

Lab Session 1¹

HDFS and MapReduce

1. Accessing the cluster

We'll use the 10 Mistral nodes, which will have been reserved and configured by your teacher with the Hadoop environment.

```
ssh [username]@formation.plafrim.fr  
ssh plafrim
```

That will take you to the mistral01 node, from where you will work. To set the environment variables required to give you access to the Hadoop cluster:

```
source /var/tmp/bigdata/user-env.sh
```

You will also need to change some settings in order to use maven to compile your programs (you just need to do it once for the whole semester):

```
cp /var/tmp/settings.xml ~/.m2/settings.xml
```

2. First steps with HDFS

Have a look at the list of commands available at HDFS:

```
hdfs dfs
```

For more information on a given command, you can use:

¹ Parts of this were adapted from Prof. David Auber, from the *Université de Bordeaux*
<https://www.labri.fr/perso/auber/BigDataGL/index.html>

```
hdfs dfs -usage [COMMAND]
```

There are hdfs-versions of the traditional *mkdir*, *ls*, *chmod*, etc.

- Try exploring the folders and files already present in the cluster's HDFS.
- Create a folder for yourself at `/users/[YOUR USERNAME]`.
- Add a file to your folder (send it to HDFS) using the `hdfs dfs -put` command.
- Where is your file stored? In how many pieces?

```
hdfs fsck / -files -blocks -locations
```
- Obtain the `/data/worldcitiespop.txt` file (from HDFS to the machine you are using) with the `hdfs dfs -get` command.

3. First steps with MapReduce

- Run the Pi code, provided with Hadoop as an example:

```
yarn jar /var/tmp/bigdata/hadoop/share/hadoop/mapreduce/  
hadoop-mapreduce-examples-2.8.1.jar pi 10 1000
```

- Take a look at the applications that were recently submitted to yarn, try to find the one you executed.

```
yarn application -list -appStates ALL
```

4. The Word Counter

- Recover and run the MapReduce code available at `/var/tmp/mapreduce_wordcounter.tgz`. It implements the word count algorithm we discussed earlier. To compile it:

```
cp /var/tmp/mapreduce_wordcounter.tgz ~/
cd ~
tar xzf mapreduce_wordcounter.tgz
cd mapreduce_wordcounter
mvn package
```

A target/ folder will have been created, containing a jar you can submit to yarn. You will need to provide two parameters, the input and output paths, both in HDFS. As input, use the file already present in /data/LesMiserables.txt, and as output a new (non-existing) folder under /users/[YOUR USERNAME].

- Inspect the newly created output.
- Look at the counters that are shown in the console at the end of the job, specially at the File System Counters. Compare them to the sizes of the input and output. What do they mean?
- Add a combiner to your code. In fact, you will not need to write a new class, you can reuse something. To add a combiner, add `job.setCombinerClass([CLASS])` to your main².
- Compare the counters obtained with both versions of the code.

5. World city populations

- Inspect the /data/worldcitiespop.txt file. It contains information about cities, including their population. In the first line of the file, you will find a description of all columns.
- Write a MapReduce program that receives that file as input and counts the number of cities by order of magnitude of their populations. The order of magnitude is to be calculated as `Math.pow(10, (int) Math.log10(population))`. To each order of magnitude, we want the average, maximum and minimum population. Your output should look like this:

1	5	7	8	7	
10	174	55	99	10	
100	2187	570	999	100	
1000	20537	4498	9998	1000	
10000	21550	30600	99922	10001	
100000		3248	249305	997545	100023
1000000		269	2205586	9797536	1001553
10000000		10	13343569	31480498	10021437

² If you do not remember/know what combiners are, there is a nice explanation here: https://www.tutorialspoint.com/map_reduce/map_reduce_combiners.htm