\*) FROM orders;

= 3421083

hdfs dfs -cat /user/edureka\_672184/use\_cases/instacart/order\_products\_train/\*|head -5

1,49302,1,1

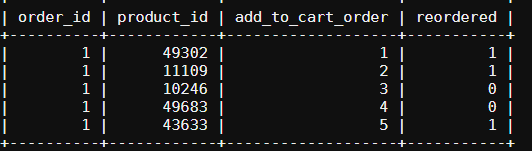
1,11109,2,1

1,10246,3,0

1,49683,4,0

1,43633,5,1

SELECT \* FROM order\_products\_train LIMIT 5;



**8. Create a new directory in HDFS named cheeses and load only rows where aisle is “specialty cheeses“**

- create directory

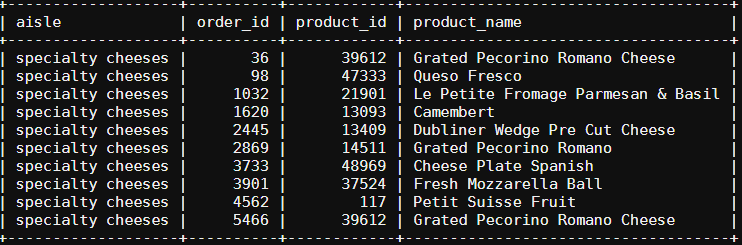
hdfs dfs -mkdir /user/edureka\_672184/use\_cases/instacart/cheeses

* Test this query on mysql by creating a view

CREATE VIEW chessy AS

SELECT a.aisle, op.order\_id, p.product\_id, p.product\_name FROM aisles a JOIN products p ON a.aisle\_id = p.aisle\_id JOIN order\_products\_train op ON op.product\_id = p.product\_id WHERE a.aisle LIKE 'specialty cheeses'

SELECT \* FROM chessy LIMIT 10;



sqoop import --connect jdbc:mysql://mysqldb.edu.cloudlab.com/instacart672184 --username labuser --password edureka --query 'SELECT \* FROM chessy WHERE $CONDITIONS' -m 1 -target-dir '/user/edureka\_672184/instacart/cheeses'

**9. update “specialty cheeses” to “ specialty cheese” and transfer only updated rows in the above created directory.**

- update table through the view

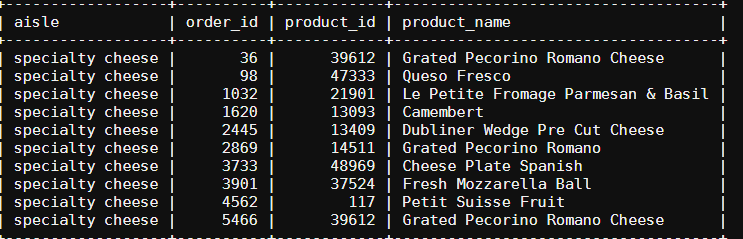
UPDATE cheesy SET aisle = ‘specialty cheese’ WHERE aisle LIKE ‘specialty cheeses’;

* Recreate the view

CREATE VIEW chessy AS

SELECT a.aisle, op.order\_id, p.product\_id, p.product\_name FROM aisles a JOIN products p ON a.aisle\_id = p.aisle\_id JOIN order\_products\_train op ON op.product\_id = p.product\_id WHERE a.aisle LIKE 'specialty cheese'

SELECT \* FROM chessy LIMIT 10;



* Create a column that stores last modified date for each row: last\_mod
* Transfer updated view based last modified date

sqoop import --connect jdbc:mysql://mysqldb.edu.cloudlab.com/instacart672184 --username=labuser --password=edureka --table chessy -target-dir '/user/edureka\_672184/cheeses' --incremental lastmodified --check-column last\_mod --merge-key order\_id --last-value 2019.07.16 -m 1