Lab 1

This introductory lab is composed of two main tasks. Your first objective is to run your first MapReduce application on the BigData@Polito cluster. For this goal, you must learn how to import a predefined project, compile the source code, produce a jar file, copy your application on a remote machine (the gateway) and submit your application on the BigData@Polito cluster. The second objective is to write your first MapReduce application.

1. Compiling using the VSCode IDE

In this task, we will "compile" the source code of a simple Hadoop application and we will create a jar file with the class files of the project. The shared VSCode project contains the basic libraries that are needed to develop the application and create the class and jar files.
You **cannot** use this project to run locally on the PC of the Lab or on your own PC the MapReduce application (other libraries are needed to run locally this application).

The imported project is the MapReduce implementation of the word count application.

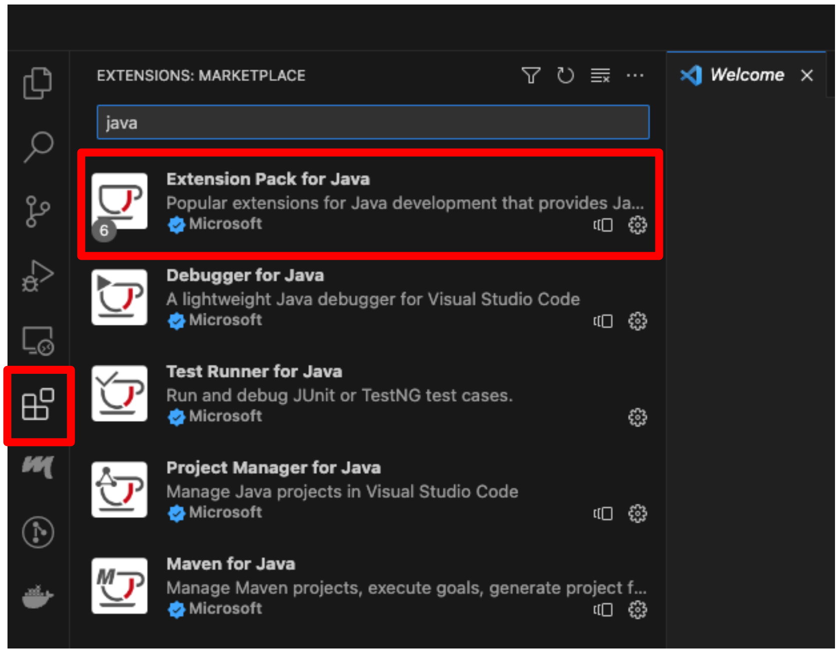
Download from the course web page the zip file Lab1_BigData_with_libraries.zip, which contains the MapReduce-based implementation of the word count application and the example data folder example_data (direct link: http://dbdmg.polito.it/dbdmg_web/wp-content/uploads/2021/10/Lab1_BigData_with_libraries.zip).

Decompress the zip file inside a local folder on the PC of the LAB or on your PC.

