1. Star Wars Data

a. Upload data to HBase

1. First create all tables with column family cf

create ‘characters’, ‘cf’

create ‘species’, ‘cf’

create ‘planets’, ‘cf’

create ‘starships’, ‘cf’

create ‘vehicles’, ‘cf’

2. Insert all the data in HBASE

- Characters:
  name,height,mass,hair_color,skin_color,eye_color,birth_year,gender,homeworld,species

bin/hbase org.apache.hadoop.hbase.mapreduce.ImportTsv -Dimporttsv.separator=',' -Dimporttsv.columns='HBASE_ROW_KEY,cf:name, cf:height, cf:mass, cf:hair_color, cf:skin_color, cf:eye_color, cf:birth_year, cf:gender, cf:homeworld, cf:species' characters /home/javi/Desktop/Workspaces/data-star/characters.csv

- Species

bin/hbase org.apache.hadoop.hbase.mapreduce.ImportTsv -Dimporttsv.separator=',' -Dimporttsv.columns='HBASE_ROW_KEY,cf:name,cf:classification,cf:designation,cf:average_height,cf:skin_colors,cf:hair_colors,cf:eye_colors,cf:average_lifespan,cf:language,cf:homeworld' species /home/javi/Desktop/Workspaces/data-star/species.csv

- Planets

bin/hbase org.apache.hadoop.hbase.mapreduce.ImportTsv -Dimporttsv.separator=',' -Dimporttsv.columns='HBASE_ROW_KEY,cf:name,cf:rotation_period,cf:orbital_period,cf:diameter,cf:climate,cf:gravity,cf:terrain,cf:surface_water,cf:population' planets /home/javi/Desktop/Workspaces/data-star/planets.csv

- Starships

bin/hbase org.apache.hadoop.hbase.mapreduce.ImportTsv -Dimporttsv.separator=',' -Dimporttsv.columns='HBASE_ROW_KEY,cf:name,cf:model,cf:manufacturer,cf:cost_in_credits,cf:length,cf:max_atmosphering_speed,cf:crew,cf:passengers,cf:cargo_capacity,cf:consum

ables,cf:hyperdrive_rating,cf:MGLT,cf:starship_class' starships /home/javi/Desktop/Workspaces/data-star/starships.csv

- Vehicles

bin/hbase org.apache.hadoop.hbase.mapreduce.ImportTsv -Dimporttsv.separator=',' -Dimporttsv.columns='HBASE_ROW_KEY,cf:name,cf:model,cf:manufacturer,cf:cost_in_credits,cf:length,cf:max_atmosphering_speed,cf:crew,passengers,cf:cargo_capacity,cf:consumables,cf:vehicle_class' vehicles /home/javi/Desktop/Workspaces/data-star/vehicles.csv

b. Draw 5 unique insights from the datasets, analytical code can be in pySpark, python Pandas, or Scala

Done in jupyter-notebook

1. PySpark

a. What is an RDD?

RDD is a data structure of Spark, this structure is immutable.

b. What is a DAG ?

DAG is a set of vertices and edges that represents the actions that will be applied to the RDDs, this DAG will split the graph into stages.

c. What is the role of a spark driver in the spark cluster?

A spark driver is the application that launches the main method of Spark. The driver coordinates the workers and the execution of tasks. This is executed using schedulers like DAGScheduler and TaskScheduler.

d. Is Spark fault Tolerant and how does Spark achieve that?

One of the reasons because Spark is fault Tolerant is because Spark runs in file-systems like HDFS. But this is not true for streaming data. Another reason is because Rdd are immutable and because in case of loss partition, it can be re-computed from the original.

2. Shell

a. How to make a shell script executable ?

To make a shell script executable you should give permissions of execution to that file with command chmod

b. What is the use of “#!/bin/bash” ?

This line goes at the beginning of a bin script file. This line says that the code will run in a bash environment.

c. How do you resolve variable in a shellscript ?

In shellscript you use the dollar sign to resolve variables.

3. Hadoop

a. What are the core components of Hadoop?

The thre core components in Hadoop are its filesystem HDFS, its resource manager YARN and its software programming model to compute large data files MapReduce.

b. What is the difference between nameNodes and dataNodes?

The main difference is the nameNode is the master node and is in charge of the file system metadata and the datanodes is the slave node who stores the data instructed by the namenode.

c. What does jps command do in Hadoop?

The jps command lists the java hotspots vms running.

d. What do you mean by metadata in Hadoop?

In hadoop the metadata is stored in the namenode, it is data about data. So, the namenode could store information about size, partitions, offset...

e. What is a block in HDFS, why block size 64MB?

A block in HDFS is a splitted shard of a bigger file. One big data file in HDFS is divided into blocks. It is a default configuration and the size could depend on the network traffic between datanodes.