



## Spark

### Objectives

- Understand Spark MapReduce programming model.
- Build several analytic tasks using Spark.

### Overview

You are required to install Spark on a single node cluster. You will then practice creating analytics applications using Spark MapReduce and then executing them using Hadoop.

### Downloading Apache Spark

Follow the following steps to download Spark on your machine:

- Download the latest version of Apache Spark (Pre-built according to your Hadoop version) from this link: [Apache Spark Download Link](#)
- Extract the downloaded file using the command:  
`tar -xvf path to downloaded compressed file`

### Setting up Apache Spark

After downloading Apache Spark, follow the coming steps to set it up:

1. If Hadoop is not installed on your machine, follow the steps in hadoop lab session.
2. Install Scala:  
`wget www.scala-lang.org/files/archive/scala-2.11.7.deb`  
`sudo dpkg -i scala-2.11.7.deb`
3. Add the following line at the end of \$HOME/.bashrc file  
`export PATH=$PATH:path to extracted spark folder/bin`  
You can open/edit the file using: `vi $HOME/.bashrc`  
After editing /.bashrcfile,execute:source \$HOME/.bashrc
4. Verify the installation using the following command:  
`spark-shell`



## Word Count

Now, you will create the Word Count application and run it as a Spark job on some files on HDFS.

- You can download the code for WordCount example from [here](#).
- Extract the downloaded zip and make it your current directory.
- Compile WordCount.java and create a jar using:  
`mvn clean install package`
- Create a text file to test the wordcount program using it and move it to HDFS using `-copyFromLocal` command
- You are now ready to run the jar file from target folder using:  
`java -jar wordcount-1.0.jar hdfs://localhost:9000/PATH_TO_INPUT_FILE_ON_HDFS`  
`hdfs://localhost:9000/PATH_TO_OUTPUT_FILE_ON_HDFS`
- Copy the outputs from HDFS to local system using `-copyToLocal` command.

## Notes

- You can work in groups of 2 or 3.

**Good Luck**