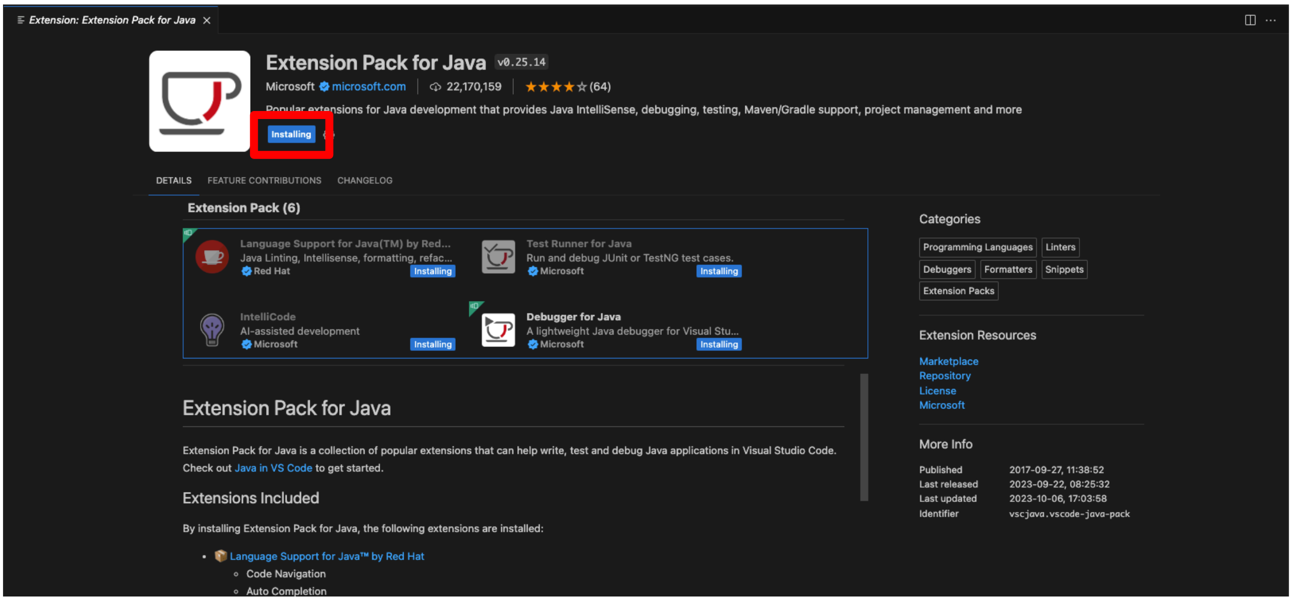
Here, you will find the steps necessary to import the project in VSCode. If you need a more complete guide on how to use VSCode throughout this course, you can find it here.

Verify that the Java extension is correctly installed on VSCode and import the project in your IDE. Finally, you will learn how to compile and export your application as a jar file.

1. Open VSCode
2. Go under Extension tab and check whether "Extension Pack for Java" is installed


EXTENSIONS: MARKETPLACE
Welcome
java
Extension Pack for Java
Popular extensions for Java development that provides Ja...
Microsoft
6
Debugger for Java
A lightweight Java debugger for Visual Studio Code
Microsoft
Test Runner for Java
Run and debug JUnit or TestNG test cases.
Microsoft
Project Manager for Java
Manage Java projects in Visual Studio Code
Microsoft
Maven for Java
Manage Maven projects, execute goals, generate project f...
Microsoft

3. If not installed, you can click on the “Install” button in the page associated to the extension.


Extension: Extension Pack for Java
Extension Pack for Java v0.25.14
Microsoft microsoft.com | 22,170,159 | ★★★★☆ (64)
Installing
DETAILS
FEATURE CONTRIBUTIONS
CHANGELOG
Extension Pack (6)
Language Support for Java(TM) by Red...
Java Linting, Intellisense, formatting, refac...
Red Hat
Installing
Test Runner for Java
Microsoft
Installing
IntelliCode
AI-assisted development
Microsoft
Installing
Debugger for Java
A lightweight Java debugger for Visual Stu...
Microsoft
Installing
Categories
Programming Languages
Linters
Debuggers
Formatters
Snippets
Extension Packs
Extension Resources
Marketplace
Repository
License
Microsoft
More Info
Published 2017-09-27, 11:38:52
Last released 2023-09-22, 08:25:32
Last updated 2023-10-06, 17:03:58
Identifier vscjava.vscode-java-pack

Popular extensions for Java development that provides Java IntelliSense, debugging, testing, Maven/Gradle support, project management and more

Run and debug JUnit or TestNG test cases.

Extension Pack for Java

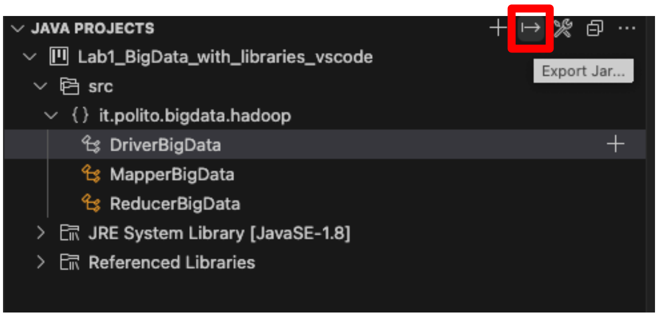
Extension Pack for Java is a collection of popular extensions that can help write, test and debug Java applications in Visual Studio Code. Check out Java in VS Code to get started.

