CentraleSupélec 2016-2017 **Big Data Platforms : Tutorial 1**

MapReduce - Hadoop

The purpose of this tutorial is to get you started with MapReduce and Hadoop (HDFS and MapReduce execution), two core elements of the Big Data stack. You will work with JAVA and the ECLIPSE Integrated Development Environment

After completing this lab, you will be able to :

— Use Hadoop commands to explore the HDFS on the Hadoop system.

— Use Hadoop commands to run MapReduce programs on the Hadoop system.

Exercice 1 Hadoop installation

Hadoop can be run in three modes :

— **Standalone (or local mode)** : in this mode, no daemons are used (a slingle JVM). Hadoop uses the local file system as an substitute for HDFS file system. In this configuration, we considerer an execution environment with 1 mapper and 1 reducer. This mode is used for the development step.

— **Pseudo-distributed mode** : this mode mimics the behavior of a cluster but all the daemons run a a single machine (different JVMs are locally executed). This mode can accept multiple mappers and reducers and uses HDFS.

— **Fully-distributed mode** : hadoop on a real cluster.

In this tutorial, you will not use a real hadoop cluster and as a consequence, you will work only in local mode (standalone or psudo-distributed).

A first work is to install Hadoop on your machine, either directly or with a virtual machine according to your OS (Linux is better for Hadoop). Nevertheless, as a first exercice, you will try to install directly Hadoop on your machine.

Question 1.1 First case : installing Hadoop directly

The manual installation of Hadoop is quite simple. According to your OS, you can refer to these different tutorials :

— **Windows** : `https://wiki.apache.org/hadoop/Hadoop2OnWindows` (I will not be able to give you many support for Windows)

— **Linux** : `https://hadoop.apache.org/docs/stable/hadoop-project-dist/hadoop-common/SingleCluster.html`

— **Mac OS** : `https://dtflaneur.wordpress.com/2015/10/02/installing-hadoop-on-mac-osx-el-capitan/`

Most of the time, installing Hadoop is done with the following steps :

1. System update and required software installations.
   — Java must be installed (at least v6 or v7). You can verified that you have a correct version of java using the following command :
   `$ java - version`
   — ssh must be installed and sshd must be running to use the Hadoop scripts that manage remote Hadoop daemons.
2. Building of a group and a specific user for Hadoop (not mandatory but highly recommanded).

```
$ addgroup hadoop
$ adduser --ingroup hadoop hadoopuser
$ adduser hadoopuser
```

3. Configuration of ssh to enable to access to localhost for the hadoopuser

```
$ ssh-keygen -t rsa -P ""
$ cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
$ ssh localhost
```

4. The Hadoop installation
   (a) Downloading Hadoop from a mirror site : `https://www.apache.org/dyn/closer.cgi/hadoop/common/`
   (b) Untar(zip) and installation of the archive in a chosen repertory : MYHADOOPREPERTORY
   (c) Go to this repertory

   ```
   $ cd MYHADOOPREPERTORY
   ```

   (d) Specify the location of JAVA into the file `etc/hadoop/hadoop-env.sh`

   ```
   export JAVA_HOME={JAVA_HOME}
   ```

   (e) Try the following command

   ```
   $ ./bin/hadoop
   ```

5. Hadoop configuration through the editing of different configurations file
   (a) Configuring Hadoop in standalone mode by editing the file `etc/hadoop/core-site.xml`

   ```
   <configuration>
       <property>
           <name>fs.defaultFS</name>
           <value>hdfs://localhost:9000</value>
       </property>
   </configuration>
   ```

   Here, the name of the file system has been specified : all the files and repertories in HDFS will be prefixed by `hdfs://localhost:9000`.
   (b) Configuration of the parameters of the HDFS file system by editing the file `etc/hadoop/hdfs-site.xml`

   ```
   <configuration>
       <property>
           <name>dfs.replication</name>
           <value>1</value>
       </property>
   </configuration>
   ```

   (c) Specification of the parameters of MapReduce by editing the file `etc/hadoop/mapred-site.xml`

   ```
   <configuration>
       <property>
           <name>mapreduce.framework.name</name>
           <value>yarn</value>
       </property>
   </configuration>
   ```

   Here, we specify that Yarn is used as the implementation of MapReduce.
   (d) At last, we can parameter Yarn by editing the file `etc/hadoop/yarn-site.xml`

   ```
   <configuration>
       <property>
           <name>yarn.nodemanager.aux-services</name>
           <value>mapreduce_shuffle</value>
       </property>
   </configuration>
   ```

Here, we precise that an operation SHUFFLE will be used.

