

Spark & HIve



Akolom Gebrezgabhier (ID#108229)

Maged Samaan (ID# 984790)

Spark Installation

* If we do Map Reduce in Cloudera virtual box, there is no need for a separate spark installation. It is already installed in it.
* We can test the existence of spark by trying to run some of its commands like:

spark-shell

spark-submit

spark-submit --class packageName.ClassName --master local ProjectName.jar Input Output

* All the procedures we followed for Hadoop Map Reduce will be the same we just use different spark commands.
* Somehow if we can’t find it installed, the steps are: -

Download file from this link (<https://www.virtualbox.org/wiki/Downloads>)

**Eclipse Setup**

* First we create a java project
* Right click on the Project > configure > convert to maven
* Inside the POM file that will be created we append the following lines of code:

<dependencies>

<dependency>

<groupId>org.apache.spark</groupId>

<artifactId>spark-core\_2.10</artifactId>

<version>1.3.1</version>

</dependency>

</dependencies>

Finally, we create the java files inside the project and we generate a jar file

Overview of the project

* In this project we tried to analyze logs collected from an online shopping games and movies website
* We processed the log files using spark and we came out with the following outputs:
* Top ten users by the amount of time they spent visiting the website.
* And much more analysis can be done on demand.
* We have 3 classes namely UserSummarize, User, Functions

Details

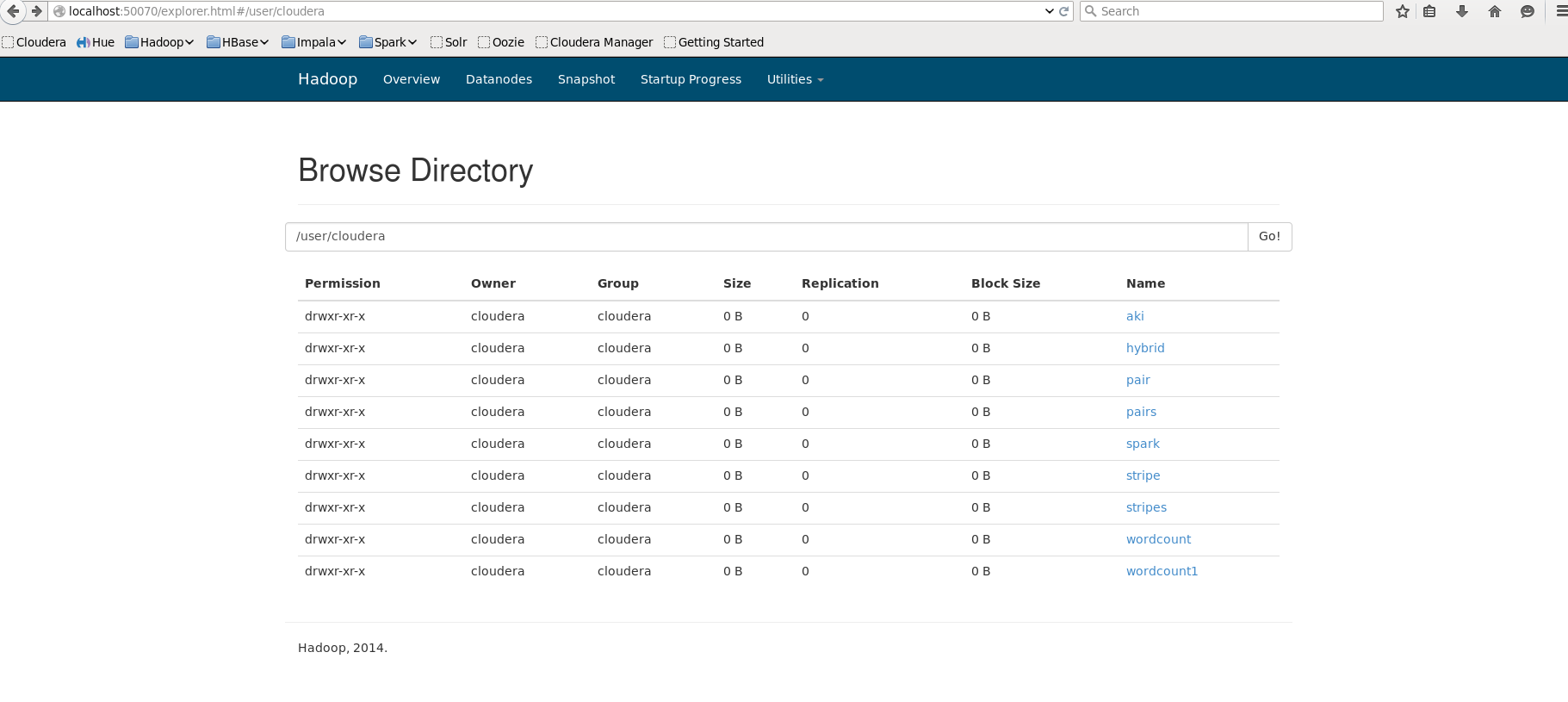
* Execute Spark job by handover jar file, main class name, input location and output location via following terminal commands.