Extensions Included

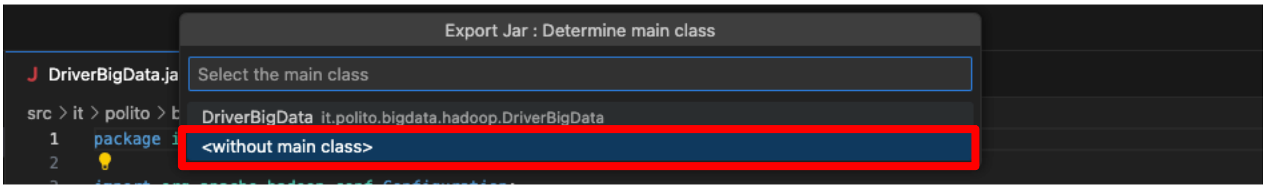
By installing Extension Pack for Java, the following extensions are installed:

- Language Support for Java™ by Red Hat
  - Code Navigation
  - Auto Completion

4. Simply Open the folder containing the project under File -> “Open folder…” or “Open folder…” in the initial screen. VSCode will automatically detect whether it’s a pure-Java project or Maven-based.
5. Have a look at the source files and the structure of the project. Where is the mapper? Where is the reducer?
6. Since we cannot use maven on the PC of the LAB, we cannot count on it to generate the jar file.
7. You can instead build a jar manually using the “Export jar…” command, as in the following steps. Remember to avoid inserting all the libraries in your jar, or you would end up producing a fat jar heavy to transfer. We need these libraries locally to compile, but they are already present in the classpath of the cluster and there is no need to include them in the jar again.
   a. In the lower-left part of VSCode, in the Java Project menu, select the Export Jar button (right arrow symbol)



   b. In the upper area, in the center, VSCode, will ask you to choose the main class for your project. Select <without main class>



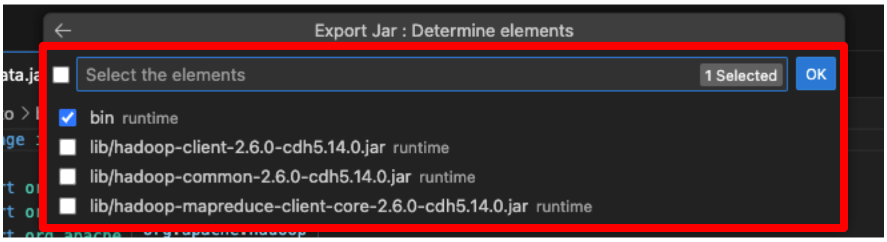
Export Jar : Determine main class

Select the main class

DriverBigData it.polito.bigdata.hadoop.DriverBigData

<without main class>

c. Next, you will be prompted to add all the files that your jar file should contain. You can keep only the content of “bin” folder (since libraries are already present on our cluster).


Export Jar : Determine elements

□ Select the elements 1 Selected OK

☑ bin runtime

□ lib/hadoop-client-2.6.0-cdh5.14.0.jar runtime

□ lib/hadoop-common-2.6.0-cdh5.14.0.jar runtime

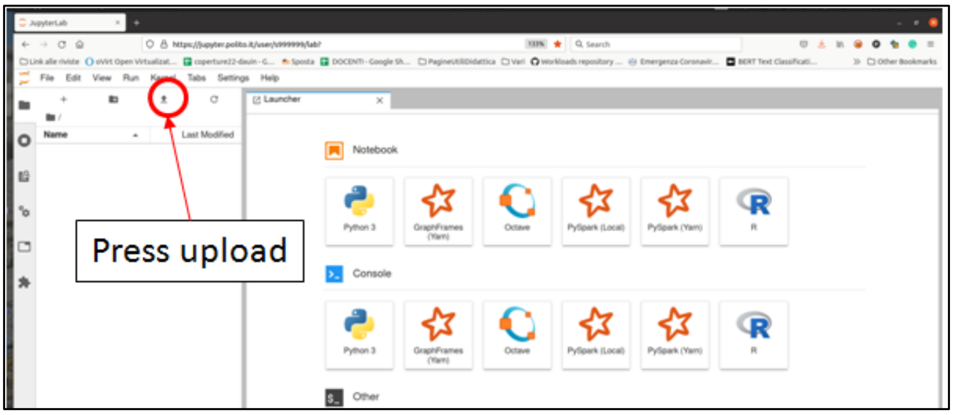
□ lib/hadoop-mapreduce-client-core-2.6.0-cdh5.14.0.jar runtime

d. You will find the jar file in the main folder of your project. You can upload such jar file directly on jupyter.polito.it cluster

2. Upload your application on BigData@Polito

The objective of this task is to upload your application on the cluster BigData@Polito.