6. At this step, Hadoop is correctly installed and configured, we will just have to format the local HDFS system file and to start Hadoop

```
$ hdfs namenode -format
$ start-dfs.sh
$ start-yarn.sh
```

Question 1.2 Second case : Using a virtual machine environment

An alternative (recommanded for not linux OS) is to use a virtual machine environment and a package distribution of Hadoop. You also have to do this work in order to use Hadoop in pseudo-distributed mode easily. For this case, we will follow the following Hadoop tutorial : `http://web.stanford.edu/class/cs246/homeworks/hw0/tutorialv3.pdf`.

1. Download and install *VirtualBox* on your machine : `https://www.virtualbox.org/wiki/Downloads`.
2. Download the *Cloudera Quickstart VM* : `https://downloads.cloudera.com/demo_vm/virtualbox/cloudera-quickstart-vm-5.5.0-0-virtualbox.zip`.
3. Uncompress the VM archive.
4. Start *VirtualBox* and click *Import Appliance* in the *File* dropdown menu. Click the folder icon beside the location field. Browse to the uncompressed archive folder, select the .ovf file, and click to the *Open* button. Click the *Continue* button. Click the *Import* button.
5. Your virtual machine shoud now appear in the left column. Select it and click on *Start* to lauch it.
6. To verify that the VM is running and to access it, open a browser to the URL : `http://localhost:8088`. You should see the ressource manager UI. The VM uses port forwarding for the common Hadoop ports, so when the VM is running, those ports on localhost will redirect to the VM.

Exercice 2 Exploring Hadoop Distributed File System (HDFS)

In this part of the tutorial, you will learn about HDFS (how it works, best practices...). HDFS allows user data to be organized in the form of files and directories. At the end of this part, you will ba able to :

— Verify HDFS is running and check the health of an HDFS file system.
— Get data in and out of HDFS using both the HDFS commands and the HDFS java API
— Understand strategies for streaming data into HDFS.

Question 2.1 Main HDFS commands

In this part, we will overview the main and most frequently used HDFS commands. A more comprehensive list of file system commands and options can be found on the Hadoop project website : `http://hadoop.apache.org/docs/r2.8.0/hadoop-project-dist/hadoop-common/FileSystemShell.html`

Try and understand the following commands from the shell of your Hadoop client :

1. `hdfs dfs` (same as `hadoop fs`)
2. `hdfs dfs -ls` : this command displays the files on the Hadoop file system. (Try the linux command `ls -l`) to see the difference)
3. `hdfs dfs -ls /`
4. `hdfs dfs -mkdir fichiersLab1` : to create a directory for this lab
5. Build a file that contains the word Hello and named *Hello.txt* and add it to HDFS with the following command :
   `hdfs dfs -put Hello.txt`

6. Try `hdfs dfs -ls -R`.
7. `hdfs dfs -cat Hello.txt` : to view the content of the file on HDFS
8. `hdfs dfs -cp Hello.txt Salut.txt` : to copy the file. Verify using `hdfs dfs -ls`.
9. `hdfs dfs -rm Salut.txt` : to remove a file. Verify using `hdfs dfs -ls`.
10. Try `hdfs dfs -rm Hello.txt` and `hdfs dfs -copyfromLocal Hello.txt` . What's happened ?
11. `hdfs dfs -get Hello.txt Salut.txt` : download the file from HDFS back to the client.

Question 2.2 HDFS Best Practices

A common practice for HDFS storage is to place files of a common data set into the same folder. The following steps are common in HDFS :

1. Start by making a folder for the dataset.
2. Upload all the targeted files into this folder.
3. Verify the files are there within the `hdfs dfs -ls` command.
4. Use the command `hdfs dfs -cat folder/*` to view all the content of a folder as a single file.
5. `hdfs dfs -getmerge folder/* result.txt` to download all the files of a folder and return them as a single file. Why this command useful ?

Question 2.3 HDFS Block Storage

HDFS splits file into blocks and those blocks are then distributed to data node managed by HDFS. For a disk, a block is the minimum amount of data that can be read or write. In HDFS, the size of a block is by default 128MB. Files in HDFS are broken into block-sized chunks, which are stored as independent units.

The command `hdfs fsck / -files -blocks` lists akk the blocks that make up each file in the filesystem. You can also specify the size of a block size when you upload file on HDFS. For instant, try to put the file Categorie.csv using the following commands :

— `hdfs dfs - D dfs.blocksize=500 -put Categorie.csv folder/`
— `hdfs dfs - D dfs.blocksize=1200500 -put Categorie.csv folder/`
— `hdfs dfs - D dfs.blocksize=1048576 -put Categorie.csv folder/`

Question 2.4 HDFS Health check

The command `hdfs dfsadmin -report` gives you a health check report of the HDFS file system.

