# *Hadoop Assignment*[1]

In this assignment you are going to implement a Hadoop program that gets text files as input and finds the word with maximum frequency in the whole data-set.

## Install HDP 2.2 (Hortonworks Data Platform)

Install HDP on your computer using this link. HDP is an open-source apache Hadoop framework. To be able to install HDP you will need a virtual machine. You can use VirtualBox which is free or VMWare workstation which is free for OSU engineering students. You can follow the instruction here to install and set up HDP. After installing HDP make sure that you can access the Hortonworks services using your web browser.

## Download the Hadoop

You can download it here. Extract the tar ball. In the next part you will add some of these libraries to your project.

## Eclipse Project Setup

First we need to add Hadoop libraries to our project:

1- Create an Eclipse Java Project
2- Go to Project Properties window and in "Java Build Path" section, click on "Add External Jars"
3- In the JAR Selection dialog, select the following jars from the extracted Hadoop tar.gz file.
   - `share/hadoop/common/*.jar`
   - `share/hadoop/common/lib/*.jar`
   - `share/hadoop/mapreduce/*.jar`
   - `share/hadoop/hdfs/Hadoop-hdfs-*.jar`

Next, we are going to adjust the java compiler version. Go to Project Properties window. Go to “Java Compiler” section and make sure the “compiler compliance level” is 1.7.

## Implementation

You should code the Mapper, Reducer and Application as explained in the Hadoop lecture. Once you are done you should create a runnable jar file including the libraries.

## Copying the Jar file to HDP

We are going to setup the directory ~/Desktop to be shared between our Host computer and the Hortonworks Sandbox. We need this to transfer the jar file to the Hortonworks Sandbox.

1- Before powering up the Hortonworks Sandbox, open the settings of Hortonworks Sandbox VM.
2- Navigate to "Shared Folders" section.
3- Click on the "Add Shared Folder" button present at the right of the dialog box.
4- In the Folder Path, browse to your Desktop folder.
5- Check the "Auto-mount" check box. Click on OK.

[1] This assignment has been designed using the resources provided at https://github.com/hortonworks/hadoop-tutorials

Now, you can access your ~/Desktop folder from within Hortonworks Sandbox. Check if you can access the shared folder and jar file. `Alt+F5` and login as root/hadoop. To change to mounted directories, type `cd /media/` and press Enter. Now, type `ls` and press Enter to bring up a list of shared folders. This should show `sf_Desktop` as an entry. This means that you can access ~/Desktop of your computer from within Hortonworks Sandbox VM.

We are going to run our MapReduce Job as hue user. So, we need to copy the `jar` to that user's home directory `/usr/lib/hue`, with permissions for hue to execute the same. In the command prompt, enter the command `cp /media/sf_Desktop/Your_file_name.jar /usr/lib/hue/` and press Enter.

Now, enter the command `cd /usr/lib/hue/ ; chmod 777 your_file_name.jar` and press Enter. This will make the JAR file readable and executable by all.

## Test Input Data Setup

For our MapReduce job to execute, we need test input data. The test input data is nothing but a set of files, having multiple lines, one name in each line. So, go to File Browser and create a directory `Input` under `/user/hue` directory. Inside the directory, create two files file-1.txt and file-2.txt with sample contents (Multiple lines, one name in each line).

## Run MapReduce Job Execution

We are going to run our MapReduce Job as `hue` user. Let us now change to the user `hue` by typing `su – hue` command pressing enter. Now, you are the user `hue`.

Now, we can use `hadoop jar` command to fire the job. Run the following command:

```
hadoop jar your_file_name.jar /user/hue/Input /user/hue/Output
```

Seeing the way the program is written, the above command means that the input files lie in /user/hue/Input directory and the output will be produced in /user/hue/Output directory in HDFS.

## Monitor MapReduce Job Execution

You can monitor the progress of the triggered job using Job Browser. If you wish, you could take a peek into the generated log files.

## Deliverables

- Your runnable jar file (It should be compatible with HDP 2.2 and Hadoop 2.6.0)
- Brief Description of your Map and Reduce function