1. Connect to https://jupyter.polito.it
2. Upload the jar file containing your application on the local file system of the gateway

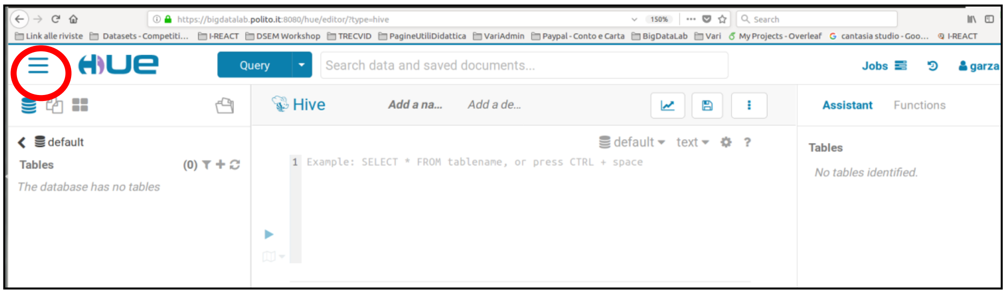

Press upload

3. Manage HDFS through the HUE web interface

In this task, you will learn how to do basic management of the HDFS file system. To this goal, we will use a web interface called HUE.

1. Go to https://hue.polito.it/ and login with your usual BigData@Polito credentials.
2. Go to the “Browsers/Files” tab. You should find **your HDFS home**, as shown below. Note that this is not the same file system as in task 2 (i.e., it is not the local file system of the gateway), so you will find probably an empty folder now.[1]
   Your **HDFS home** is not located on the gateway, but is stored in the Hadoop cluster.

[1] If the difference between the two “homes” is not clear to you at this point, do not keep on. Spend some time to clarify your ideas.

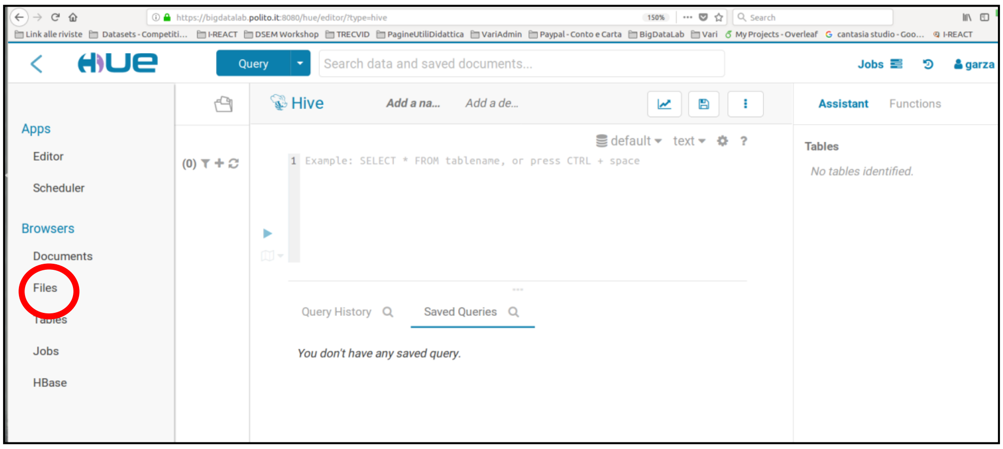

https://bigdatalab.polito.it:8080/hue/editor/?type=hive
Query
Search data and saved documents...
Jobs
garza
Hive
Add a na...
Add a de...
Assistant
Functions
default
Tables
default
text

The database has no tables

```
1 Example: SELECT * FROM tablename, or press CTRL + space
```

Tables

No tables identified.



```
1 Example: SELECT * FROM tablename, or press CTRL + space
```

You don't have any saved query.