Question 2.5 On your own !

1. Upload the file *arbres.csv* from the datasets in Claroline into a TreeDataSet folder on HDFS. We will use this file in a next section.
2. We will work with the meteorological data available here : `https://www1.ncdc.noaa.gov/pub/data/noaa/`. Try to understand these data. You will have to build a script (.sh) that will download and store on HDFS the data of four of your choice for the year 2015,2016 and 2017.

Question 2.6 Working with the HDFS API

In this part, we will use the HDFS JAVA API and we will use Eclipse. You can use the template project given in the course website. For this, you will have to :

1. Download the file `Lab1Hadoop.zip` on your computer.
2. In Eclipse, click on `File > Import ...`

3. In the General category, select `Existing Projects into Workspace`
4. Select `select archive file` and find the targeted archive.
5. Click on `Finish`.

Question 2.6 .1 A first simple program

In a first program, you will write a program that counts the lines of the file `arbres.csv` previously put on HDFS. For this, you have to build a new Eclipse project. Look at the code with attention. To run your program, you can follow the section 2.2 and 2.3 of the following tutorial : `http://web.stanford.edu/class/cs246/homeworks/hw0/tutorialv3.pdf`.

Question 2.7 Displaying the content of a CSV file

The objective of this exercice is to display the the year and the height of each tree of the file `arbres.csv`. For this, you will have to build a new Eclipse project. You will have to upload this project on Claroline.

Question 2.8 Displaying the content of a compact file

We will work with the file `isd-history.txt` of the NOAA. This file is available here : `https://www1.ncdc.noaa.gov/pub/data/noaa/isd-history.txt`. Put it on HDFS and then use the command `hdfs -dfs - tail file` to see this last lines. Your job is to write a program that will display the USAF code, the name, the country (FIPS country ID) and the elevation of each station.

Some important points :

— The first 22 lines of the file have to be ignored (they do not contain data).
— The name of each station begins at the character 13 of a line and its size is 29 characters long.
— The FIPS begins at the character 43 of a line and its size is 2 characters long.
— The altitude begins at the character 74 of a line and its size is 7 characters long.

You also have to upload this project on Claroline

Exercice 3 Map Reduce : Wordcount

In this section, you will write the Hello world of MapReduce : WordCount. You will work on the text of Anna Karenina of Tolstoy available here : `http://www.textfiles.com/etext/FICTION/anna_karenina`. You can also use the template eclipse project given on the website of the course and follows the instructions given in the section 2.5 of the Hadoop tutorial : `http://web.stanford.edu/class/cs246/homeworks/hw0/tutorialv3.pdf`. The main points are :

— Build a new project by copying the template project.
— Create a new package called `ecp.Lab1.WordCount`.
— Create three classes in that package : `WordCountMapper` that extends the class `Mapper` of the Hadoop java API, `WordCountReducer` that extends the class `Reducer` of the Hadoop java API and `WordCountDriver` that extends the class `Configured` and implements the interface `Tool` of the Hadoop java API. Write the code of these classes in order to count the number of occurences of each word in a given document.
— Compile your code either using the menu `Run as -> Run Configurations` menu or directly using `javac`.

```
$ export HADOOP_CLASSPATH=$($HADOOP_HOME/bin/hadoop classpath)
$ javac -classpath $HADOOP_CLASSPATH WordCount*.java
```

— Then export your code as a .jar either using the `Export` menu or directly using :

```
$ mkdir -p ecp/Lab1/wordcount
$ mv *.class ecp/Lab1/wordcount
$ jar -cvf ecp_Lab1_wordcount.jar -C . ecp
```

— Use Hadoop to run WordCount on your file.
`$ hadoop jar ecp_Lab1_wordcount.jar ecp.Lab1.wordcount.WordCountDriver /input/anna-karenina.txt /results`
— Run the command `hadoop fs -ls /results`
— Use the interface at `http://localhost:8088` to see the logs of the job

Exercice 4 Our own MapReduce programs

This part of the work will have to be upload on Claroline. For each MapReduce problem, you have to submit the following files :

— the source code of your program in JAVA.
— A brief document that gives the answer to the question asked in the problem description.
— A screen-shot image of your EMR Job Flows console that shows your program's COMPLETED state as well as the elapsed time and your AWS account name (you will see this in a next course).