hdfs dfs –mkdir spark/input

hdfs dfs –put input spark

spark-submit --class sparkPackage.UserSummarize --master local SparkProject.jar spark/input spark/output

* you can find the output by browsing <http://localhost:50070/> (see the below image)



* you can read the output files using the below command:

hdfs dfs –cat spark/output/part-00000

* Or you can run simply on the consul directly
* Challenges to the project: To know the spark technology was little bit hard because it’s new technology we couldn’t find much resource.

What is the purpose of this project?

* After analyze top 5 users visiting a website it’s better to send an update information
* This analyze helps to know which area is more users available.
* This data analyze helps to some company to introduce an advertisement to the website

**Introduction**

Apache Hive is a data warehouse infrastructure built on top of Hadoop for providing data summarization, query, and analysis. So you can use queries to manipulate your relational data. In this project we will use Hive to store the input data which is relational (can be stored in form of tables, columns, and relations).

Note: we will use the console to do all our work.

**Store Your Data**

* Open the console and run hive command to be able to write the hive commands through the shell.

[cloudera@quickstart ~]$ **hive**

* Now you have your hive interface active, so you can start write your queries

Logging initialized using configuration in file:/etc/hive/conf.dist/hive-log4j.properties

WARNING: Hive CLI is deprecated and migration to Beeline is recommended.

hive>

* Here are some commands to show you how to be able to see the database, tables, columns and table structure in hive platform.

hive> **show databases**;

default

hive> **show tables**;

* Create a database for your project, and start suing it

hive> **create database** BigDataDB;

* if you have many databases, and you need to know which database you’re currently using, this below command will be helpful.

hive> **set** hive.cli.print.current.db=true;

hive (bigDataDb)>

* You should create a table who with the same format as your data, use the below query to create the input table

hive> **create table** input (subject1 String, subject2 String, subject3 String, subject4 String) ROW FORMAT DELIMITED FIELDS TERMINATED BY ‘ ‘;

* You can check structure of a table by using

hive> **describe input**;

id int

subject1 int

subject2 int

subject3 int

subject4 int

* Loading data from the input file to the newly created table:

hive (bigDataDb)> LOAD DATA LOCAL INPATH 'input.txt' INTO TABLE input;

* Loading data from the input file to the newly created table:

hive (bigDataDb)> select \* from input;

425 450 466 471

472 522 525 475

545 440 422 435

545 522 525 475

Now you can use JDBC to connect Java using Hive so you can read the data, or you can use hdfs to run map and reduce on the hive data.

**Creating Batch file**

* And add the below commands to a file and call it script.hql

create database BigDataDB;

use bigdatadb;

create table input (subject1 String, subject2 String, subject3 String, subject4 String) ROW FORMAT DELIMITED FIELDS TERMINATED BY ' ';

LOAD DATA LOCAL INPATH 'input.txt' INTO TABLE input;

select \* from input;

* Then run the command belwo

[cloudera@quickstart ~]$ hive -f script.hql