Tables

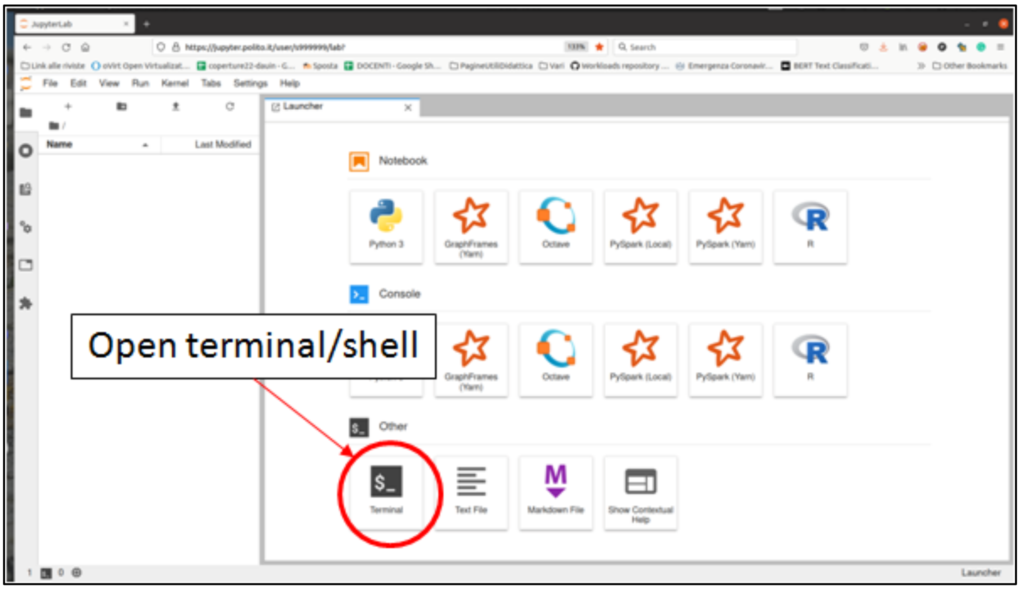
No tables identified.

3. Create the folder example_data on HDFS
4. Upload the sample files from the local folder example_data available on your PC to the folder example_data on HDFS.
5. Find out on your own how to delete/move the files, or download them. It will be helpful in the next labs.

4. Submit a job

Now we have everything we need to submit our sample application. It is finally time to open a shell.