Question 4.1 Problem 1 : TF-IDF

Your goal is to calculate Term Frequency ?Inverse Document Frequency (TF-IDF) of a set of documents using MapReduce. In particular, you will use the following documents that you will add to HDFS

```
hadoop fs -mkdir input
 wget http://www.textfiles.com/etext/FICTION/defoe-robinson-103.txt
hadoop fs -copyFromLocal defoe-robinson-103.txt input
wget http://www.textfiles.com/etext/FICTION/callwild
hadoop fs -copyFromLocal callwild input
```

Some links :

— `https://nlp.stanford.edu/IR-book/html/htmledition/tf-idf-weighting-1.html`
— `http://www.dcs.bbk.ac.uk/~dell/teaching/cc/book/ditp/ditp_ch4.pdf`

Which 20 words have the highest tf-idf scores in these documents ? List them in descending order.

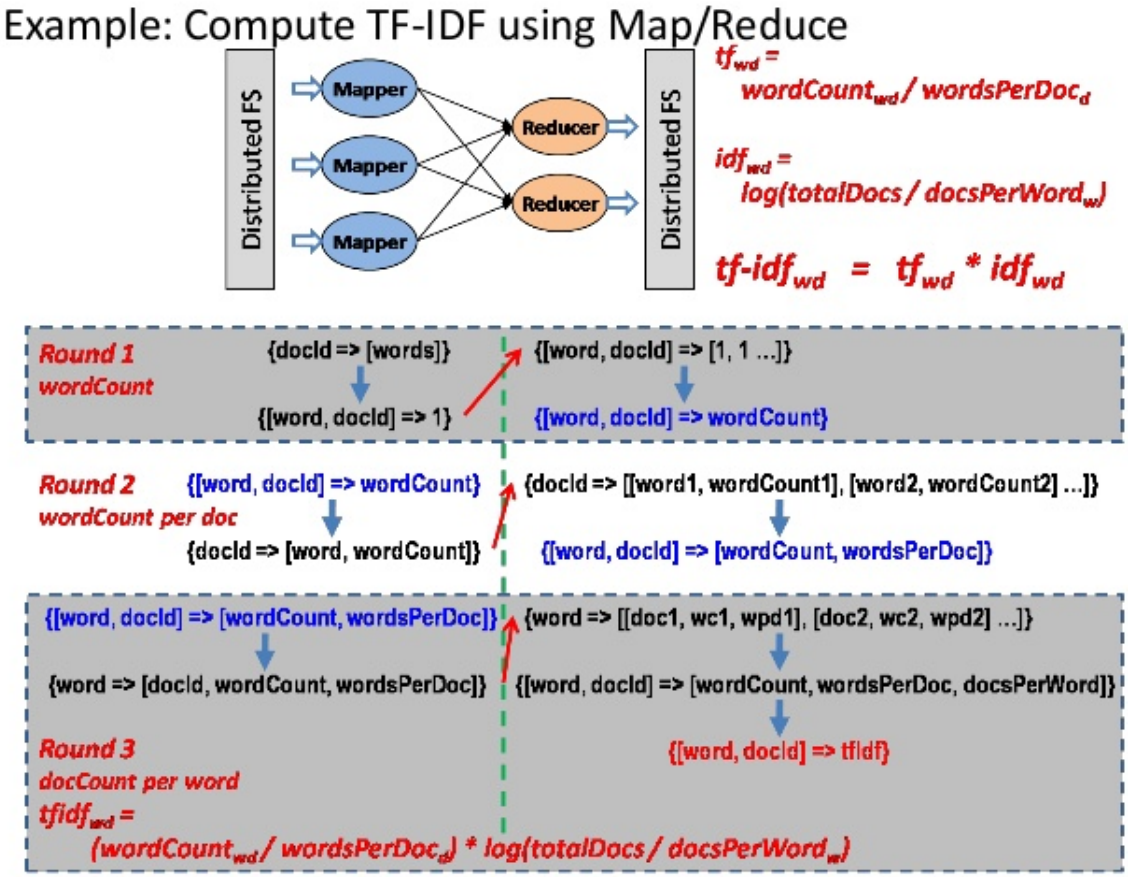

FIGURE 1 – Source : `https://www.slideshare.net/subhaskghosh/01-map-reduce-concepts`

Question 4.2 Problem 2 : Page Rank

Write a MapReduce program to calculate the PageRank score (with damping factor 0.85) for each user in the Epinions who-trust-whom online social network : `https://snap.stanford.edu/data/soc-Epinions1.html`. Which 10 users have the highest PageRank scores in this social network ? List them in descending order.

Some links for this problem :

— `http://www.dcs.bbk.ac.uk/~dell/teaching/cc/dell_cc_07.pdf`
— `http://www.dcs.bbk.ac.uk/~dell/teaching/cc/book/ditp/ditp_ch5.pdf`
— `http://infolab.stanford.edu/~ullman/mmds/ch5.pdf`
— `http://www.dcs.bbk.ac.uk/~dell/teaching/cc/paper/mlg10/lin_mr_graph.pdf`