**Big Data Management - Fall 2022**

**Project ZERO: Project Precursor and Practice for Project 1**

|  |  |
| --- | --- |
| **Total Points:** | **1 point (to encourage you to get started)** |
| **Given Out:** | **Week 3: Sep 9, 2022** |
| **Due Date:** | **Week 4: Sep 16, 2022**  **Submit the project via CANVAS.** |
| **Teams:** | **Anyone is to be turned in by each student individually.** |

**Project Overview**

In this project, you will work on setting up your Hadoop ecosystem so to get ready for project 1. In addition, you will load an existing dataset we provide in CANVAS onto your Hadoop HDFS file system. Then you will practice to execute the java code we provide to you on this dataset.

The goal is to have you set up the big data environment on your own computer and start to practice how to debug the code within some editor. **You don’t need to write any code at this stage.**

There are many online resources about Hadoop setup, and we are also providing some resources below to get you started as well as on CANVAS.

**Guidelines.**

For this project, you can freely communicate with each other and help each other, and use any resource you would like, since our goal is for you to become familiar with this big data environment. You are also encouraged to post on CANVAS discussion board to ask others for help. Or, you can visit the TA office hour to get help.

**Project Submission (required to receive a grade for this project)**

1. Screenshot of HDFS file upload;
2. Compile Jar files for java;
3. Result of Java code in IDEA editor
4. Result of Java in Hadoop.

You will submit the above to us via CANVAS.

**Project**

Tasks:

* Setup Hadoop environment.
* Use IDEA Editor to compile Java code to .jar file.
* Upload the given text “data.txt” given on CANVAS under Project-0 to HDFS.
* Run code given to you ‘WordCount.java’ given on CANVAS under Project-0

in IDEA Editor and learn the debug code of “WordCountTest.java”.

* Run this code in Hadoop environment.

**Some Resources:**

1. Resources on install of Hadoop:

[*https://hadoop.apache.org/docs/r3.2.2/hadoop-project-dist/hadoop-common/SingleCluster.html*](https://hadoop.apache.org/docs/r3.2.2/hadoop-project-dist/hadoop-common/SingleCluster.html)

*Hadoop version: 3.0+*

*Use the “Pseudo-Distributed Operation” mode.*

*Notes that: Hadoop supports Mac/Linux/Windows environments. VM is slow and hard to debug code, so if you install the Hadoop environment on your own machine you will be much happier thereafter.*

1. Use hadoop file system commands (e.g., put) to upload a data file into Hadoop cluster.

[*https://hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-common/FileSystemShell.html#put*](https://hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-common/FileSystemShell.html#put)

*Check the hdfs node state:*

[*http://localhost:9870/*](http://localhost:9870/)

* 1. go to Hadoop/sbin directory, start Hadoop hdfs:

./start-dfs.sh

* 1. go to Hadoop/bin directory, create a folder, here let us call it cs585 but any other filename will be fine:

hdfs dfs -mkdir /cs585

* 1. upload data.txt to cs585 folder, notes that you may need add the full path for your data.txt file.

hdfs dfs -put data.txt /cs585/

* 1. check the uploaded file in web

Graphical user interface, text, application

Description automatically generated

1. Resources/link to IDEA editor
2. Download IDEA from <https://www.jetbrains.com/idea/>
3. Create a new java project and select maven as the package management. (you also can use other package management tool like sbt, if you prefer)
4. Copy the dependencies and build block from given pom.xml file to the pom.xml of your project.
5. Click on the “circle” in the “Maven” to update package (it may takes several minutes).
6. Create a new WordCount.java file under src/main/java folder.

Text

Description automatically generated

1. Debug:

Right click the debug function and select “Generate…”

A screenshot of a computer

Description automatically generated with medium confidence

Select “Junit4” and click ok

Graphical user interface, application

Description automatically generated

Copy the code from given WordCountTest.java, and run this code.

Text, chat or text message

Description automatically generated

Check the output in the web.

Graphical user interface, application

Description automatically generated

1. Refer to “Package an application into a JAR﻿” in the following link to package codes to jar file

<https://www.jetbrains.com/help/idea/compiling-applications.html#package_into_jar>

and run Hadoop command to run jar file (note update the path to Hadoop ahd path to jar file).

path/hadoop jar path/project-0.jar WordCount /cs585/data.txt /cs585/output2.txt

Graphical user interface, application, table

Description automatically generated

**Note:**

if your os is macos, go to the jar file directory: out/artifacts/project\_0\_jar and execute the following command to delete META-INF/LICENSE from jar file.

zip -d project-0.jar META-INF/LICENSE

jar tvf project-0.jar | grep -i license

Read the following link to learn more about the wordcount example.

<http://hadoop.apache.org/docs/r2.9.2/hadoop-mapreduce-client/hadoop-mapreduce-client-core/MapReduceTutorial.html>

*----------------------------- the end ------------------------------*