1. Connect to https://jupyter.polito.it
2. Open a terminal clicking on “Terminal” inside the Launcher area.

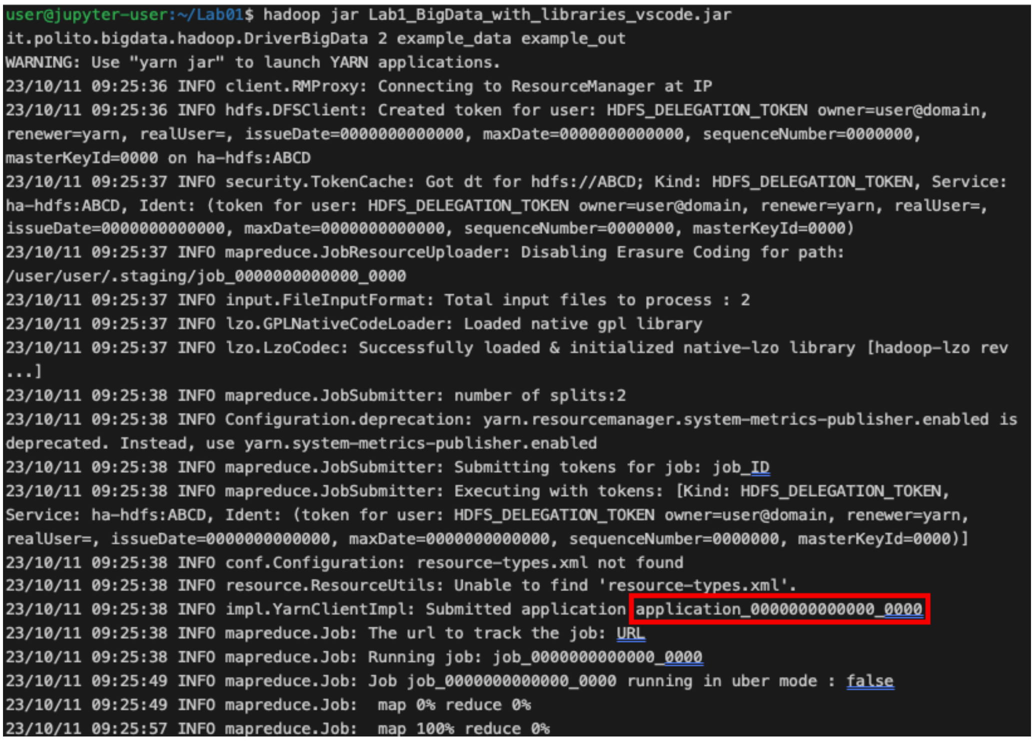


Open terminal/shell

3. Now that you have a terminal on the gateway, launch a MapReduce job (i.e., run your application on the cluster) using the following command **(which must be specified in single line, i.e., do not press enter after “Exercise1-1.0.0.jar”!)**:

```
hadoop jar Exercise1-1.0.0.jar
it.polito.bigdata.hadoop.DriverBigData 2 example_data ex1_out
```

   where:
   - “**`Exercise1-1.0.0.jar`**” is the JAR file containing your application
   - “**`example_data`**” is the input HDFS folder. A relative path starts in your home in HDFS. You can also use an absolute path in HDFS.
   - “**`ex1_out`**” is the output folder in HDFS, not on the gateway local file system. You can see its content in HUE (a relative path starts in your home in HDFS)
4. Check the number of mappers (i.e., the number of instances of the mapper class)
   - You can retrieve the information about the number of mappers (map tasks) and other statistics in the information showed on the terminal during the execution of your application (last part of the showed information).
5. Find your job from the command line. You have two alternative options available:
   a. Find the applicationId directly from the output of `hadoop` command (point 3):



```
user@jupyter-user:~/Lab01$ hadoop jar Lab1_BigData_with_libraries_vscode.jar
it.polito.bigdata.hadoop.DriverBigData 2 example_data example_out
WARNING: Use "yarn jar" to launch YARN applications.
23/10/11 09:25:36 INFO client.RMProxy: Connecting to ResourceManager at IP
23/10/11 09:25:36 INFO hdfs.DFSClient: Created token for user: HDFS_DELEGATION_TOKEN owner=user@domain,
renewer=yarn, realUser=, issueDate=0000000000000, maxDate=0000000000000, sequenceNumber=0000000,
masterKeyId=0000 on ha-hdfs:ABCD
23/10/11 09:25:37 INFO security.TokenCache: Got dt for hdfs://ABCD; Kind: HDFS_DELEGATION_TOKEN, Service:
ha-hdfs:ABCD, Ident: (token for user: HDFS_DELEGATION_TOKEN owner=user@domain, renewer=yarn, realUser=,
issueDate=0000000000000, maxDate=0000000000000, sequenceNumber=0000000, masterKeyId=0000)
23/10/11 09:25:37 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path:
/user/user/.staging/job_0000000000000_0000
23/10/11 09:25:37 INFO input.FileInputFormat: Total input files to process : 2
23/10/11 09:25:37 INFO lzo.GPLNativeCodeLoader: Loaded native gpl library
23/10/11 09:25:37 INFO lzo.LzoCodec: Successfully loaded & initialized native-lzo library [hadoop-lzo rev
...]
23/10/11 09:25:38 INFO mapreduce.JobSubmitter: number of splits:2
23/10/11 09:25:38 INFO Configuration.deprecation: yarn.resourcemanager.system-metrics-publisher.enabled is
deprecated. Instead, use yarn.system-metrics-publisher.enabled
23/10/11 09:25:38 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_ID
23/10/11 09:25:38 INFO mapreduce.JobSubmitter: Executing with tokens: [Kind: HDFS_DELEGATION_TOKEN,
Service: ha-hdfs:ABCD, Ident: (token for user: HDFS_DELEGATION_TOKEN owner=user@domain, renewer=yarn,
realUser=, issueDate=0000000000000, maxDate=0000000000000, sequenceNumber=0000000, masterKeyId=0000)]
23/10/11 09:25:38 INFO conf.Configuration: resource-types.xml not found
23/10/11 09:25:38 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
23/10/11 09:25:38 INFO impl.YarnClientImpl: Submitted application application_0000000000000_0000
23/10/11 09:25:38 INFO mapreduce.Job: The url to track the job: URL
23/10/11 09:25:38 INFO mapreduce.Job: Running job: job_0000000000000_0000
23/10/11 09:25:49 INFO mapreduce.Job: Job job_0000000000000_0000 running in uber mode : false
23/10/11 09:25:49 INFO mapreduce.Job:  map 0% reduce 0%
23/10/11 09:25:57 INFO mapreduce.Job:  map 100% reduce 0%
```

    b. On the terminal type the following command: `yarn application -list` and find the applicationId associated to your job (by finding your username).

6. Retrieve the logs as stated in the “**Retrieve the logs**” section in this document
7. Try to re-run the same job. Does it succeed this time? What’s the problem?
    - To remove a folder from HDFS, you can use HUE or you can use the hdfs command line tool executing the following command in the terminal you opened on jupyter.polito.it:
      hdfs dfs -rm -r <path of the HDFS folder you want to delete>
8. Run again the application on the cluster by changing the number of reducers (first parameter of the application) and analyze the content of the output folder of the HDFS file system.

Retrieve the logs

If you need to access the log files associated with the execution of your application by using the command line, use the following commands in the terminal of jupyter.polito.it:

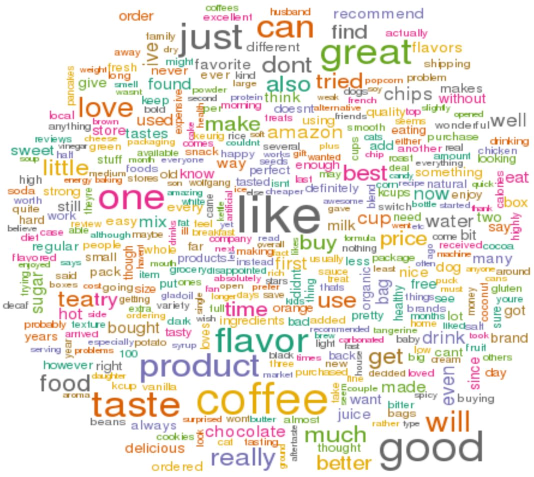
1. To retrieve the log associated with the standard output
    - ◦ yarn logs -applicationId <id of your application> -log_files stdout
        - ▪ The “id of your application” is printed on the terminal at the beginning of the execution of your application
            - □ The format of “id of your application” is application_number_number
            - □ Example of “application id” application_1584304411500_0009
        - ▪ You can retrieve the application id also from the HUE interface
    - ◦ The returned result contains one stdout log section for each task (driver, map and reduce tasks)
        - ▪ One for the Driver
        - ▪ One for each Mapper
        - ▪ One for each Reducer

2. To retrieve the log associated with the standard error
    - ◦ yarn logs -applicationId <id of your application> -log_files stderr

Test on a bigger file

Now you will execute your application on a larger file that we already uploaded on the HDFS file system of BigData@Polito.
The absolute path of that file in HDFS is the following:
/data/students/bigdata-01QYD/Lab1/finefoods_text.txt
It is a large collection of Amazon reviews in the food category. Each line of finefoods_text.txt contains one review.

1. Launch your application on the Amazon review file:
    a. Set the second parameter of your application to /data/students/bigdata-01QYD/Lab1/finefoods_text.txt
2. Analyze the results.
    a. Can you understand any interesting facts from your results?
    b. Do you see any space for improvements in your analysis?
3. The following figure was done on a small sample of your data (10000 reviews).
    a. Is it consistent with what you found on the complete dataset?
    b. Do you think a small sample is enough to represent the whole?



5. Bonus track

A word count can be seen as a special case of an n-gram count, where n is equal to 1. n-grams, in our context, are sets of contiguous words of length n.

For example, in the sentence “She sells seashells by the seashore“, 2-grams are: “She sells”, “sells seashells”, “seashells by”, “by the”, “the seashore”.

Modify your word count program to count 2-grams frequencies. Consider each line of text as a separate document, as in the Amazon reviews file (so, do not count as contiguous words on separate lines).

1. What is the time/space complexity of your program?
2. How long would you expect it to run, compared to the simple word count?
3. Try to run it on the toy text and on the Amazon reviews.